CHAPTER 10

MATHEMATICAL MODELS OF DYNAMIC PHYSICAL SYSTEMS

K. Preston White, Jr.
Department of Systems and Information Engineering
University of Virginia
Charlottesville, Virginia

1 RATIONALE **300**

2 IDEAL ELEMENTS **302**
2.1 Physical Variables 303
2.2 Power and Energy 303
2.3 One-Port Element Laws 304
2.4 Multiport Elements 307

3 SYSTEM STRUCTURE AND INTERCONNECTION LAWS **311**
3.1 A Simple Example 311
3.2 Structure and Graphs 311
3.3 System Relations 313
3.4 Analogs and Duals 314

4 STANDARD FORMS FOR LINEAR MODELS **314**
4.1 I/O Form 315
4.2 Deriving the I/O Form—An Example 316
4.3 State-Variable Form 318
4.4 Deriving the "Natural" State Variables—A Procedure 320
4.5 Deriving the "Natural" State Variables—An Example 320
4.6 Converting from I/O to "Phase-Variable" Form 321

5 APPROACHES TO LINEAR SYSTEMS ANALYSIS **321**
5.1 Transform Methods 322
5.2 Transient Analysis Using Transform Methods 328
5.3 Response to Periodic Inputs Using Transform Methods 332

6 STATE-VARIABLE METHODS **340**
6.1 Solution of the State Equation 342
6.2 Eigenstructure 350

7 SIMULATION **352**
7.1 Simulation—Experimental Analysis of Model Behavior 352
7.2 Digital Simulation 353

8 MODEL CLASSIFICATIONS **359**
8.1 Stochastic Systems 359
8.2 Distributed-Parameter Models 364
8.3 Time-Varying Systems 365
8.4 Nonlinear Systems 366
8.5 Discrete and Hybrid Systems 376

REFERENCES **382**

BIBLIOGRAPHY **382**

1 RATIONALE

The design of modern control systems relies on the formulation and analysis of mathematical models of dynamic physical systems. This is simply because a model is more accessible to study than the physical system the model represents. Models typically are less costly and less time consuming to construct and test. Changes in the structure of a model are easier to implement, and changes in the behavior of a model are easier to isolate and understand. A model often can be used to achieve insight when the corresponding physical system cannot, because experimentation with the actual system is too dangerous or too demanding. Indeed, a model can be used to answer "what if" questions about a system that has not yet been realized or actually cannot be realized with current technologies.

The type of model used by the control engineer depends upon the nature of the system the model represents, the objectives of the engineer in developing the model, and the tools which the engineer has at his or her disposal for developing and analyzing the model. A mathematical model is a description of a system in terms of equations. Because the physical systems of primary interest to the control engineer are dynamic in nature, the mathematical models used to represent these systems most often incorporate difference or differential equations. Such equations, based on physical laws and observations, are statements of the fundamental relationships among the important variables that describe the system. Difference and differential equation models are expressions of the way in which the current values assumed by the variables combine to determine the future values of these variables.

Mathematical models are particularly useful because of the large body of mathematical and computational theory that exists for the study and solution of equations. Based on this theory, a wide range of techniques has been developed specifically for the study of control systems. In recent years, computer programs have been written that implement virtually all of these techniques. Computer software packages are now widely available for both simulation and computational assistance in the analysis and design of control systems.

It is important to understand that a variety of models can be realized for any given physical system. The choice of a particular model always represents a trade-off between the fidelity of the model and the effort required in model formulation and analysis. This trade-off is reflected in the nature and extent of simplifying assumptions used to derive the model. In general, the more faithful the model is as a description of the physical system modeled, the more difficult it is to obtain general solutions. In the final analysis, the best engineering model is not necessarily the most accurate or precise. It is, instead, the simplest model that yields the information needed to support a decision. A classification of various types of models commonly encountered by control engineers is given in Section 8.

A large and complicated model is justified if the underlying physical system is itself complex, if the individual relationships among the system variables are well understood, if it is important to understand the system with a great deal of accuracy and precision, and if time and budget exist to support an extensive study. In this case, the assumptions necessary to formulate the model can be minimized. Such complex models cannot be solved analytically, however. The model itself must be studied experimentally, using the techniques of computer simulation. This approach to model analysis is treated in Section 7.

Simpler models frequently can be justified, particularly during the initial stages of a control system study. In particular, systems that can be described by linear difference or differential equations permit the use of powerful analysis and design techniques. These include the transform methods of classical control theory and the state-variable methods of modern control theory. Descriptions of these standard forms for linear systems analysis are presented in Sections 4, 5, and 6.

During the past several decades, a unified approach for developing lumped-parameter models of physical systems has emerged. This approach is based on the idea of idealized system elements, which store, dissipate, or transform energy. Ideal elements apply equally well to the many kinds of physical systems encountered by control engineers. Indeed, because control engineers most frequently deal with systems that are part mechanical, part electrical, part fluid, and/or part thermal, a unified approach to these various physical systems is especially useful and economic. The modeling of physical systems using ideal elements is discussed further in Sections 2, 3, and 4.

Frequently, more than one model is used in the course of a control system study. Simple models that can be solved analytically are used to gain insight into the behavior of the system and to suggest candidate designs for controllers. These designs are then verified and refined in more complex models, using computer simulation. If physical components are developed during the course of a study, it is often practical to incorporate these components

directly into the simulation, replacing the corresponding model components. An iterative, evolutionary approach to control systems analysis and design is depicted in Fig. 1.

2 IDEAL ELEMENTS

Differential equations describing the dynamic behavior of a physical system are derived by applying the appropriate physical laws. These laws reflect the ways in which energy can be stored and transferred within the system. Because of the common physical basis provided by the concept of energy, a general approach to deriving differential equation models is possible. This approach applies equally well to mechanical, electrical, fluid, and thermal systems and is particularly useful for systems that are combinations of these physical types.

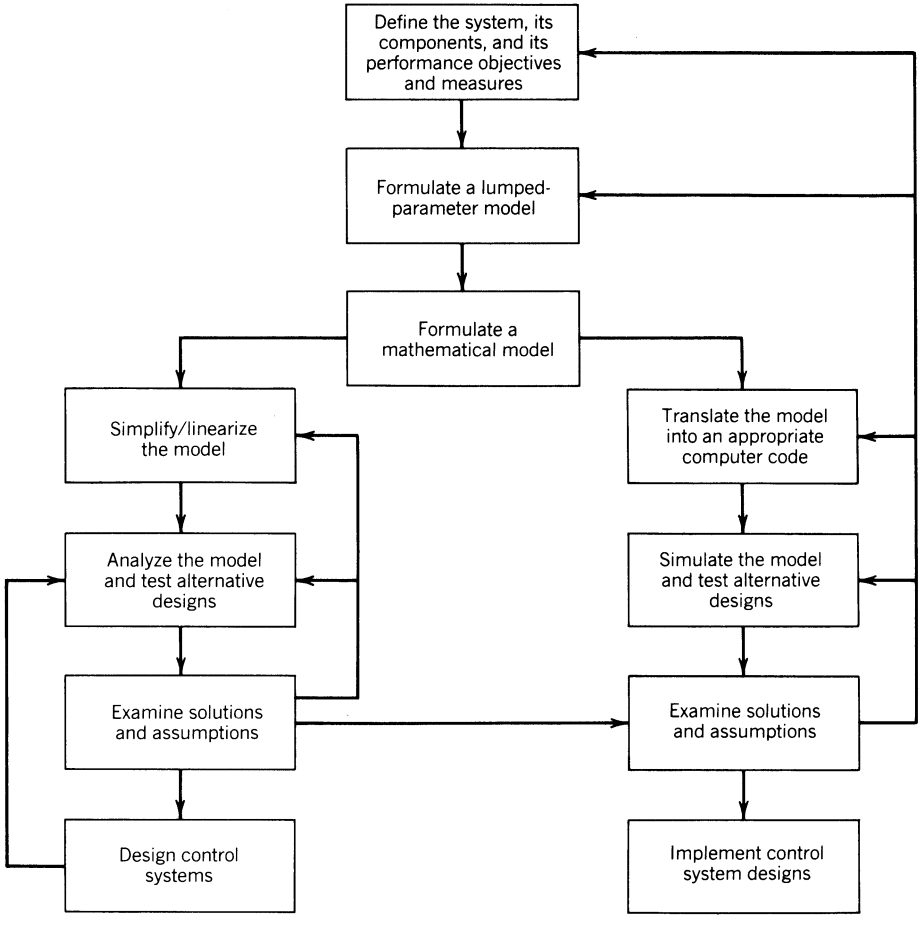


Figure 1 An iterative approach to control system design, showing the use of mathematical analysis and computer simulation.

2.1 Physical Variables

An idealized *two-terminal* or *one-port* element is shown in Fig. 2. Two *primary physical variables* are associated with the element: a through variable $f(t)$ and an across variable $v(t)$. *Through variables* represent quantities that are transmitted through the element, such as the force transmitted through a spring, the current transmitted through a resistor, or the flow of fluid through a pipe. Through variables have the same value at both ends or terminals of the element. *Across variables* represent the difference in state between the terminals of the element, such as the velocity difference across the ends of a spring, the voltage drop across a resistor, or the pressure drop across the ends of a pipe. *Secondary physical variables* are the integrated through variable $h(t)$ and the integrated across variable $x(t)$. These represent the accumulation of quantities within an element as a result of the integration of the associated through and across variables. For example, the momentum of a mass is an integrated through variable, representing the effect of forces on the mass integrated or accumulated over time. Table 1 defines the primary and secondary physical variables for various physical systems.

2.2 Power and Energy

The flow of *power* $P(t)$ into an element through the terminals 1 and 2 is the product of the through variable $f(t)$ and the difference between the across variables $v_2(t)$ and $v_1(t)$. Suppressing the notation for time dependence, this may be written as

$$P = f(v_2 - v_1) = fv_{21}$$

A negative value of power indicates that power flows out of the element. The *energy* $E(t_a, t_b)$ transferred to the element during the time interval from t_a to t_b is the integral of power, that is,

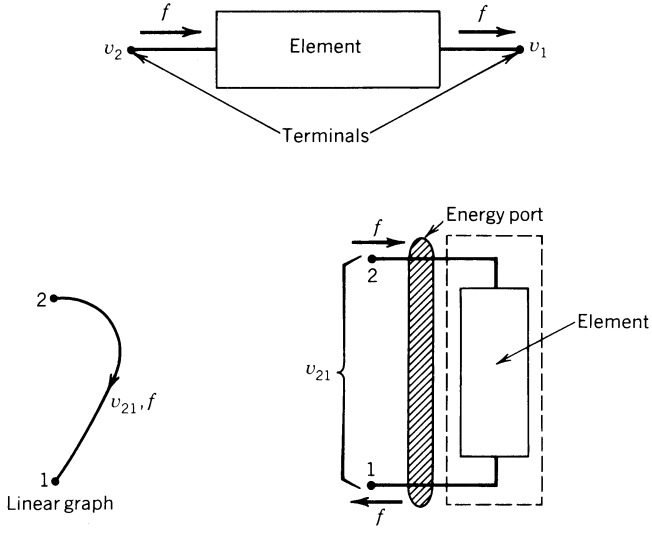


Figure 2 A two-terminal or one-port element, showing through and across variables.[1]

Table 1 Primary and Secondary Physical Variables for Various Systems[1]

System	Through Variable f	Integrated through Variable h	Across Variable v	Integrated across Variable x
Mechanical–translational	Force F	Translational momentum p	Velocity difference v_{21}	Displacement difference x_{21}
Mechanical–rotational	Torque T	Angular momentum h	Angular velocity difference Ω_{21}	Angular displacement difference Θ_{21}
Electrical	Current i	Charge q	Voltage difference v_{21}	Flux linkage λ_{21}
Fluid	Fluid flow Q	Volume V	Pressure difference P_{21}	Pressure–momentum Γ_{21}
Thermal	Heat flow q	Heat energy $\mathcal{H}$	Temperature difference θ_{21}	Not used in general

$$E = \int_{t_a}^{t_b} P\, dt = \int_{t_a}^{t_b} f v_{21}\, dt$$

A negative value of energy indicates a net transfer of energy out of the element during the corresponding time interval.

Thermal systems are an exception to these generalized energy relationships. For a thermal system, power is identically the through variable $q(t)$, heat flow. Energy is the integrated through variable $\mathcal{H}(t_a, t_b)$, the amount of heat transferred.

By the *first law of thermodynamics,* the net energy stored within a system at any given instant must equal the difference between all energy supplied to the system and all energy dissipated by the system. The generalized classification of elements given in the following sections is based on whether the element stores or dissipates energy within the system, supplies energy to the system, or transforms energy between parts of the system.

2.3 One-Port Element Laws

Physical devices are represented by idealized system elements, or by combinations of these elements. A physical device that exchanges energy with its environment through one pair of across and through variables is called a *one-port* or *two-terminal* element. The behavior of a one-port element expresses the relationship between the physical variables for that element. This behavior is defined mathematically by a *constitutive relationship.* Constitutive relationships are derived empirically, by experimentation, rather than from any more fundamental principles. The *element law,* derived from the corresponding constitutive relationship, describes the behavior of an element in terms of across and through variables and is the form most commonly used to derive mathematical models.

Table 2 summarizes the element laws and constitutive relationships for the one-port elements. Passive elements are classified into three types. *T-type* or *inductive storage* elements are defined by a single-valued constitutive relationship between the through variable $f(t)$ and the integrated across-variable difference $x_{21}(t)$. Differentiating the constitutive relationship yields the element law. For a linear (or ideal) T-type element, the element law states that the across-variable difference is proportional to the rate of change of the through variable. Pure translational and rotational compliance (springs), pure electrical inductance, and

Table 2 Element Laws and Constitutive Relationships for Various One-Port Elements[1]

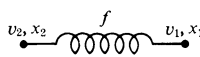
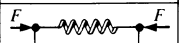
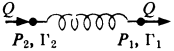
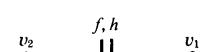
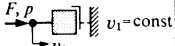
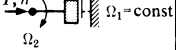
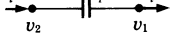
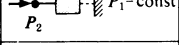
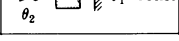
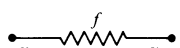
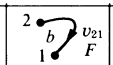
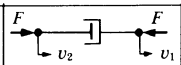
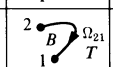
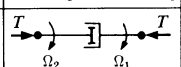
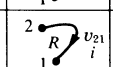
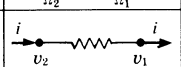
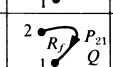
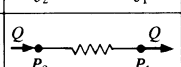
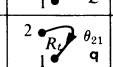
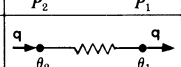
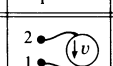
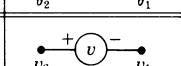
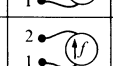
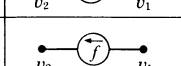
Type of element	Physical element	Linear graph	Diagram	Constitutive relationship	Energy or power function	Ideal elemental equation	Ideal energy or power
T-type energy storage $\mathcal{E} \geqslant 0$  Pure: $x_{21}=f(f)$; $\mathcal{E}=\int_0^f f\,dx_{21}$ Ideal: $x_{21}=Lf$; $\mathcal{E}=\frac{1}{2}Lf^2$	Translational spring			$x_{21}=f(F)$	$\mathcal{E}=\int_0^F F\,dx_{21}$	$v_{21}=\frac{1}{k}\frac{dF}{dt}$	$\mathcal{E}=\frac{1}{2}\frac{F^2}{k}$
	Rotational spring			$\Theta_{21}=f(T)$	$\mathcal{E}=\int_0^T T\,d\Theta_{21}$	$\Omega_{21}=\frac{1}{K}\frac{dT}{dt}$	$\mathcal{E}=\frac{1}{2}\frac{T^2}{K}$
	Inductance			$\lambda_{21}=f(i)$	$\mathcal{E}=\int_0^i i\,d\lambda_{21}$	$v_{21}=L\frac{di}{dt}$	$\mathcal{E}=\frac{1}{2}Li^2$
	Fluid inertance			$\Gamma_{21}=f(Q)$	$\mathcal{E}=\int_0^Q Q\,d\Gamma_{21}$	$P_{21}=I\frac{dQ}{dt}$	$\mathcal{E}=\frac{1}{2}IQ^2$
A-type energy storage $\mathcal{E} \geqslant 0$  Pure: $h=f(v_{21})$; $\mathcal{E}=\int_0^{v_{21}} v_{21}\,dh$ Ideal: $h=Cv_{21}$; $\mathcal{E}=\frac{1}{2}Cv_{21}^2$	Translational mass			$p=f(v_2)$	$\mathcal{E}=\int_0^{v_2} v_2\,dp$	$F=m\frac{dv_2}{dt}$	$\mathcal{E}=\frac{1}{2}mv_2^2$
	Inertia			$h=f(\Omega_2)$	$\mathcal{E}=\int_0^{\Omega_2} \Omega_2\,dh$	$T=J\frac{d\Omega_2}{dt}$	$\mathcal{E}=\frac{1}{2}J\Omega_2^2$
	Electrical capacitance			$q=f(v_{21})$	$\mathcal{E}=\int_0^{v_{21}} v_{21}\,dq$	$i=C\frac{dv_{21}}{dt}$	$\mathcal{E}=\frac{1}{2}Cv_{21}^2$
	Fluid capacitance			$V=f(P_2)$	$\mathcal{E}=\int_0^{P_2} P_2\,dV$	$Q=C_f\frac{dP_2}{dt}$	$\mathcal{E}=\frac{1}{2}C_fP_2^2$
	Thermal capacitance			$\mathfrak{K}=f(\theta_2)$	$\mathcal{E}=\int_0^{\theta_2} \mathbf{q}\,dt=\mathfrak{K}$	$\mathbf{q}=C_t\frac{d\theta_2}{dt}$	$\mathcal{E}=C_t\theta_2$

Table 2 (*Continued*)

D-type energy dissipators $\mathcal{P} \geqslant 0$  Pure: $f = f(v_{21})$, $\mathcal{P} = v_{21} f(v_{21})$ Ideal: $f = \frac{1}{R} v_{21}$, $\mathcal{P} = \frac{1}{R} v_{21}^2 = Rf^2$	Translational damper			$F = f(v_{21})$	$\mathcal{P} = Fv_{21}$	$F = bv_{21}$	$\mathcal{P} = bv_{21}^2$
	Rotational damper			$T = f(\Omega_{21})$	$\mathcal{P} = T\Omega_{21}$	$T = B\Omega_{21}$	$\mathcal{P} = B\Omega_{21}^2$
	Electrical resistance			$i = f(v_{21})$	$\mathcal{P} = iv_{21}$	$i = \frac{1}{R} v_{21}$	$\mathcal{P} = \frac{1}{R} v_{21}^2$
	Fluid resistance			$Q = f(P_{21})$	$\mathcal{P} = QP_{21}$	$Q = \frac{1}{R_f} P_{21}$	$\mathcal{P} = \frac{1}{R_f} P_{21}^2$
	Thermal resistance			$\mathfrak{q} = f(\theta_{21})$	$\mathcal{P} = \mathfrak{q}$	$\mathfrak{q} = \frac{1}{R_t} \theta_{21}$	$\mathcal{P} = \frac{1}{R_t} \theta_{21}$
Energy sources $\mathcal{P} \gtrless 0$ $\mathcal{E} \gtrless 0$	*A*-type across-variable source			$v_{21} = f(t)$	$\mathcal{P} = fv_{21}$		
	T-type through-variable source			$f = f(t)$	$\mathcal{P} = fv_{21}$		

Nomenclature

λ = energy, $\mathcal{P}$ = power

f = generalized through variable, F = force, T = torque, i = current, Q = fluid flow rate, q = heat flow rate

h = generalized integrated through variable, p = translational momentum, h = angular momentum, q = charge, l' = fluid volume displaced, $\mathcal{H}$ = heat

v = generalized across variable, v = translational velocity, Ω = angular velocity, v = voltage, P = pressure, θ = temperature

x = generalized integrated across variable, x = translational displacement, Θ = angular displacement, λ = flux linkage, Γ = pressure-momentum

L = generalized ideal inductance, $1/k$ = reciprocal translational stiffness, $1/K$ = reciprocal rotational stiffness, L = inductance, I = fluid inertance

C = generalized ideal capacitance, m = mass, J = moment of insertia, C = capacitance, C_j = fluid capacitance, C_t = thermal capacitance

R = generalized ideal resistance, $1/b$ = reciprocal translational damping, $1/B$ = reciprocal rotational damping, R = electrical resistance, R_j = fluid resistance, R_t = thermal resistance

pure fluid inertance are examples of *T*-type storage elements. There is no corresponding thermal element.

A-type or *capacitive storage elements* are defined by a single-valued constitutive relationship between the across-variable difference $v_{21}(t)$ and the integrated through variable $h(t)$. These elements store energy by virtue of the across variable. Differentiating the constitutive relationship yields the element law. For a linear *A*-type element, the element law states that the through variable is proportional to the derivative of the across-variable difference. Pure translational and rotational inertia (masses) and pure electrical, fluid, and thermal capacitance are examples.

It is important to note that when a nonelectrical capacitance is represented by an *A*-type element, one terminal of the element must have a constant (reference) across variable, usually assumed to be zero. In a mechanical system, for example, this requirement expresses the fact that the velocity of a mass must be measured relative to a noninertial (nonaccelerating) reference frame. The constant-velocity terminal of a pure mass may be thought of as being attached in this sense to the reference frame.

D-type or *resistive elements* are defined by a single-valued constitutive relationship between the across and the through variables. These elements dissipate energy, generally by converting energy into heat. For this reason, power always flows into a *D*-type element. The element law for a *D*-type energy dissipator is the same as the constitutive relationship. For a linear dissipator, the through variable is proportional to the across-variable difference. Pure translational and rotational friction (dampers or dashpots) and pure electrical, fluid, and thermal resistance are examples.

Energy storage and energy-dissipating elements are called *passive* elements, because such elements do not supply outside energy to the system. The fourth set of one-port elements are *source elements,* which are examples of *active* or power-supplying elements. Ideal sources describe interactions between the system and its environment. A pure *A-type source* imposes an across-variable difference between its terminals, which is a prescribed function of time, regardless of the values assumed by the through variable. Similarly, a pure *T-type source* imposes a through-variable flow through the source element, which is a prescribed function of time, regardless of the corresponding across variable.

Pure system elements are used to represent physical devices. Such models are called *lumped-element models.* The derivation of lumped-element models typically requires some degree of approximation, since (1) there rarely is a one-to-one correspondence between a physical device and a set of pure elements and (2) there always is a desire to express an element law as simply as possible. For example, a coil spring has both mass and compliance. Depending on the context, the physical spring might be represented by a pure translational mass, or by a pure translational spring, or by some combination of pure springs and masses. In addition, the physical spring undoubtedly will have a nonlinear constitutive relationship over its full range of extension and compression. The compliance of the coil spring may well be represented by an ideal translational spring, however, if the physical spring is approximately linear over the range of extension and compression of concern.

2.4 Multiport Elements

A physical device that exchanges energy with its environment through two or more pairs of through and across variables is called a *multiport element.* The simplest of these, the idealized *four-terminal* or *two-port* element, is shown in Fig. 3. Two-port elements provide for transformations between the physical variables at different energy ports, while maintaining instantaneous continuity of power. In other words, net power flow into a two-port element is always identically zero:

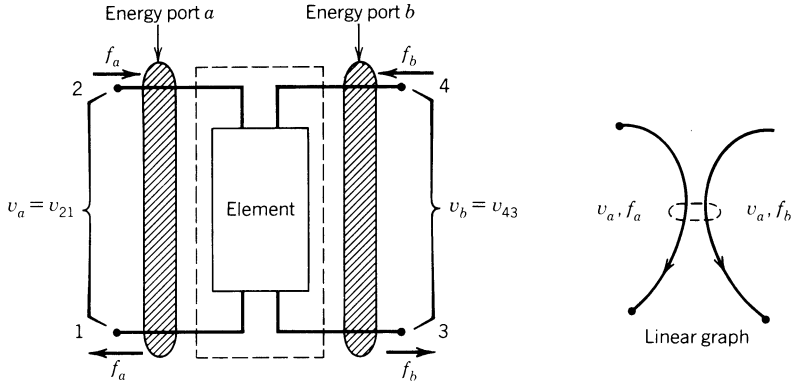


Figure 3 A four-terminal or two-port element, showing through and across variables.

$$P = f_a v_a + f_b v_b = 0$$

The particulars of the transformation between the variables define different categories of two-port elements.

A *pure transformer* is defined by a single-valued constitutive relationship between the integrated across variables or between the integrated through variables at each port:

$$x_b = f(x_a) \qquad \text{or} \qquad h_b = f(h_a)$$

For a linear (or ideal) transformer, the relationship is proportional, implying the following relationships between the primary variables:

$$v_b = nv_a \qquad f_b = -\frac{1}{n} f_a$$

where the constant of proportionality n is called the *transformation ratio.* Levers, mechanical linkages, pulleys, gear trains, electrical transformers, and differential-area fluid pistons are examples of physical devices that typically can be approximated by pure or ideal transformers. Figure 4 depicts some examples. *Pure transmitters,* which serve to transmit energy over a distance, frequently can be thought of as transformers with $n = 1$.

A *pure gyrator* is defined by a single-valued constitutive relationship between the across variable at one energy port and the through variable at the other energy port. For a linear gyrator, the following relations apply:

$$v_b = rf_a \qquad f_b = \frac{-1}{r} v_a$$

where the constant of proportionality is called the *gyration ratio* or *gyrational resistance.* Physical devices that perform pure gyration are not as common as those performing pure transformation. A mechanical gyroscope is one example of a system that might be modeled as a gyrator.

In the preceding discussion of two-port elements, it has been assumed that the type of energy is the same at both energy ports. A *pure transducer,* on the other hand, changes energy from one physical medium to another. This change may be accomplished as either a transformation or a gyration. Examples of *transforming transducers* are gears with racks (mechanical rotation to mechanical translation) and electric motors and electric generators

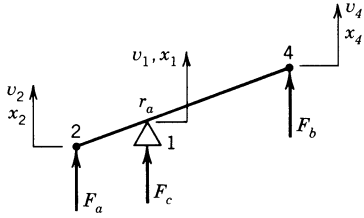
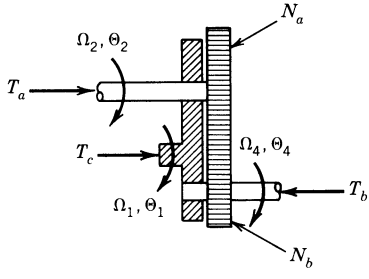
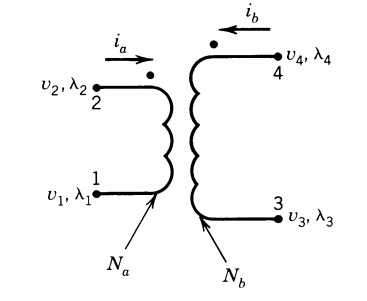
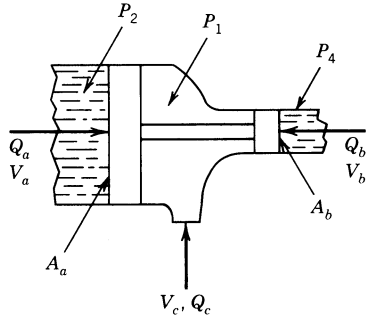
System	Symbol	Pure transformer	Ideal transformer	Transformation ratio
Mechanical translation (lever)		$x_{41} = f(x_{21})$	$v_{41} = nv_{21}$ $F_b = -\frac{1}{n} F_a$	$n = -\frac{r_b}{r_a}$ Lever ratio
Mechanical rotational (gears)		$\Theta_{41} = f(\Theta_2)$	$\Omega_{41} = n\Omega_{21}$ $T_b = -\frac{1}{n} T_a$	$n = -\frac{N_a}{N_b}$ Gear ratio
Electrical (magnetic)		$\lambda_{43} = f(\lambda_{21})$	$v_{43} = nv_{21}$ $i_b = -\frac{1}{n} i_a$	$n = \frac{N_b}{N_a}$ Turns ratio
Fluid (differential piston)		$V_b = f(V_a)$	$P_{41} = nP_{21}$ $Q_b = -\frac{1}{n} Q_a$	$n = \frac{A_a}{A_b}$ Area ratio

Figure 4*a* Examples of transforms and transducers: pure transformers.[1]

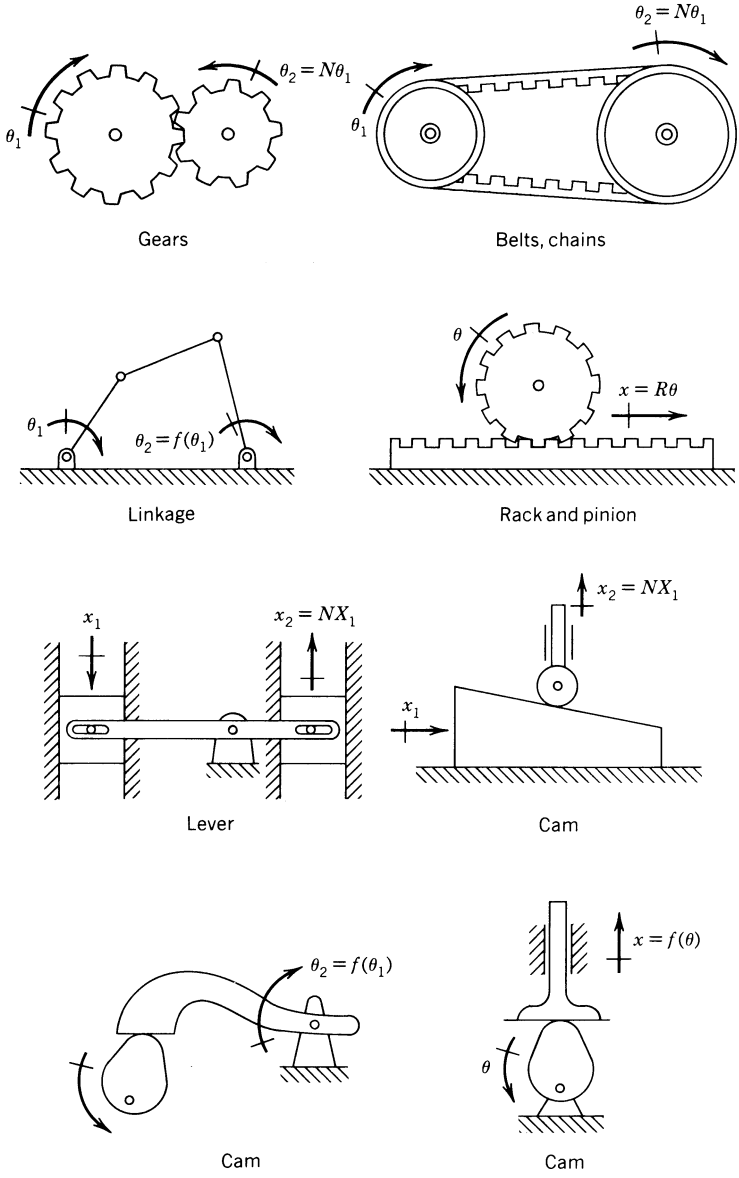


Figure 4*b* Examples of transformers and transducers: pure mechanical transformers and transforming transducers.[2]

(electrical to mechanical rotation and vice versa). Examples of *gyrating transducers* are the piston-and-cylinder (fluid to mechanical) and piezoelectric crystals (mechanical to electrical).

More complex systems may have a large number of energy ports. A common *six-terminal* or *three-port element* called a *modulator* is depicted in Fig. 5. The flow of energy between ports *a* and *b* is controlled by the energy input at the modulating port *c*. Such devices inherently dissipate energy, since

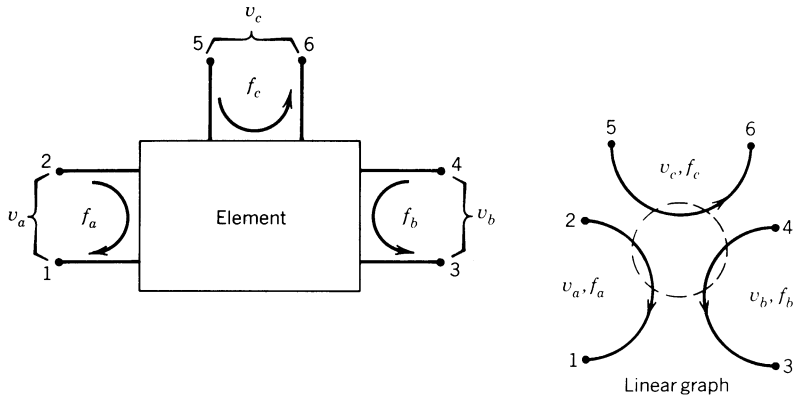


Figure 5 A six-terminal or three-port element, showing through and across variables.

$$P_a + P_c \geq P_b$$

although most often the modulating power P_c is much smaller than the power input P_a or the power output P_b. When port a is connected to a pure source element, the combination of source and modulator is called a *pure dependent source.* When the modulating power P_c is considered the input and the modulated power P_b is considered the output, the modulator is called an *amplifier.* Physical devices that often can be modeled as modulators include clutches, fluid valves and couplings, switches, relays, transistors, and variable resistors.

3 SYSTEM STRUCTURE AND INTERCONNECTION LAWS

3.1 A Simple Example

Physical systems are represented by connecting the terminals of pure elements in patterns that approximate the relationships among the properties of component devices. As an example, consider the mechanical–translational system depicted in Fig. 6*a*, which might represent an idealized automobile suspension system. The inertial properties associated with the masses of the chassis, passenger compartment, engine, and so on, all have been lumped together as the pure mass m_1. The inertial properties of the unsprung components (wheels, axles, etc.) have been lumped into the pure mass m_2. The compliance of the suspension is modeled as a pure spring with stiffness k_1 and the frictional effects (principally from the shock absorbers) as a pure damper with damping coefficient b. The road is represented as an input or source of vertical velocity, which is transmitted to the system through a spring of stiffness k_2, representing the compliance of the tires.

3.2 Structure and Graphs

The *pattern of interconnections* among elements is called the *structure* of the system. For a one-dimensional system, structure is conveniently represented by a *system graph.* The system graph for the idealized automobile suspension system of Fig. 6*a* is shown in Fig. 6*b*. Note that each distinct across variable (velocity) becomes a distinct *node* in the graph. Each distinct through variable (force) becomes a *branch* in the graph. Nodes coincide with the

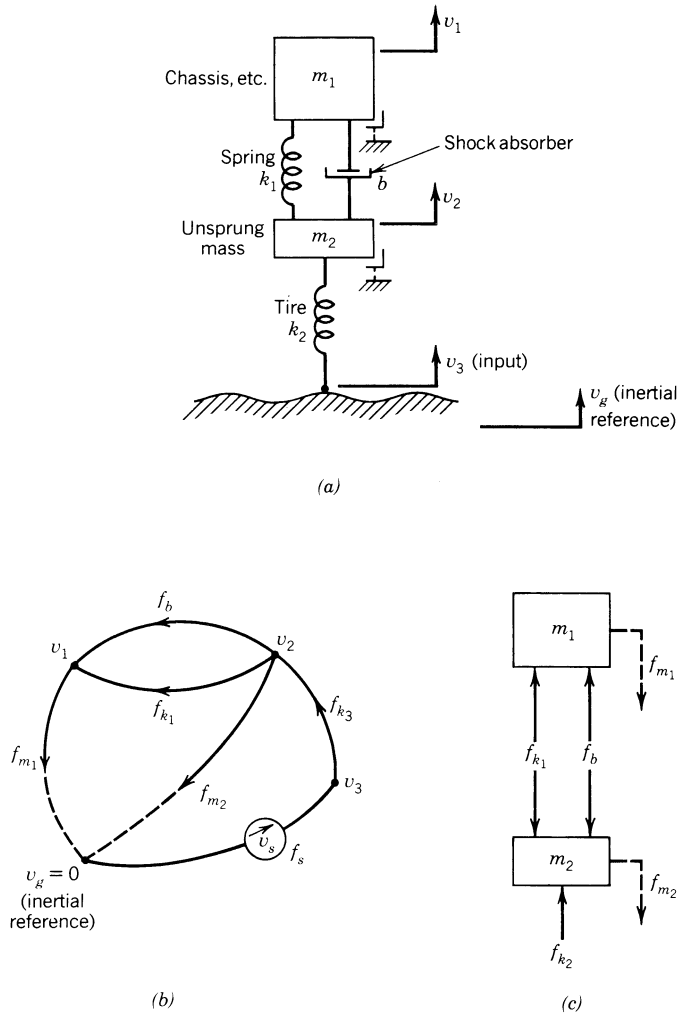


Figure 6 An idealized model of an automobile suspension system: (*a*) lumped-element model, (*b*) system graph, (*c*) free-body diagram.

terminals of elements and branches coincide with the elements themselves. One node always represents *ground* (the constant velocity of the inertial reference frame v_g), and this is usually assumed to be zero for convenience. For nonelectrical systems, all the *A*-type elements (masses) have one terminal connection to the reference node. Because the masses are not physically connected to ground, however, the convention is to represent the corresponding branches in the graph by dashed lines.

System graphs are oriented by placing arrows on the branches. The orientation is arbitrary and serves to assign reference directions for both the through-variable and the across-variable difference. For example, the branch representing the damper in Fig. 6*b* is directed from node 2 (tail) to node 1 (head). This assigns $v_b = v_{21} = v_2 - v_1$ as the across-variable difference to be used in writing the damper elemental equation

$$f_b = bv_b = bv_{21}$$

The reference direction for the through variable is determined by the convention that power flow $P_b = f_b v_b$ into an element is positive. Referring to Fig. 6*a*, when v_{21} is positive, the damper is in compression. Therefore, f_b must be positive for compressive forces in order to obey the sign convention for power. By similar reasoning, tensile forces will be negative.

3.3 System Relations

The structure of a system gives rise to two sets of *interconnection laws* or *system relations.* Continuity relations apply to through variables and compatibility relations apply to across variables. The interpretation of system relations for various physical systems is given in Table 3.

Continuity is a general expression of dynamic equilibrium. In terms of the system graph, continuity states that the algebraic sum of all through variables entering a given node must be zero. Continuity applies at each node in the graph. For a graph with n nodes, continuity gives rise to n continuity equations, $n - 1$ of which are independent. For node i, the continuity equation is

$$\sum_j f_{ij} = 0$$

where the sum is taken over all branches (i, j) incident on i.

For the system graph depicted in Fig. 6*b*, the four continuity equations are

$$\text{node 1:} \qquad f_{k_1} + f_b - f_{m_1} = 0$$

$$\text{node 2:} \qquad f_{k_2} - f_{k_1} - f_b - f_{m_2} = 0$$

$$\text{node 3:} \qquad f_s - f_{k_2} = 0$$

$$\text{node } g\text{:} \qquad f_{m_1} + f_{m_2} - f_s = 0$$

Only three of these four equations are independent. Note, also, that the equations for nodes 1–3 could have been obtained from the conventional *free-body diagrams* shown in Fig. 6*c*, where f_{m_1} and f_{m_2} are the *D'Alembert forces* associated with the pure masses. Continuity relations are also known as *vertex, node, flow,* and *equilibrium relations.*

Compatibility expresses the fact that the magnitudes of all across variables are scalar quantities. In terms of the system graph, compatibility states that the algebraic sum of the across-variable differences around any closed path in the graph must be zero. Compatibility

Table 3 System Relations for Various Systems

System	Continuity	Compatibility
Mechanical	Newton's first and third laws (conservation of momentum)	Geometrical constraints (distance is a scalar)
Electrical	Kirchhoff's current law (conservation of charge)	Kirchhoff's voltage law (potential is a scalar)
Fluid	Conservation of matter	Pressure is a scalar
Thermal	Conservation of energy	Temperature is a scalar

applies to any closed path in the system. For convenience and to ensure the independence of the resulting equations, continuity is usually applied to the *meshes* or "windows" of the graph. A one-part graph with n nodes and b branches will have $b - n + 1$ meshes, each mesh yielding one independent compatibility equation. A planar graph with p separate parts (resulting from multiport elements) will have $b - n + p$ independent compatibility equations. For a closed path q, the compatibility equation is

$$\sum_{q} v_{ij} = 0$$

where the summation is taken over all branches (i, j) on the path.

For the system graph depicted in Fig. 6*b*, the three compatibility equations based on the meshes are

$$\text{path } 1 \to 2 \to g \to 1: \qquad -v_b + v_{m_2} - v_{m_1} = 0$$

$$\text{path } 1 \to 2 \to 1: \qquad -v_{k_1} + v_b = 0$$

$$\text{path } 2 \to 3 \to g \to 2: \qquad -v_{k_2} - v_s - v_{m_2} = 0$$

These equations are all mutually independent and express apparent geometric identities. The first equation, for example, states that the velocity difference between the ends of the damper is identically the difference between the velocities of the masses it connects. Compatibility relations are also known as *path, loop,* and *connectedness* relations.

3.4 Analogs and Duals

Taken together, the element laws and system relations are a complete mathematical model of a system. When expressed in terms of generalized through and across variables, the model applies not only to the physical system for which it was derived, but also to any physical system with the same generalized system graph. Different physical systems with the same generalized model are called *analogs.* The mechanical rotational, electrical, and fluid analogs of the mechanical translational system of Fig. 6*a* are shown in Fig. 7. Note that because the original system contains an inductive storage element, there is no thermal analog.

Systems of the same physical type but in which the roles of the through variables and the across variables have been interchanged are called *duals.* The analog of a dual—or, equivalently, the dual of an analog—is sometimes called a *dualog.* The concepts of analogy and duality can be exploited in many different ways.

4 STANDARD FORMS FOR LINEAR MODELS

The element laws and system relations together constitute a complete mathematical description of a physical system. For a system graph with n nodes, b branches, and s sources, there will be $b - s$ element laws, $n - 1$ continuity equations, and $b - n + 1$ compatibility equations. This is a total of $2b - s$ differential and algebraic equations. For systems composed entirely of linear elements, it is always possible to reduce these $2b - s$ equations to either of two standard forms. The *input/output,* or *I/O*, *form* is the basis for *transform* or so-called *classical linear systems analysis.* The *state-variable form* is the basis for *state-variable* or so-called *modern linear systems analysis.*

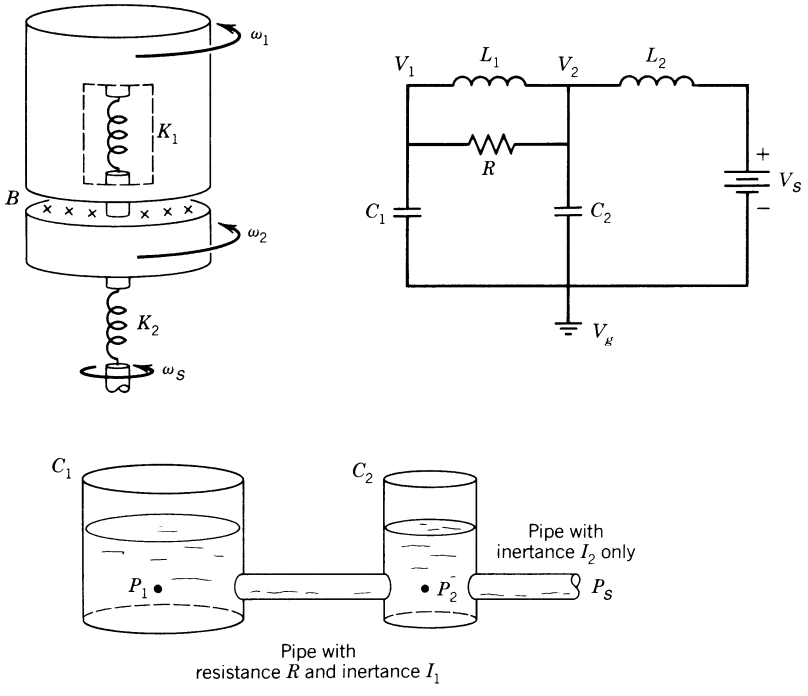


Figure 7 Analogs of the idealized automobile suspension system depicted in Fig. 6.

4.1 I/O Form

The classical representation of a system is the "blackbox," depicted in Fig. 8. The system has a set of p *inputs* (also called *excitations* or *forcing functions*), $u_j(t)$, $j = 1, 2, \ldots, p$. The system also has a set of q *outputs* (also called *response variables*), $y_k(t)$, $k = 1, 2, \ldots, q$. Inputs correspond to sources and are assumed to be known functions of time. Outputs correspond to physical variables that are to be measured or calculated.

Linear systems represented in I/O form can be modeled mathematically by *I/O differential equations.* Denoting as $y_{kj}(t)$ that part of the kth output $y_k(t)$ that is attributable to the jth input $u_j(t)$, there are $(p \times q)$ I/O equations of the form

$$\frac{d^n y_{kj}}{dt^n} + a_{n-1}\frac{d^{n-1}y_{kj}}{dt^{n-1}} + \cdots + a_1\frac{dy_{kj}}{dt} + a_0 y_{kj}(t)$$

$$= b_m\frac{d^m u_j}{dt^m} + b_{m-1}\frac{d^{m-1}u_j}{dt^{m-1}} + \cdots + b_1\frac{du_j}{dt} + b_0 u_j(t)$$

where $j = 1, 2, \ldots, p$ and $k = 1, 2, \ldots, q$. Each equation represents the dependence of one output and its derivatives on one input and its derivatives. By the *principle of superposition,* the kth output in response to all of the inputs acting simultaneously is

$$y_k(t) = \sum_{j=1}^{p} y_{kj}(t)$$

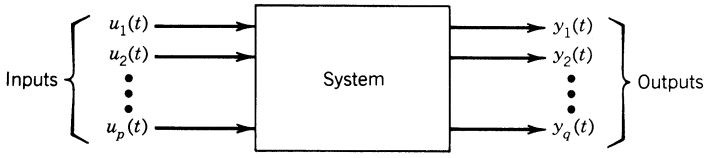


Figure 8 Input/output (I/O), or blackbox, representation of a dynamic system.

A system represented by *n*th-order I/O equations is called an *n*th-*order system.* In general, the order of a system is determined by the number of *independent* energy storage elements within the system, that is, by the combined number of *T*-type and *A*-type elements for which the initial energy stored can be independently specified.

The coefficients $a_0, a_1, \ldots, a_{n-1}$ and $b_0, b_1, \ldots, b_m$ are parameter groups made up of algebraic combinations of the system physical parameters. For a system with constant parameters, therefore, these coefficients are also constant. Systems with constant parameters are called *time-invariant* systems and are the basis for classical analysis.

4.2 Deriving the I/O Form—An Example

I/O differential equations are obtained by combining element laws and continuity and compatibility equations in order to eliminate all variables except the input and the output. As an example, consider the mechanical system depicted in Fig. 9*a*, which might represent an idealized milling machine. A rotational motor is used to position the table of the machine tool through a rack and pinion. The motor is represented as a torque source T with inertia J and internal friction B. A flexible shaft, represented as a torsional spring K, is connected to a pinion gear of radius R. The pinion meshes with a rack, which is rigidly attached to the table of mass m. Damper b represents the friction opposing the motion of the table. The problem is to determine the I/O equation that expresses the relationship between the input torque T and the position of the table x.

The corresponding system graph is depicted in Fig. 9*b*. Applying continuity at nodes 1, 2, and 3 yields

$$\text{node 1:} \quad T - T_J - T_B - T_K = 0$$

$$\text{node 2:} \quad T_K - T_p = 0$$

$$\text{node 3:} \quad -f_r - f_m - f_b = 0$$

Substituting the elemental equation for each of the one-port elements into the continuity equations and assuming zero ground velocities yield

$$\text{node 1:} \quad T - J\dot{\omega}_1 - B\omega_1 - K\int(\omega_1 - \omega_2)\,dt = 0$$

$$\text{node 2:} \quad K\int(\omega_1 - \omega_2)\,dt - T_p = 0$$

$$\text{node 3:} \quad -f_r - m\dot{v} - bv = 0$$

Note that the definition of the across variables for each element in terms of the node variables, as above, guarantees that the compatibility equations are satisfied. With the addition of the constitutive relationships for the rack and pinion

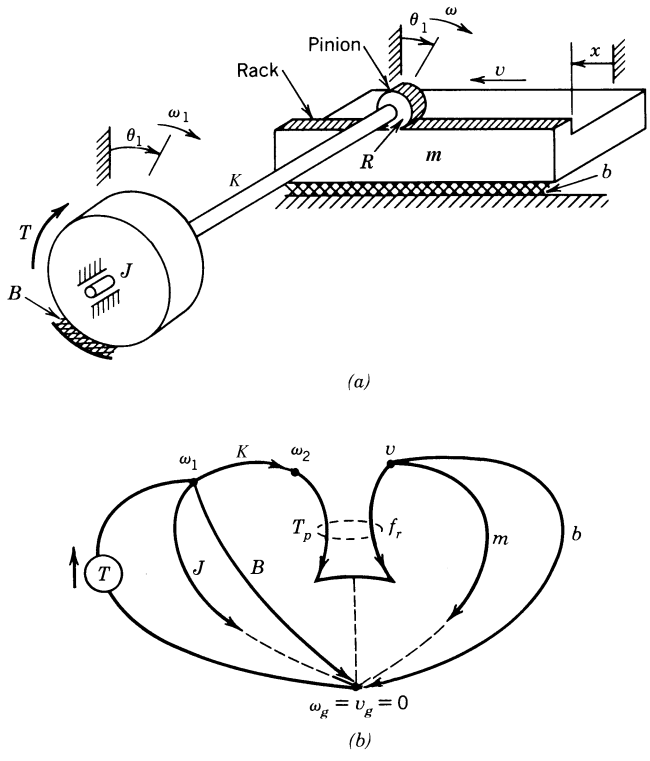


Figure 9 An idealized model of a milling machine: (*a*) lumped-element model,[3] (*b*) system graph.

$$\omega_2 = \frac{1}{R} v \qquad \text{and} \qquad T_p = -Rf_r$$

there are now five equations in the five unknowns ω_1, ω_2, v, T_p, and f_r. Combining these equations to eliminate all of the unknowns except v yields, after some manipulation,

$$a_3 \frac{d^3v}{dt^3} + a_2 \frac{d^2v}{dt^2} + a_1 \frac{dv}{dt} + a_0 v = b_1 T$$

where

$$a_3 = Jm \qquad a_1 = \frac{JK}{R^2} + Bb + mK \qquad b_1 = \frac{K}{R}$$

$$a_2 = Jb + mB \qquad a_0 = \frac{BK}{R^2} + Kb$$

Differentiating yields the desired I/O equation

$$a_3 \frac{d^3x}{dt^3} + a_2 \frac{d^2x}{dt^2} + a_1 \frac{dx}{dt} + a_0 x = b_1 \frac{dT}{dt}$$

where the coefficients are unchanged.

For many systems, combining element laws and system relations can best be achieved by ad hoc procedures. For more complicated systems, formal methods are available for the orderly combination and reduction of equations. These are the so-called *loop method* and *node method* and correspond to procedures of the same names originally developed in connection with electrical networks. The interested reader should consult Ref. 1.

4.3 State-Variable Form

For systems with multiple inputs and outputs, the I/O model form can become unwieldy. In addition, important aspects of system behavior can be suppressed in deriving I/O equations. The "modern" representation of dynamic systems, called the *state-variable form,* largely eliminates these problems. A state-variable model is the maximum reduction of the original element laws and system relations that can be achieved without the loss of any information concerning the behavior of a system. State-variable models also provide a convenient representation for systems with multiple inputs and outputs and for systems analysis using computer simulation.

State variables are a set of variables $x_1(t)$, $x_2(t)$, . . . , $x_n(t)$ internal to the system from which any set of outputs can be derived, as depicted schematically in Fig. 10. A set of state variables is the minimum number of independent variables such that by knowing the values of these variables at any time t_0 and by knowing the values of the inputs for all time $t \geq t_0$, the values of the state variables for all future time $t \geq t_0$ can be calculated. For a given system, the number n of state variables is unique and is equal to the order of the system. The definition of the state variables is not unique, however, and various combinations of one set of state variables can be used to generate alternative sets of state variables. For a physical system, the state variables summarize the *energy state* of the system at any given time.

A complete state-variable model consists of two sets of equations, the *state* or *plant equations* and the *output equations.* For the most general case, the state equations have the form

$$\begin{aligned}
\dot{x}_1(t) &= f_1[x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_p(t)] \\
\dot{x}_2(t) &= f_2[x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_p(t)] \\
&\vdots \\
\dot{x}_n(t) &= f_n[x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_p(t)]
\end{aligned}$$

and the output equations have the form

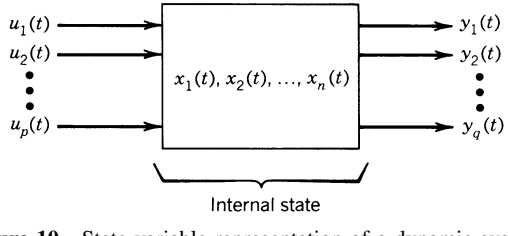


Figure 10 State-variable representation of a dynamic system.

$$
\begin{aligned}
y_1(t) &= g_1[x_1(t),x_2(t), \ldots , x_n(t),u_1(t),u_2(t), \ldots , u_p(t)] \\
y_2(t) &= g_2[x_1(t),x_2(t), \ldots , x_n(t),u_1(t),u_2(t), \ldots , u_p(t)] \\
&\vdots \\
y_q(t) &= g_q[x_1(t),x_2(t), \ldots , x_n(t),u_1(t),u_2(t), \ldots , u_p(t)]
\end{aligned}
$$

These equations are expressed more compactly as the two vector equations

$$
\begin{aligned}
\dot{x}(t) &= f[x(t),u(t)] \\
y(t) &= g[x(t),u(t)]
\end{aligned}
$$

where

$$
\begin{aligned}
\dot{x}(t) &= n \times 1 \textit{ state vector} \\
u(t) &= p \times 1 \textit{ input} \text{ or } \textit{control vector} \\
y(t) &= q \times 1 \textit{ output} \text{ or } \textit{response vector}
\end{aligned}
$$

and f and g are vector-valued functions.

For linear systems, the state equations have the form

$$
\begin{aligned}
\dot{x}_1(t) &= a_{11}(t)x_1(t) + \cdots + a_{1n}(t)x_n(t) + b_{11}(t)u_1(t) + \cdots + b_{1p}(t)u_p(t) \\
\dot{x}_2(t) &= a_{21}(t)x_1(t) + \cdots + a_{2n}(t)x_n(t) + b_{21}(t)u_1(t) + \cdots + b_{2p}(t)u_p(t) \\
&\vdots \\
\dot{x}_n(t) &= a_{n1}(t)x_1(t) + \cdots + a_{nn}(t)x_n(t) + b_{n1}(t)u_1(t) + \cdots + b_{np}(t)u_p(t)
\end{aligned}
$$

and the output equations have the form

$$
\begin{aligned}
y_1(t) &= c_{11}(t)x_1(t) + \cdots + c_{1n}(t)x_n(t) + d_{11}(t)u_1(t) + \cdots + d_{1p}(t)u_p(t) \\
y_2(t) &= c_{21}(t)x_1(t) + \cdots + c_{2n}(t)x_n(t) + d_{21}(t)u_1(t) + \cdots + d_{2p}(t)u_p(t) \\
&\vdots \\
y_n(t) &= c_{q1}(t)x_1(t) + \cdots + c_{qn}(t)x_n(t) + d_{q1}(t)u_1(t) + \cdots + d_{qp}(t)u_p(t)
\end{aligned}
$$

where the coefficients are groups of parameters. The linear model is expressed more compactly as the two linear vector equations

$$
\begin{aligned}
\dot{x}(t) &= A(t)x(t) + B(t)u(t) \\
y(t) &= C(t)x(t) + D(t)u(t)
\end{aligned}
$$

where the vectors x, u, and y are the same as the general case and the matrices are defined as

$$
\begin{aligned}
A &= [a_{ij}] \text{ is the } n \times n \textit{ system matrix} \\
B &= [b_{jk}] \text{ is the } n \times p \textit{ control}, \textit{ input}, \text{ or} \\
&\quad \textit{distribution matrix} \\
C &= [c_{lj}] \text{ is the } q \times n \textit{ output matrix} \\
D &= [d_{lk}] \text{ is the } q \times p \textit{ output distribution matrix}
\end{aligned}
$$

For a time-invariant linear system, all of these matrices are constant.

4.4 Deriving the "Natural" State Variables—A Procedure

Because the state variables for a system are not unique, there are an unlimited number of alternative (but equivalent) state-variable models for the system. Since energy is stored only in generalized system storage elements, however, a natural choice for the state variables is the set of through and across variables corresponding to the independent *T*-type and *A*-type elements, respectively. This definition is sometimes called the set of *natural state variables* for the system.

For linear systems, the following procedure can be used to reduce the set of element laws and system relations to the natural state-variable model.

Step 1. For each independent *T*-type storage, write the element law with the derivative of the through variable isolated on the left-hand side, that is, $\dot{f} = L^{-1}v$.

Step 2. For each independent *A*-type storage, write the element law with the derivative of the across variable isolated on the left-hand side, that is, $\dot{v} = C^{-1}f$.

Step 3. Solve the compatibility equations, together with the element laws for the appropriate *D*-type and multiport elements, to obtain each of the across variables of the independent *T*-type elements in terms of the natural state variables and specified sources.

Step 4. Solve the continuity equations, together with the element laws for the appropriate *D*-type and multiport elements, to obtain the through variables of the *A*-type elements in terms of the natural state variables and specified sources.

Step 5. Substitute the results of step 3 into the results of step 1; substitute the results of step 4 into the results of step 2.

Step 6. Collect terms on the right-hand side and write in vector form.

4.5 Deriving the "Natural" State Variables—An Example

The six-step process for deriving a natural state-variable representation, outlined in the preceding section, is demonstrated for the idealized automobile suspension depicted in Fig. 6:

Step 1

$$\dot{f}_{k_1} = k_1 v_{k_1} \qquad \dot{f}_{k_2} = k_2 v_{k_2}$$

Step 2

$$\dot{v}_{m_1} = m_1^{-1} f_{m_1} \qquad \dot{v}_{m_2} = m_2^{-1} f_{m_2}$$

Step 3

$$v_{k_1} = v_b = v_{m_2} - v_{m_1} \qquad v_{k_2} = -v_{m_2} - v_s$$

Step 4

$$f_{m_1} = f_{k_1} + f_b = f_{k_1} + b^{-1}(v_{m_2} - v_{m_1})$$

$$f_{m_2} = f_{k_2} - f_{k_1} - f_b = f_{k_2} - f_{k_1} - b^{-1}(v_{m_2} - v_{m_1})$$

Step 5

$$\dot{f}_{k_1} = k_1(v_{m_2} - v_{m_1}) \qquad \dot{v}_{m_1} = m_1^{-1}[f_{k_1} + b^{-1}(v_{m_2} - v_{m_1})]$$

$$\dot{f}_{k_2} = k_2(-v_{m_2} - v_s) \qquad \dot{v}_{m_2} = m_2^{-1}[f_{k_2} - f_{k_1} - b^{-1}(v_{m_2} - v_{m_1})]$$

Step 6

$$\frac{d}{dt}\begin{bmatrix} f_{k_1} \\ f_{k_2} \\ v_{m_1} \\ v_{m_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -k_1 & k_1 \\ 0 & 0 & 0 & -k_2 \\ 1/m_1 & 0 & -1/m_1 b & 1/m_1 b \\ -1/m_2 & 1/m_2 & 1/m_2 b & -1/m_2 b \end{bmatrix}\begin{bmatrix} f_{k_1} \\ f_{k_2} \\ v_{m_1} \\ v_{m_2} \end{bmatrix} + \begin{bmatrix} 0 \\ -k_2 \\ 0 \\ 0 \end{bmatrix} v_s$$

4.6 Converting from I/O to "Phase-Variable" Form

Frequently, it is desired to determine a state-variable model for a dynamic system for which the I/O equation is already known. Although an unlimited number of such models is possible, the easiest to determine uses a special set of state variables called the *phase variables.* The phase variables are defined in terms of the output and its derivatives as follows:

$$\begin{aligned} x_1(t) &= y(t) \\ x_2(t) &= \dot{x}_1(t) = \frac{d}{dt}\, y(t) \\ x_3(t) &= \dot{x}_2(t) = \frac{d^2}{dt^2}\, y(t) \\ &\vdots \\ x_n(t) &= \dot{x}_{n-1}(t) = \frac{d^{n-1}}{dt^{n-1}}\, y(t) \end{aligned}$$

This definition of the phase variables, together with the I/O equation of Section 4.1, can be shown to result in a state equation of the form

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t)$$

and an output equation of the form

$$y(t) = [b_0 \quad b_1 \cdots b_m]\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}$$

This special form of the system matrix, with 1s along the upper off-diagonal and 0s elsewhere except for the bottom row, is called a *companion matrix.*

5 APPROACHES TO LINEAR SYSTEMS ANALYSIS

There are two fundamental approaches to the analysis of linear, time-invariant systems. *Transform methods* use rational functions obtained from the Laplace transformation of the

system I/O equations. Transform methods provide a particularly convenient algebra for combining the component submodels of a system and form the basis of so-called *classical control theory. State-variable methods* use the vector state and output equations directly. State-variable methods permit the adaptation of important ideas from linear algebra and form the basis for so-called *modern control theory.* Despite the deceiving names of "classical" and "modern," the two approaches are complementary. Both approaches are widely used in current practice and the control engineer must be conversant with both.

5.1 Transform Methods

A *transformation* converts a given mathematical problem into an equivalent problem, according to some well-defined rule called a *transform.* Prudent selection of a transform frequently results in an equivalent problem that is easier to solve than the original. If the solution to the original problem can be recovered by an inverse transformation, the three-step process of (1) transformation, (2) solution in the *transform domain,* and (3) inverse transformation may prove more attractive than direct solution of the problem in the original problem domain. This is true for fixed linear dynamic systems under the *Laplace transform,* which converts differential equations into equivalent algebraic equations.

Laplace Transforms: Definition

The one-sided Laplace transform is defined as

$$F(s) = \mathcal{L}[f(t)] = \int_0^\infty f(t)e^{-st}\,dt$$

and the inverse transform as

$$f(t) = \mathcal{L}^{-1}[F(s)] = \frac{1}{2\pi j}\int_{\sigma-j\omega}^{\sigma+j\omega} F(s)e^{-st}\,ds$$

The Laplace transform converts the function $f(t)$ into the transformed function $F(s)$; the inverse transform recovers $f(t)$ from $F(s)$. The symbol $\mathcal{L}$ stands for the "Laplace transform of"; the symbol $\mathcal{L}^{-1}$ stands for "the inverse Laplace transform of."

The Laplace transform takes a problem given in the *time domain,* where all physical variables are functions of the *real variable t*, into the *complex-frequency domain,* where all physical variables are functions of the complex frequency $s = \sigma + j\omega$, where $j = \sqrt{-1}$ is the imaginary operator. Laplace transform pairs consist of the function $f(t)$ and its transform $F(s)$. Transform pairs can be calculated by substituting $f(t)$ into the defining equation and then evaluating the integral with s held constant. For a transform pair to exist, the corresponding integral must converge, that is,

$$\int_0^\infty |f(t)|e^{-\sigma^* t}\,dt < \infty$$

for some real $\sigma^* > 0$. Signals that are physically realizable always have a Laplace transform.

Tables of Transform Pairs and Transform Properties

Transform pairs for functions commonly encountered in the analysis of dynamic systems rarely need to be calculated. Instead, pairs are determined by reference to a *table of transforms* such as that given in Table 4. In addition, the Laplace transform has a number of properties that are useful in determining the transforms and inverse transforms of functions

Table 4 Laplace Transform Pairs

$F(s)$	$f(t),\ t \geq 0$
1. 1	$\delta(t)$, the unit impulse at $t = 0$
2. $\frac{1}{s}$	1, the unit step
3. $\frac{n!}{s^{n+1}}$	t^n
4. $\frac{1}{s + a}$	e^{-at}
5. $\frac{1}{(s + a)^n}$	$\frac{1}{(n - 1)!} t^{n-1} e^{-at}$
6. $\frac{a}{s(s + a)}$	$1 - e^{-at}$
7. $\frac{1}{(s + a)(s + b)}$	$\frac{1}{b - a} (e^{-at} - e^{-bt})$
8. $\frac{s + p}{(s + a)(s + b)}$	$\frac{1}{b - a} [(p - a)e^{-at} - (p - b)e^{-bt}]$
9. $\frac{1}{(s + a)(s + b)(s + c)}$	$\frac{e^{-at}}{(b - a)(c - a)} + \frac{e^{-bt}}{(c - b)(a - b)} + \frac{e^{-ct}}{(a - c)(b - c)}$
10. $\frac{s + p}{(s + a)(s + b)(s + c)}$	$\frac{(p - a)e^{-at}}{(b - a)(c - a)} + \frac{(p - b)e^{-bt}}{(c - b)(a - b)} + \frac{(p - c)e^{-ct}}{(a - c)(b - c)}$
11. $\frac{b}{s^2 + b^2}$	$\sin bt$
12. $\frac{s}{s^2 + b^2}$	$\cos bt$
13. $\frac{b}{(s + a)^2 + b^2}$	$e^{-at} \sin bt$
14. $\frac{s + a}{(s + a)^2 + b^2}$	$e^{-at} \cos bt$
15. $\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$	$\frac{\omega_n}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_n t} \sin \omega_n \sqrt{1 - \zeta^2}\, t, \quad \zeta < 1$
16. $\frac{\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)}$	$1 + \frac{1}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_n \sqrt{1 - \zeta^2}\, t + \phi)$ $\phi = \tan^{-1} \frac{\sqrt{1 - \zeta^2}}{\zeta} + \pi$ (third quadrant)

in terms of the tabulated pairs. The most important of these are given in a *table of transform properties* such as that given in Table 5.

Poles and Zeros

The response of a dynamic system most often assumes the following form in the complex-frequency domain:

Table 5 Laplace Transform Properties

$f(t)$	$F(s) = \int_0^\infty f(t)e^{-st}\,dt$
1. $af_1(t) + bf_2(t)$	$aF_1(s) + bF_2(s)$
2. $\dfrac{df}{dt}$	$sF(s) - f(0)$
3. $\dfrac{d^2f}{dt^2}$	$s^2F(s) - sf(0) - \left.\dfrac{df}{dt}\right\|_{t=0}$
4. $\dfrac{d^nf}{dt^n}$	$s^nF(s) - \sum_{k=1}^{n} s^{n-k}g_{k-1}$, $g_{k-1} = \left.\dfrac{d^{k-1}f}{dt^{k-1}}\right\|_{t=0}$
5. $\int_0^t f(t)\,dt$	$\dfrac{F(s)}{s} + \dfrac{h(0)}{s}$, $h(0) = \left.\int f(t)\,dt\right\|_{t=0}$
6. $\begin{cases} 0, & t < D \\ f(t-D), & t \geq D \end{cases}$	$e^{-sD}F(s)$
7. $e^{-at}f(t)$	$F(s + a)$
8. $f\left(\dfrac{t}{a}\right)$	$aF(as)$
9. $f(t) = \int_0^t x(t-\tau)y(\tau)\,d\tau = \int_0^t y(t-\tau)x(\tau)\,d\tau$	$F(s) = X(s)Y(s)$
10. $f(\infty) = \lim_{s\to 0} sF(s)$	
11. $f(0+) = \lim_{s\to\infty} sF(s)$	

$$F(s) = \frac{N(s)}{D(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \tag{1}$$

Functions of this form are called *rational functions,* because these are the ratio of two polynomials $N(s)$ and $D(s)$. If $n \geq m$, then $F(s)$ is a *proper rational function;* if $n > m$, then $F(s)$ is a *strictly proper rational function.*

In factored form, the rational function $F(s)$ can be written as

$$F(s) = \frac{N(s)}{D(s)} = \frac{b_m(s - z_1)(s - z_2)\cdots(s - z_m)}{(s - p_1)(s - p_2)\cdots(s - p_n)} \tag{2}$$

The roots of the numerator polynomial $N(s)$ are denoted by z_j, $j = 1, 2, \ldots, m$. These numbers are called the *zeros* of $F(s)$, since $F(z_j) = 0$. The roots of the denominator polynomial are denoted by p_i, $1, 2, \ldots, n$. These numbers are called the *poles* of $F(s)$, since $\lim_{s\to p_i} F(s) = \pm\infty$.

Inversion by Partial-Fraction Expansion

The *partial-fraction expansion theorem* states that a strictly proper rational function $F(s)$ with *distinct* (*nonrepeated*) *poles* p_i, $i = 1, 2, \ldots, n$, can be written as the sum

$$F(s) = \frac{A_1}{s - p_1} + \frac{A_2}{s - p_2} + \cdots + \frac{A_n}{s - p_n} = \sum_{i=1}^{n} A_i \left(\frac{1}{s - p_i} \right) \tag{3}$$

where the A_i, $i = 1, 2, \ldots, n$, are constants called *residues.* The inverse transform of $F(s)$ has the simple form

$$f(t) = A_1 e^{p_1 t} + A_2 e^{p_2 t} + \cdots + A_n e^{p_n t} = \sum_{i=1}^{n} A_i e^{p_i t}$$

The *Heaviside expansion theorem* gives the following expression for calculating the residue at the pole p_i,

$$A_i = (s - p_i) F(s)\big|_{s=p_i} \qquad \text{for } i = 1, 2, \ldots, n$$

These values can be checked by substituting into Eq. (3), combining the terms on the right-hand side of Eq. (3), and showing that the result yields the values for all the coefficients b_j, $j = 1, 2, \ldots, m$, originally specified in the form of Eq. (3).

Repeated Poles

When two or more poles of a strictly proper rational function are identical, the poles are said to be *repeated* or *nondistinct.* If a pole is repeated q times, that is, if $p_i = p_{i+1} = \cdots = p_{i+q-1}$, then the pole is said to be of *multiplicity* q. A strictly proper rational function with a pole of multiplicity q will contain q terms of the form

$$\frac{A_{i1}}{(s - p_i)^q} + \frac{A_{i2}}{(s - p_i)^{q-1}} + \cdots + \frac{A_{iq}}{s - p_i}$$

in addition to the terms associated with the distinct poles. The corresponding terms in the inverse transform are

$$\left(\frac{1}{(q-1)!} A_{i1} t^{(q-1)} + \frac{1}{(q-2)!} A_{i2} t^{(q-2)} + \cdots + A_{iq} \right) e^{p_i t}$$

The corresponding residues are

$$A_{i1} = (s - p_i)^q F(s)\big|_{s=p_i}$$

$$A_{i2} = \left(\frac{d}{ds} [(s - p_i)^q F(s)] \right) \Bigg|_{s=p_i}$$

$$\vdots$$

$$A_{iq} = \frac{1}{(q-1)!} \left(\frac{d^{(q-1)}}{ds^{(q-1)}} [(s - p_i)^q F(s)] \right) \Bigg|_{s=p_i}$$

Complex Poles

A strictly proper rational function with complex-conjugate poles can be inverted using partial-fraction expansion. Using a method called *completing the square,* however, is almost always easier. Consider the function

$$F(s) = \frac{B_1 s + B_2}{(s + \sigma - j\omega)(s + \sigma + j\omega)}$$

$$= \frac{B_1 s + B_2}{s^2 + 2\sigma s + \sigma^2 + \omega^2}$$

$$= \frac{B_1 s + B_2}{(s + \sigma)^2 + \omega_2}$$

From the transform tables the Laplace inverse is

$$f(t) = e^{-\sigma t}[B_1 \cos \omega t + B_3 \sin \omega t]$$

$$= Ke^{-\sigma t} \cos(\omega t + \phi)$$

where $B_3 = (1/\omega)(B_2 - aB_1)$
$K = \sqrt{B_1^2 + B_3^2}$
$\phi = -\tan^{-1}(B_3/B_1)$

Proper and Improper Rational Functions

If $F(s)$ is not a strictly proper rational function, then $N(s)$ must be divided by $D(s)$ using *synthetic division.* The result is

$$F(s) = \frac{N(s)}{D(s)} = P(s) + \frac{N^*(s)}{D(s)}$$

where $P(s)$ is a polynomial of degree $m - n$ and $N^*(s)$ is a polynomial of degree $n - 1$. Each term of $P(s)$ may be inverted directly using the transform tables. The strictly proper rational function $N^*(s)/D(s)$ may be inverted using partial-fraction expansion.

Initial-Value and Final-Value Theorems

The limits of $f(t)$ as time approaches zero or infinity frequently can be determined directly from the transform $F(s)$ without inverting. The *initial-value theorem* states that

$$f(0_+) = \lim_{s \to \infty} sF(s)$$

where the limit exists. If the limit does not exist (i.e., is infinite), the value of $f(0_+)$ is undefined. The *final-value theorem* states that

$$f(\infty) = \lim_{s \to 0} sF(s)$$

provided that (with the possible exception of a single pole at $s = 0$) $F(s)$ has no poles with nonnegative real parts.

Transfer Functions

The Laplace transform of the system I/O equation may be written in terms of the transform $Y(s)$ of the system response $y(t)$ as

$$Y(s) = \frac{G(s)N(s) + F(s)D(s)}{P(s)D(s)}$$

$$= \left(\frac{G(s)}{P(s)}\right)\left(\frac{N(s)}{D(s)}\right) + \frac{F(s)}{P(s)}$$

where (a) $P(s) = a_n s^n + a_{n-1} + \cdots + a_1 s + a_0$ is the *characteristic polynomial* of the system

(b) $G(s) = b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0$ represents the *numerator dynamics* of the system

(c) $U(s) = N(s)/D(s)$ is the transform of the input to the system, $u(t)$, assumed to be a rational function

(d) $$F(s) = a_n y(0) s^{n-1} + \left(a_n \frac{dy}{dt}(0) + a_{n-1} y(0) \right) s^{n-2} + \cdots + \left(a_n \frac{d^{n-1} y}{dt^{n-1}}(0) + a_{n-1} \frac{d^{n-2} y}{dt}(0) + \cdots + a_1 y(0) \right)$$

reflects the initial system state [i.e., the initial conditions on $y(t)$ and its first $n - 1$ derivatives]

The transformed response can be thought of as the sum of two components,

$$Y(s) = Y_{zs}(s) + Y_{zi}(s)$$

where (e) $Y_{zs}(s) = [G(s)/P(s)][N(s)/D(s)] = H(s)U(s)$ is the transform of the *zero-state response,* that is, the response of the system to the input alone

(f) $Y_{zi}(s) = F(s)/P(s)$ is the transform of the *zero-input response,* that is, the response of the system to the initial state alone

The rational function

(g) $H(s) = Y_{zs}(s)/U(s) = G(s)/P(s)$ is the *transfer function* of the system, defined as the Laplace transform of the ratio of the system response to the system input, assuming zero initial conditions

The transfer function plays a crucial role in the analysis of fixed linear systems using transforms and can be written directly from knowledge of the system I/O equation as

$$H(s) = \frac{b_m s^m + \cdots + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}$$

Impulse Response

Since $U(s) = 1$ for a unit impulse function, the transform of the zero-state response to a unit impulse input is given by the relation (g) as

$$Y_{zs}(s) = H(s)$$

that is, the system transfer function. In the time domain, therefore, the unit *impulse response* is

$$h(t) = \begin{cases} 0 & \text{for } t \leq 0 \\ \mathcal{L}^{-1}[H(s)] & \text{for } t > 0 \end{cases}$$

This simple relationship is profound for several reasons. First, this provides for a direct characterization of time-domain response $h(t)$ in terms of the properties (poles and zeros) of the rational function $H(s)$ in the complex-frequency domain. Second, applying the convolution transform pair (Table 5) to relation (e) above yields

$$Y_{zs}(t) = \int_0^t h(\tau) u(t - \tau)\, d\tau$$

In words, the zero-state output corresponding to an arbitrary input $u(t)$ can be determined by convolution with the impulse response $h(t)$. In other words, the impulse response completely characterizes the system. The impulse response is also called the system *weighing function.*

Block Diagrams

Block diagrams are an important conceptual tool for the analysis and design of dynamic systems, because block diagrams provide a graphic means for depicting the relationships among system variables and components. A block diagram consists of unidirectional blocks representing specified system components or subsystems interconnected by arrows representing system variables. Causality follows in the direction of the arrows, as in Fig. 11, indicating that the output is caused by the input acting on the system defined in the block.

Combining transform variables, transfer functions, and block diagrams provides a powerful graphical means for determining the overall transfer function of a system when the transfer functions of its component subsystems are known. The basic blocks in such diagrams are given in Fig. 12. A block diagram comprising many blocks and summers can be reduced to a single transfer function block by using the diagram transformations given in Fig. 13.

5.2 Transient Analysis Using Transform Methods

Basic to the study of dynamic systems are the concepts and terminology used to characterize system behavior or performance. These ideas are aids in *defining* behavior in order to consider for a given context those features of behavior which are desirable and undesirable; in *describing* behavior in order to communicate concisely and unambiguously various behavioral attributes of a given system; and in *specifying* behavior in order to formulate desired behavioral norms for system design. Characterization of dynamic behavior in terms of standard concepts also leads in many cases to analytical shortcuts, since key features of the system response frequently can be determined without actually solving the system model.

Parts of the Complete Response

A variety of names are used to identify terms in the response of a fixed linear system. The complete response of a system may be thought of alternatively as the sum of the following:

1. The *free response* (or complementary or homogeneous solution) and the *forced response* (or particular solution). The free response represents the natural response of a system when inputs are removed and the system responds to some initial stored energy. The forced response of the system depends on the form of the input only.
2. The *transient response* and the *steady-state response.* The transient response is that part of the output that decays to zero as time progresses. The steady-state response is that part of the output that remains after all the transients disappear.

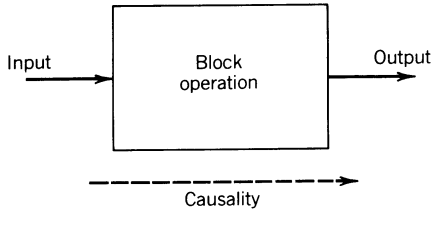


Figure 11 Basic block diagram, showing assumed direction of causality or loading.

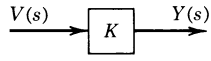
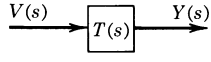
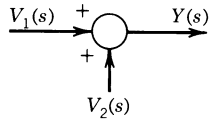
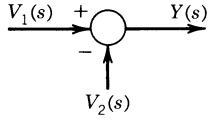
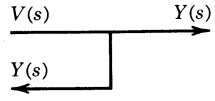
Type	Input–Output Relations: Time Domain	Input–Output Relations: Transform Domain	Symbol
(*a*) Multiplier	$y(t) = Kv(t)$	$Y(s) = KV(s)$	
(*b*) General transfer function	$y(t) = \mathcal{L}^{-1}[T(s)V(s)]$	$Y(s) = T(s)V(s)$	
(*c*) Summer	$y(t) = v_1(t) + v_2(t)$	$Y(s) = V_1(s) + V_2(s)$	
(*d*) Comparator	$y(t) = v_1(t) - v_2(t)$	$Y(s) = V_1(s) - V_2(s)$	
(*e*) Takeoff point	$y(t) = v(t)$	$Y(s) = V(s)$	

Figure 12 Basic block diagram elements.[4]

3. The *zero-state response* and the *zero-input response.* The zero-state response is the complete response (both free and forced responses) to the input when the initial state is zero. The zero-input response is the complete response of the system to the initial state when the input is zero.

Test Inputs or Singularity Functions

For a stable system, the response to a specific input signal will provide several measures of system performance. Since the actual inputs to a system are not usually known a priori, characterization of the system behavior is generally given in terms of the response to one of a standard set of *test input signals.* This approach provides a common basis for the comparison of different systems. In addition, many inputs actually encountered can be approximated by some combination of standard inputs. The most commonly used test inputs are members of the family of *singularity functions,* depicted in Fig. 14.

First-Order Transient Response

The standard form of the I/O equation for a first-order system is

$$\frac{dy}{dt} + \frac{1}{\tau}y(t) = \frac{1}{\tau}u(t)$$

where the parameter τ is called the system *time constant.* The response of this standard first-order system to three test inputs is depicted in Fig. 15, assuming zero initial conditions on the output $y(t)$. For all inputs, it is clear that the response approaches its steady state mon-

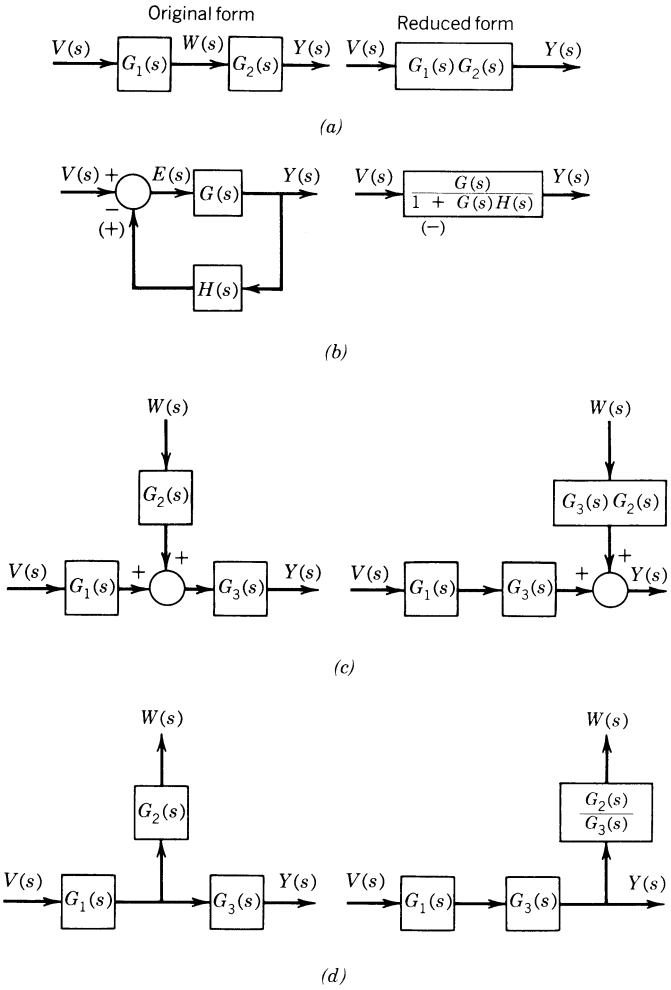


Figure 13 Representative block diagram transformations: (*a*) series or cascaded elements, (*b*) feedback loop, (*c*) relocated summer, (*d*) relocated takeoff point.[4]

otonically (i.e., without oscillations) and that the *speed of response* is completely characterized by the time constant τ. The transfer function of the system is

$$H(s) = \frac{Y(s)}{U(s)} = \frac{1/\tau}{s + 1/\tau}$$

and therefore $\tau = -p^{-1}$, where p is the system pole. As the absolute value of p increases, τ decreases and the response becomes faster.

The response of the standard first-order system to a step input of magnitude u for arbitrary initial condition $y(0) = y_0$ is

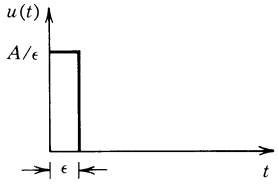
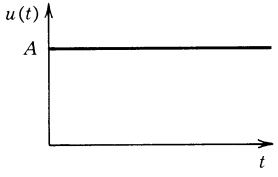
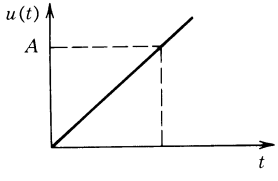
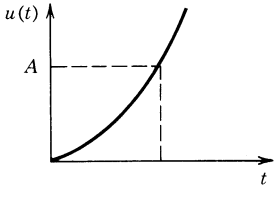
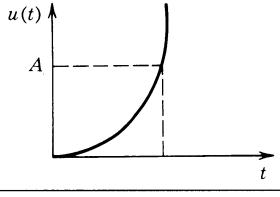
Function	Graph	$u(t)$	$U(s)$
Impulse		$u(t) = \begin{cases} \lim_{\epsilon \to 0} A/\epsilon, & 0_+ < t < \epsilon \\ 0, & \text{otherwise} \end{cases}$	$U(s) = A$
Step		$u(t) = \begin{cases} A, & 0_+ < t \\ 0, & \text{otherwise} \end{cases}$	$U(s) = \dfrac{A}{s}$
Ramp		$u(t) = \begin{cases} At, & 0_+ < t \\ 0, & \text{otherwise} \end{cases}$	$U(s) = \dfrac{A}{s^2}$
Parabolic		$u(t) = \begin{cases} At^2, & 0_+ < t \\ 0, & \text{otherwise} \end{cases}$	$U(s) = \dfrac{2A}{s^3}$
General		$u(t) = \begin{cases} At^n, & 0_+ < t \\ 0, & \text{otherwise} \end{cases}$	$U(s) = \dfrac{n!A}{s^{n+1}}$

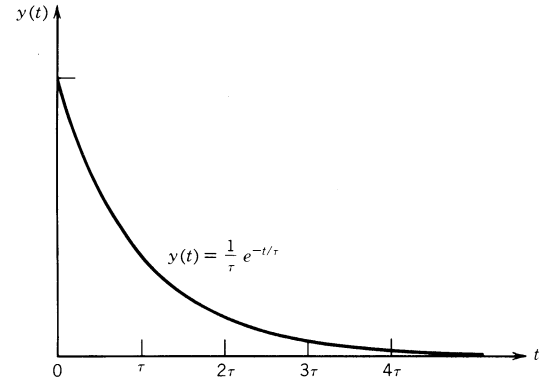
Figure 14 Family of singularity functions commonly used as test inputs.

$$y(t) = y_{ss} - [y_{ss} - y_0]e^{-t/\tau}$$

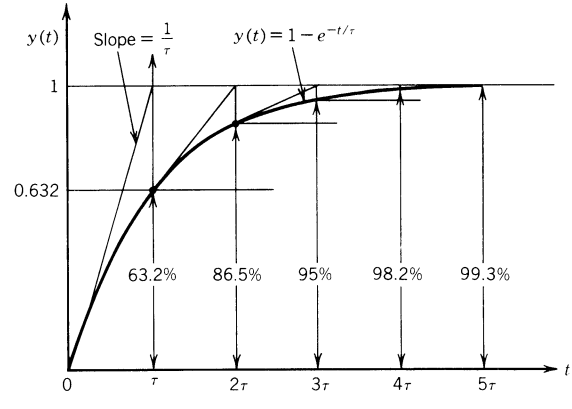
where $y_{ss} = u$ is the steady-state response. Table 6 and Fig. 16 record the values of $y(t)$ and $\dot{y}(t)$ for $t = k\tau$, $k = 0, 1, \ldots, 6$. Note that over any time interval of duration τ, the response increases approximately 63% of the difference between the steady-state value and the value at the beginning of the time interval, that is,

$$y(t + \tau) - y(t) \approx 0.63212[y_{ss} - y(t)]$$

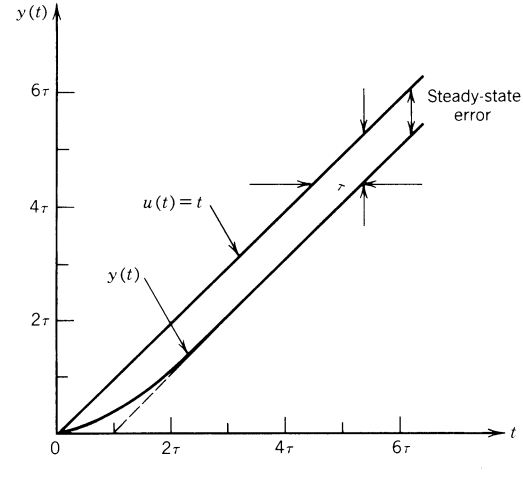
Note also that the slope of the response at the beginning of any time interval of duration τ intersects the steady-state value y_{ss} at the end of the interval, that is,



(a)



(b)



(c)

Figure 15 Response of a first-order system to (*a*) unit impulse, (*b*) unit step, and (*c*) unit ramp inputs.

Table 6 Tabulated Values of the Response of a First-Order System to a Unit Step Input

t	$y(t)$	$\dot{y}(t)$
0	0	τ^{-1}
τ	0.632	$0.368\tau^{-1}$
2τ	0.865	$0.135\tau^{-1}$
3τ	0.950	$0.050\tau^{-1}$
4τ	0.982	$0.018\tau^{-1}$
5τ	0.993	$0.007\tau^{-1}$
6τ	0.998	$0.002\tau^{-1}$

$$\frac{dy}{dt}(t) = \frac{y_{ss} - y(t)}{\tau}$$

Finally, note that after an interval of four time constants, the response is within 98% of the steady-state value, that is,

$$y(4\tau) \approx 0.98168(y_{ss} - y_0)$$

For this reason, $T_s = 4\tau$ is called the (2%) *setting time.*

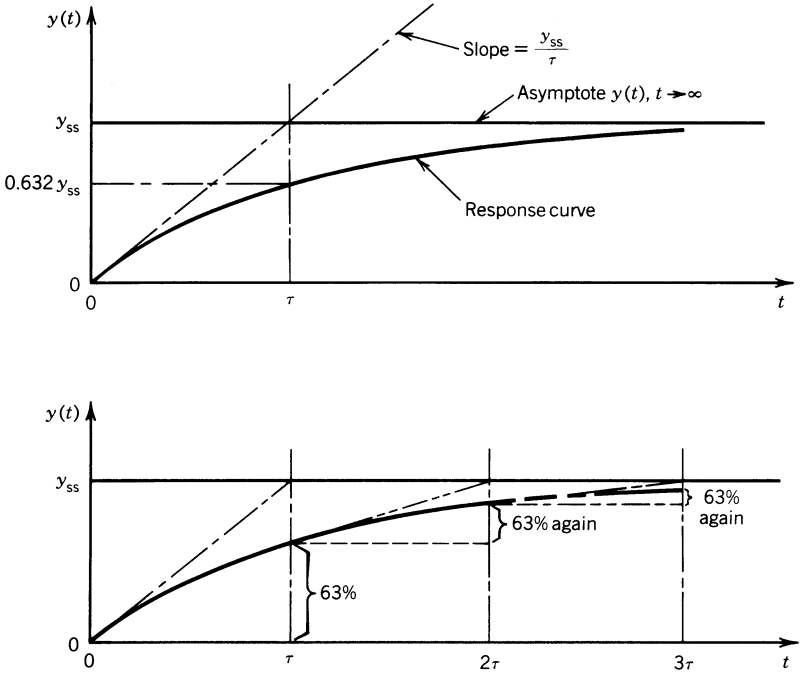


Figure 16 Response of a first-order system to a unit step input, showing the relationship to the time constant.

Second-Order Transient Response

The standard form of the I/O equation for a second-order system is

$$\frac{d^2y}{dt^2} + 2\zeta\omega_n \frac{dy}{dt} + \omega_n^2 y(t) = \omega_n^2 u(t)$$

with transfer function

$$H(s) = \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The system poles are obtained by applying the quadratic formula to the characteristic equation as

$$p_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$$

where the following parameters are defined: ζ is the *damping ratio,* ω_n is the *natural frequency,* and $\omega_d = \omega_n\sqrt{1 - \zeta^2}$ is the *damped natural frequency.*

The nature of the response of the standard second-order system to a step input depends on the value of the damping ratio, as depicted in Fig. 17. For a stable system, four classes of response are defined.

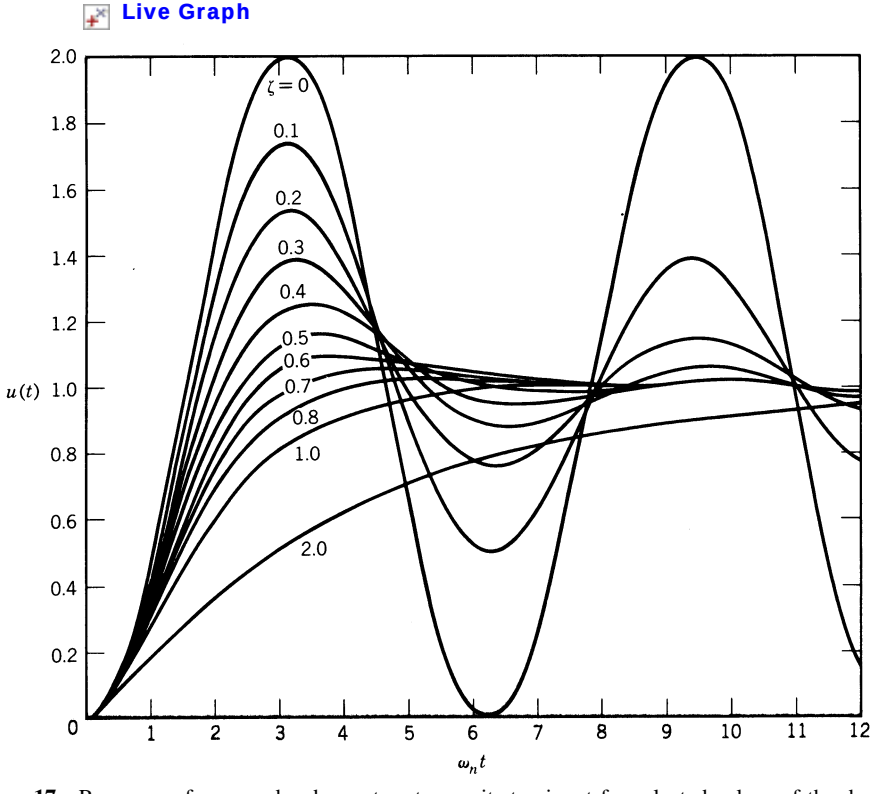


Figure 17 Response of a second-order system to a unit step input for selected values of the damping ratio.

1. *Overdamped Response* ($\zeta > 1$). The system poles are real and distinct. The response of the second-order system can be decomposed into the response of two cascaded first-order systems, as shown in Fig. 18.
2. *Critically Damped Response* ($\zeta = 1$). The system poles are real and repeated. This is the limiting case of overdamped response, where the response is as fast as possible without overshoot.
3. *Underdamped Response* ($1 > \zeta > 0$). The system poles are complex conjugates. The response oscillates at the damped frequency ω_d. The magnitude of the oscillations and the speed with which the oscillations decay depend on the damping ratio ζ.
4. *Harmonic Oscillation* ($\zeta = 0$). The system poles are pure imaginary numbers. The response oscillates at the natural frequency ω_n and the oscillations are undamped (i.e., the oscillations are sustained and do not decay).

The Complex s-Plane

The location of the system poles (roots of the characteristic equation) in the *complex s-plane* reveals the nature of the system response to test inputs. Figure 19 shows the relationship between the location of the poles in the complex plane and the parameters of the standard second-order system. Figure 20 shows the unit impulse response of a second-order system corresponding to various pole locations in the complex plane.

Transient Response of Higher Order Systems

The response of third- and higher order systems to test inputs is simply the sum of terms representing component first- and second-order responses. This is because the system poles must either be real, resulting in first-order terms, or complex, resulting in second-order underdamped terms. Furthermore, because the transients associated with those system poles having the largest real part decay the most slowly, these transients tend to dominate the output. The response of higher order systems therefore tends to have the same form as the response to the *dominant poles,* with the response to the *subdominant poles* superimposed over it. Note that the larger the relative difference between the real parts of the dominant and subdominant poles, the more the output tends to resemble the dominant mode of response.

For example, consider a fixed linear third-order system. The system has three poles. Either the poles may all be real or one may be real while the other pair is complex conjugate. This leads to the three forms of step response shown in Fig. 21, depending on the relative locations of the poles in the complex plane.

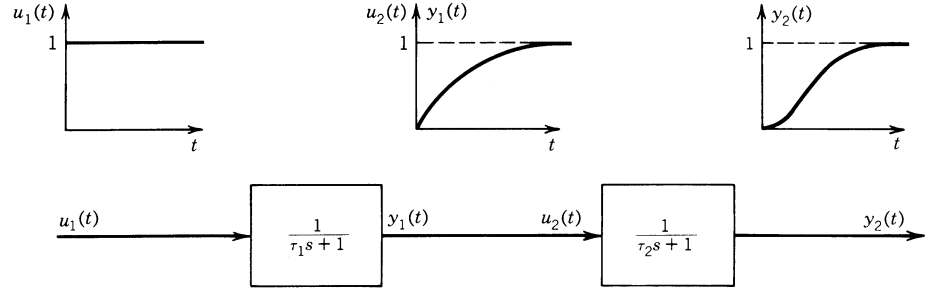


Figure 18 Overdamped response of a second-order system decomposed into the responses of two first-order systems.

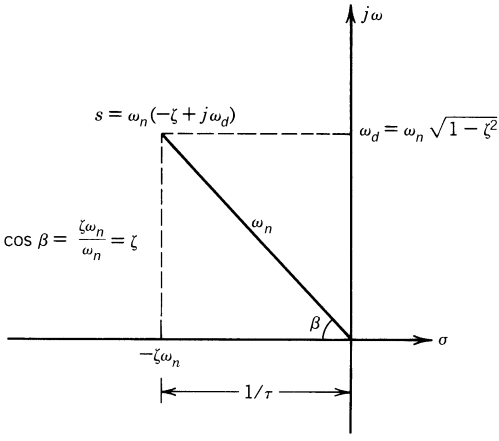


Figure 19 Location of the upper complex pole in the *s*-plane in terms of the parameters of the standard second-order system.

Transient Performance Measures

The transient response of a system is commonly described in terms of the measures defined in Table 7 and shown in Fig. 22. While these measures apply to any output, for a second-order system these can be calculated exactly in terms of the damping ratio and natural frequency, as shown in column 3 of the table. A common practice in control system design is to determine an initial design with dominant second-order poles that satisfy the performance specifications. Such a design can easily be calculated and then modified as necessary to achieve the desired performance.

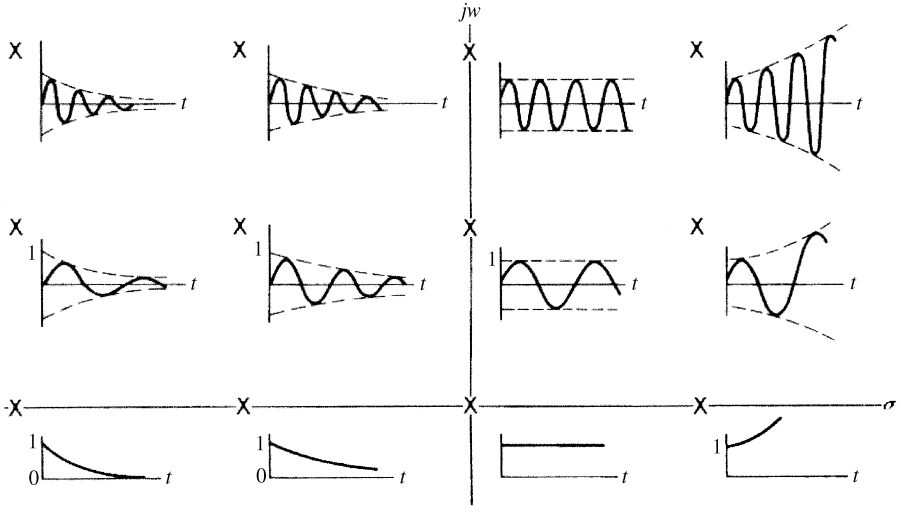


Figure 20 Unit impulse response for selected upper complex pole locations in the *s*-plane.

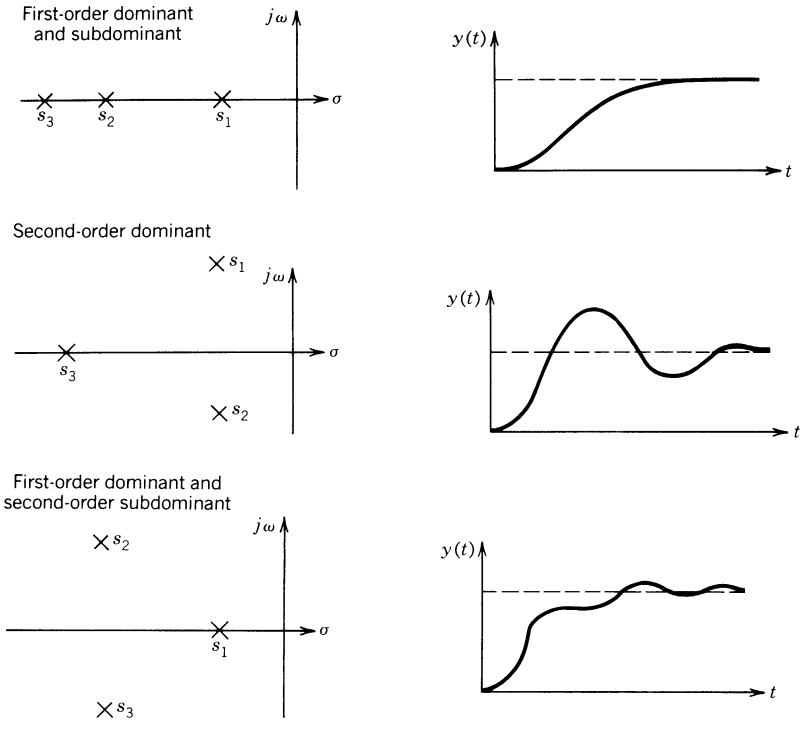


Figure 21 Step response of a third-order system for alternative upper complex pole locations in the s-plane.

The Effect of Zeros on the Transient Response

Zeros arise in a system transfer function through the inclusion of one or more derivatives of $u(t)$ among the inputs to the system. By sensing the rate(s) of change of $u(t)$, the system in effect *anticipates* the future values of $u(t)$. This tends to increase the speed of response of the system relative to the input $u(t)$.

The effect of a zero is greatest on the modes of response associated with neighboring poles. For example, consider the second-order system represented by the transfer function

$$H(s) = K\frac{s - z}{(s - p_1)(s - p_2)}$$

If $z = p_1$, then the system responds as a first-order system with $\tau = -p_2^{-1}$; whereas if $z = p_2$, then the system responds as a first-order system with $\tau = -p_1^{-1}$. Such *pole–zero cancellation* can only be achieved mathematically, but it can be approximated in physical systems. Note that by diminishing the residue associated with the response mode having the larger time constant, the system responds more quickly to changes in the input, confirming our earlier observation.

5.3 Response to Periodic Inputs Using Transform Methods

The response of a dynamic system to periodic inputs can be a critical concern to the control engineer. An input $u(t)$ is *periodic* if $u(t + T) = u(t)$ for all time t, where T is a constant

Table 7 Transient Performance Measures Based on Step Response

Performance Measure	Definition	Formula for a Second-Order System
Delay time, t_d	Time required for the response to reach half the final value for the first time	
10–90% rise time, t_r	Time required for the response to rise from 10 to 90% of the final response (used for overdamped responses)	
0–100% rise time, t_r	Time required for the response to rise from 0 to 100% of the final response (used for underdamped responses)	$t_r = \dfrac{\pi - \beta}{\omega_d}$ where $\beta = \cos^{-1} \zeta$
Peak time, t_p	Time required for the response to reach the first peak of the overshoot	$t_p = \dfrac{\pi}{\omega_d}$
Maximum overshoot, M_p	The difference in the response between the first peak of the overshoot and the final response	$M_p = e^{-\zeta\pi/\sqrt{1-\zeta^2}}$
Percent overshoot, PO	The ratio of maximum overshoot to the final response expressed as a percentage	$\text{PO} = 100e^{-\zeta\pi/\sqrt{1-\zeta^2}}$
Setting time, t_s	The time required for the response to reach and stay within a specified band centered on the final response (usually 2% or 5% of final response band)	$t_s = \dfrac{4}{\zeta\omega_n}$ (2% band) $t_s = \dfrac{3}{\zeta\omega_n}$ (5% band)

called the period. Periodic inputs are important because these are ubiquitous: rotating unbalanced machinery, reciprocating pumps and engines, ac electrical power, and a legion of noise and disturbance inputs can be approximated by periodic inputs. Sinusoids are the most important category of periodic inputs, because these are frequently occurring and easily analyzed and form the basis for analysis of general periodic inputs.

Frequency Response

The *frequency response* of a system is the steady-state response of the system to a sinusoidal input. For a linear system, the frequency response has the unique property that the response is a sinusoid of the same frequency as the input sinusoid, differing only in amplitude and phase. In addition, it is easy to show that the amplitude and phase of the response are functions of the input frequency, which are readily obtained from the system transfer function.

Consider a system defined by the transfer function $H(s)$. For an input

$$u(t) = A \sin \omega t$$

the corresponding steady-state output is

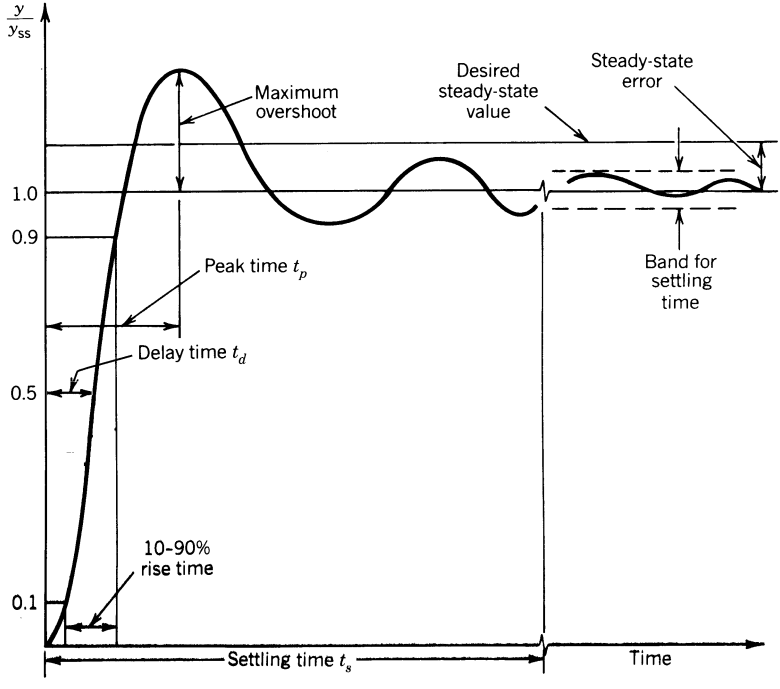


Figure 22 Transient performance measures based on step response.

$$y_{ss}(t) = AM(\omega)\sin[\omega t + \phi(\omega)]$$

where $M(\omega) = H(j\omega)|$ is called the *magnitude ratio*
$\phi(\omega) = \angle H(j\omega)$ is called the *phase angle*
$H(j\omega) = H(s)|_{s=j\omega}$ is called the *frequency transfer function*

The frequency transfer function is obtained by substituting $j\omega$ for s in the transfer function $H(s)$. If the complex quantity $H(j\omega)$ is written in terms of its real and imaginary parts as $H(j\omega) = \mathrm{Re}(\omega) + j\,\mathrm{Im}(\omega)$, then

$$M(\omega) = [\mathrm{Re}(\omega)^2 + \mathrm{Im}(\omega)^2]^{1/2}$$

$$\phi(\omega) = \tan^{-1}\left[\frac{\mathrm{Im}(\omega)}{\mathrm{Re}(\omega)}\right]$$

and in polar form

$$H(j\omega) = M(\omega)e^{j\phi(\omega)}$$

Frequency Response Plots

The frequency response of a fixed linear system is typically represented graphically using one of three types of frequency response plots. A *polar plot* is simply a plot of the vector $H(j\omega)$ in the complex plane, where $\mathrm{Re}(\omega)$ is the abscissa and $\mathrm{Im}(\omega)$ is the ordinate. A *logarithmic plot* or *Bode diagram* consists of two displays: (1) the magnitude ratio in decibels $M_{dB}(\omega)$ [where $M_{dB}(\omega) = 20 \log M(\omega)$] versus $\log \omega$ and (2) the phase angle in degrees

$\phi(\omega)$ versus log ω. Bode diagrams for normalized first- and second-order systems are given in Fig. 23. Bode diagrams for higher order systems are obtained by adding these first- and second-order terms, appropriately scaled. A *Nichols diagram* can be obtained by cross plotting the Bode magnitude and phase diagrams, eliminating log ω. Polar plots and Bode and Nichols diagrams for common transfer functions are given in Table 8.

Frequency Response Performance Measures

Frequency response plots show that dynamic systems tend to behave like *filters,* "passing" or even amplifying certain ranges of input frequencies while blocking or attenuating other frequency ranges. The range of frequencies for which the amplitude ratio is no less than 3 dB of its maximum value is called the *bandwidth* of the system. The bandwidth is defined by upper and lower *cutoff frequencies* ω_c, or by $\omega = 0$ and an upper cutoff frequency if $M(0)$ is the maximum amplitude ratio. Although the choice of "down 3 dB" used to define the cutoff frequencies is somewhat arbitrary, the bandwidth is usually taken to be a measure of the range of frequencies for which a significant portion of the input is felt in the system output. The bandwidth is also taken to be a measure of the system speed of response, since attenuation of inputs in the higher frequency ranges generally results from the inability of the system to "follow" rapid changes in amplitude. Thus, a narrow bandwidth generally indicates a sluggish system response.

Response to General Periodic Inputs

The *Fourier series* provides a means for representing a general periodic input as the sum of a constant and terms containing sine and cosine. For this reason the *Fourier series,* together with the superposition principle for linear systems, extends the results of frequency response analysis to the general case of arbitrary periodic inputs. The Fourier series representation of a periodic function $f(t)$ with period $2T$ on the interval $t^* + 2T \geq t \geq t^*$ is

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi t}{T} + b_n \sin \frac{n\pi t}{T} \right)$$

where

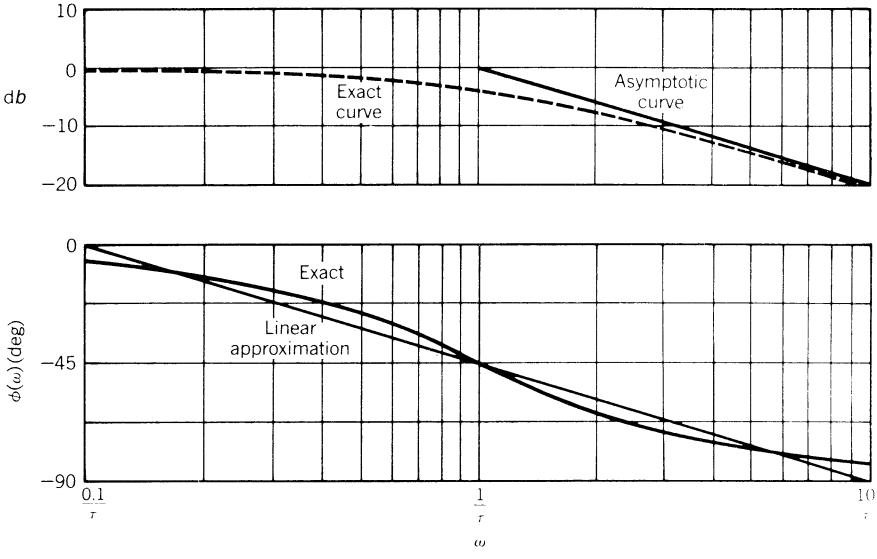
$$a_n = \frac{1}{T} \int_{t^*}^{t^*+2T} f(t) \cos \frac{n\pi t}{T} \, dt$$

$$b_n = \frac{1}{T} \int_{t^*}^{t^*+2T} f(t) \sin \frac{n\pi t}{T} \, dt$$

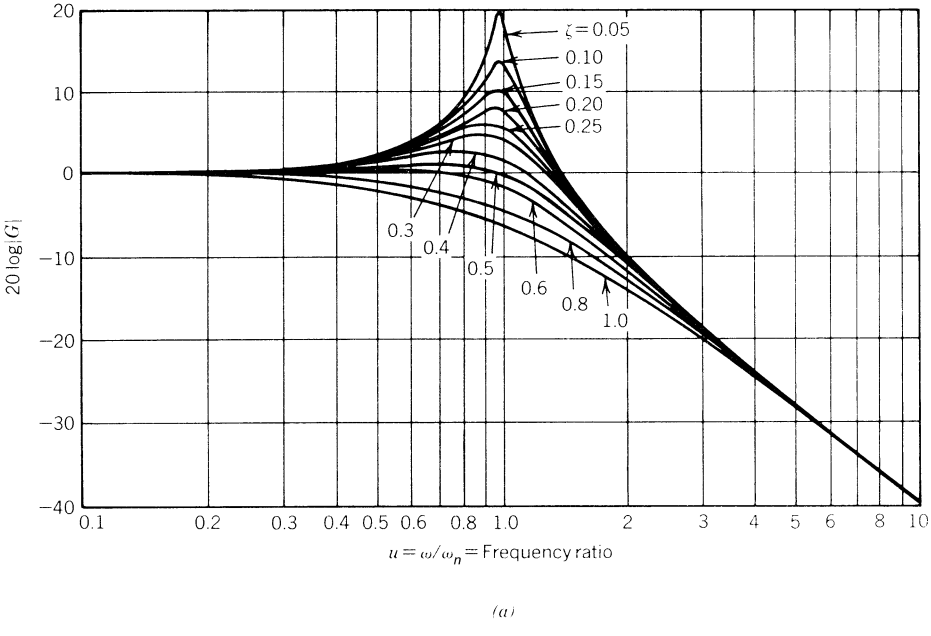
If $f(t)$ is defined outside the specified interval by a periodic extension of period $2T$ and if $f(t)$ and its first derivative are piecewise continuous, then the series converges to $f(t)$ if t is a point of continuity or to $\frac{1}{2}[f(t_+) + f(t_-)]$ if t is a point of discontinuity. Note that while the Fourier series in general is infinite, the notion of bandwidth can be used to reduce the number of terms required for a reasonable approximation.

6 STATE-VARIABLE METHODS

State-variable methods use the vector state and output equations introduced in Section 4 for analysis of dynamic systems directly in the time domain. These methods have several advantages over transform methods. First, state-variable methods are particularly advantageous for the study of multivariable (multiple-input/multiple-output) systems. Second, state-variable methods are more naturally extended for the study of linear time-varying and non-

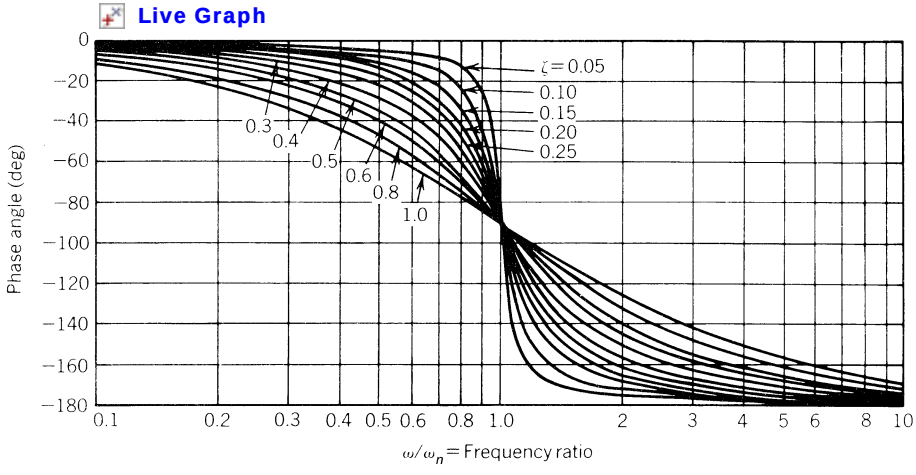


Live Graph



(a)

Figure 23 Bode diagrams for normalized (*a*) first-order and (*b*) second-order systems.



(b)

Figure 23 (*Continued*)

linear systems. Finally, state-variable methods are readily adapted to computer simulation studies.

6.1 Solution of the State Equation

Consider the vector equation of state for a fixed linear system:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

The solution to this system is

$$x(t) = \Phi(t)x(0) + \int_0^t \Phi(t - \tau)Bu(\tau)\, d\tau$$

where the matrix $\Phi(t)$ is called the *state transition matrix.* The state transition matrix represents the free response of the system and is defined by the matrix exponential series

$$\Phi(t) = e^{At} = I + At + \frac{1}{2!}A^2t^2 + \cdots = \sum_{k=0}^{\infty} \frac{1}{k!}A^kt^k$$

where I is the identity matrix. The state transition matrix has the following useful properties:

$$\Phi(0) = I$$

$$\Phi^{-1}(t) = \Phi(-t)$$

$$\Phi^k(t) = \Phi(kt)$$

$$\Phi(t_1 + t_2) = \Phi(t_1)\Phi(t_2)$$

$$\Phi(t_2 - t_1)\Phi(t_1 - t_0) = \Phi(t_2 - t_0)$$

$$\Phi(t) = A\Phi(t)$$

The Laplace transform of the state equation is

Table 8 Transfer Function Plots for Representative Transfer Functions[5]

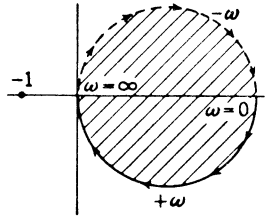
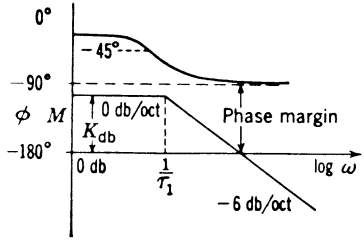
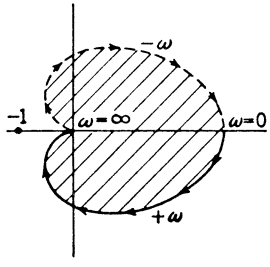
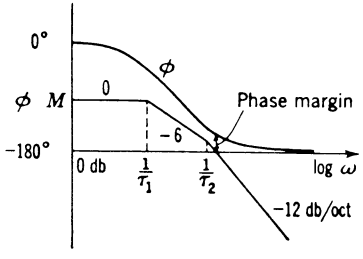
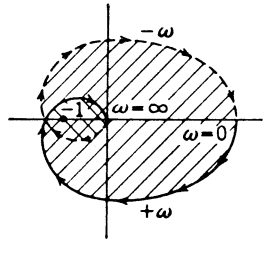
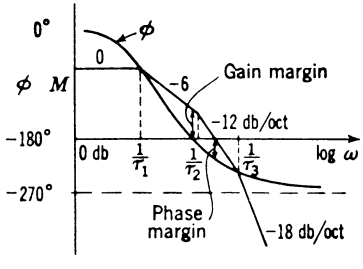
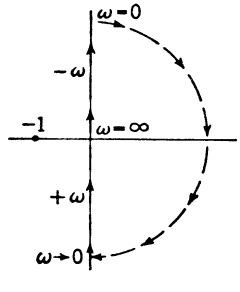
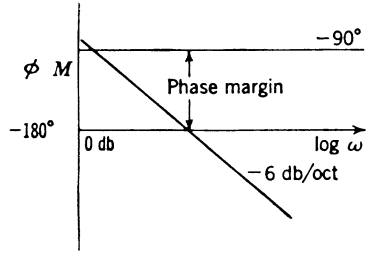
$G(s)$	Polar Plot	Bode Diagram
1. $\frac{K}{s\tau_1 + 1}$		
2. $\frac{K}{(s\tau_1 + 1)\ (s\tau_2 + 1)}$		
3. $\frac{K}{(s\tau_1 + 1)\ (s\tau_2 + 1)\ (s\tau_3 + 1)}$		
4. $\frac{K}{s}$		

Table 8 (*Continued*)

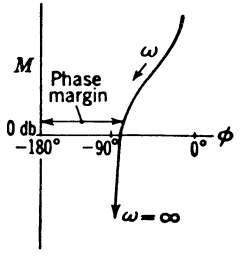
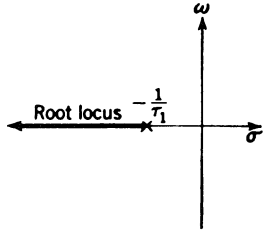
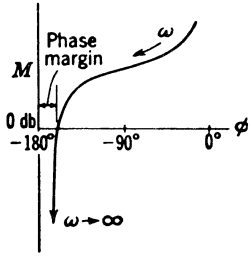
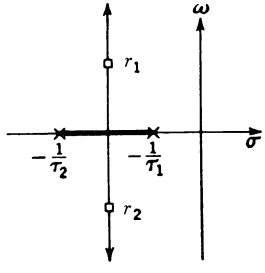
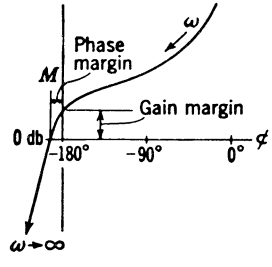
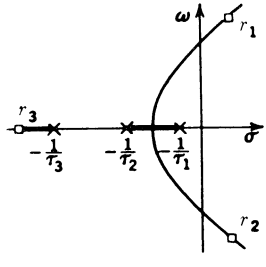
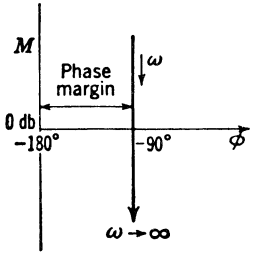
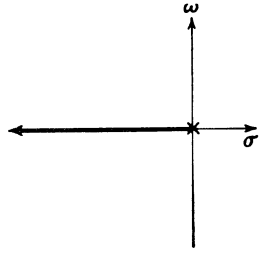
Nichols Diagram	Root Locus	Comments
		Stable; gain margin = ∞
		Elementary regulator; stable; gain margin = ∞
		Regulator with additional energy storage component; unstable, but can be made stable by reducing gain
		Ideal integrator; stable

Table 8 (*Continued*)

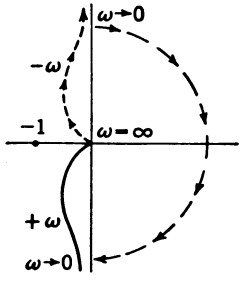
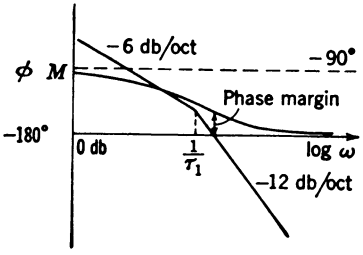
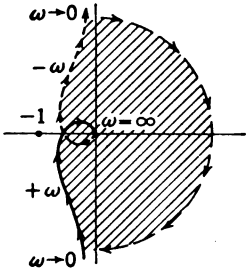
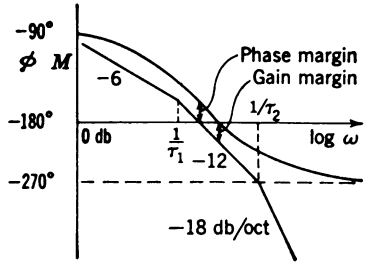
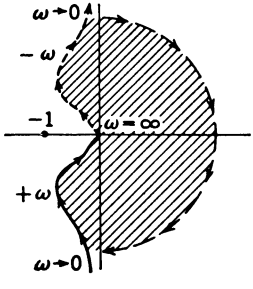
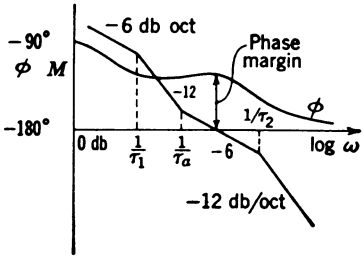
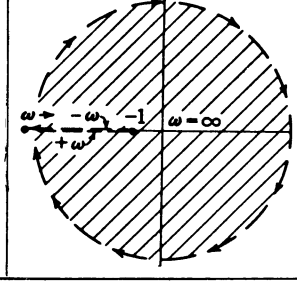
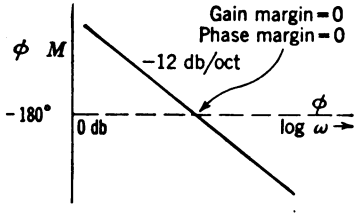
$G(s)$	Polar Plot	Bode Diagram
5. $\dfrac{K}{s(s\tau_1+1)}$		
6. $\dfrac{K}{s(s\tau_1+1)(s\tau_2+1)}$		
7. $\dfrac{K(s\tau_a+1)}{s(s\tau_1+1)(s\tau_2+1)}$		
8. $\dfrac{K}{s^2}$		

Table 8 (*Continued*)

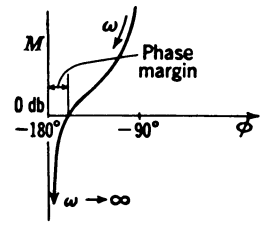
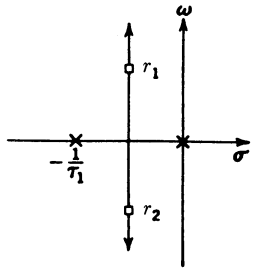
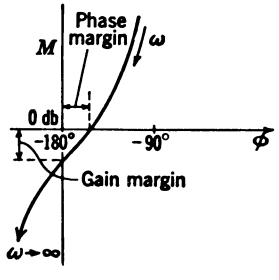
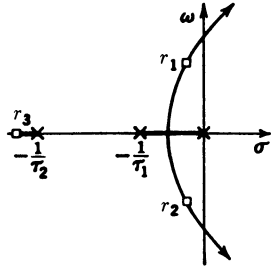
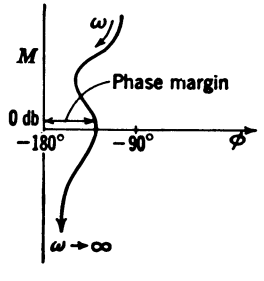
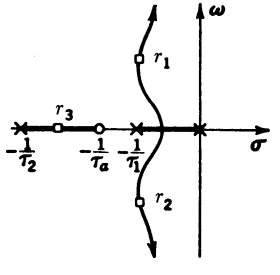
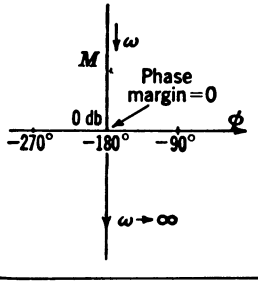
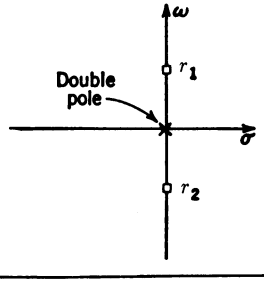
Nichols Diagram	Root Locus	Comments
		Elementary instrument servo; inherently stable; gain margin = ∞
		Instrument servo with field control motor or power servo with elementary Ward–Leonard drive; stable as shown but may become unstable with increased gain
		Elementary instrument servo with phase lead (derivative) compensator; stable
		Inherently unstable; must be compensated

Table 8 (*Continued*)

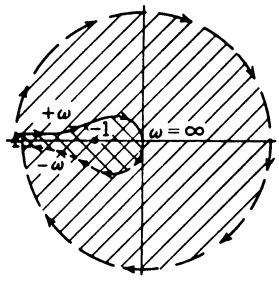
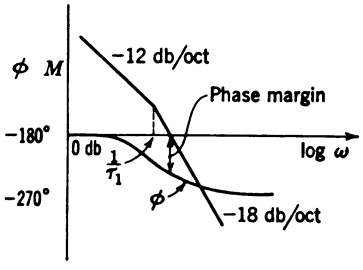
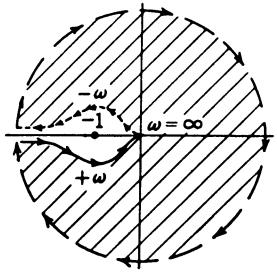
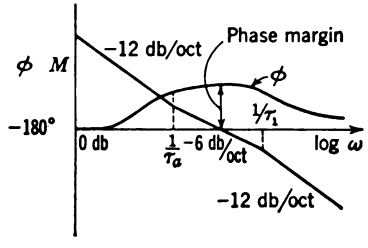
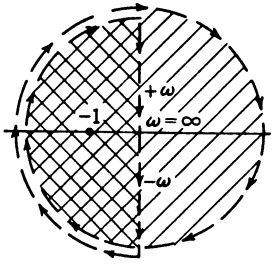
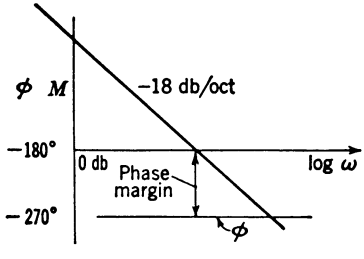
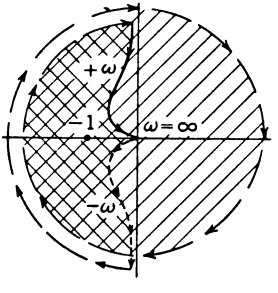
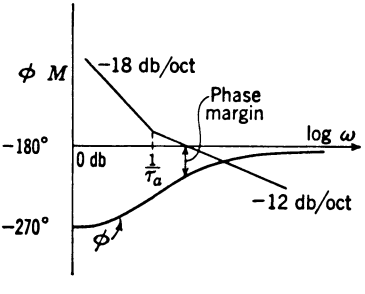
$G(s)$	Polar Plot	Bode Diagram
9. $\dfrac{K}{s^2(s\tau_1+1)}$		
10. $\dfrac{K(s\tau_a+1)}{s^2(s\tau_1+1)}$ $\tau_a > \tau_1$		
11. $\dfrac{K}{s^3}$		
12. $\dfrac{K(s\tau_a+1)}{s^3}$		

Table 8 (*Continued*)

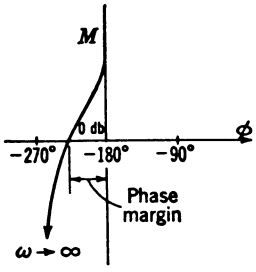
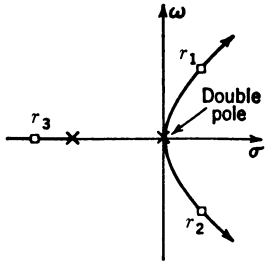
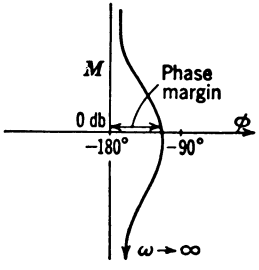
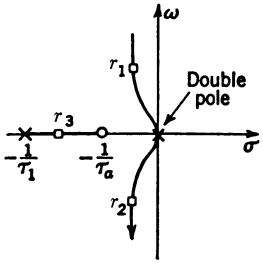
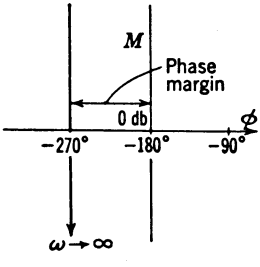
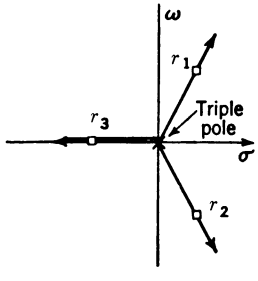
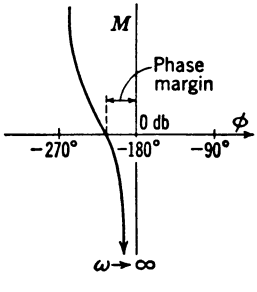
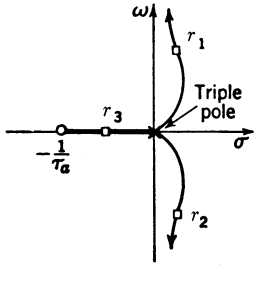
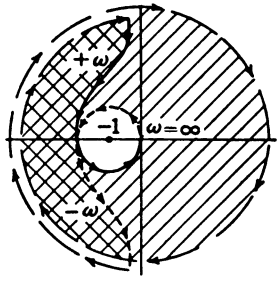
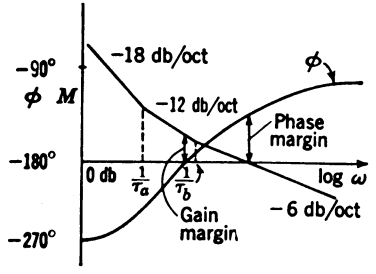
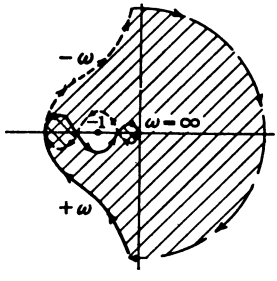
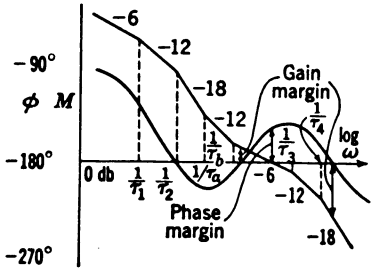
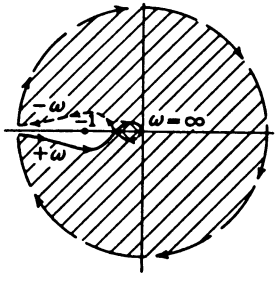
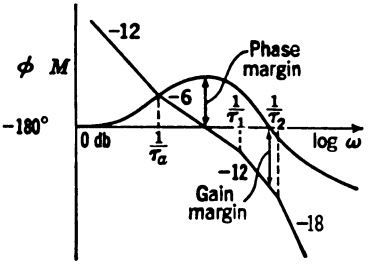
Nichols Diagram	Root Locus	Comments
		Inherently unstable; must be compensated
		Stable for all gains
		Inherently unstable
		Inherently unstable

Table 8 (*Continued*)

$G(s)$	Polar Plot	Bode Diagram
13. $\frac{K(s\tau_a+1)(s\tau_b+1)}{s^3}$		
14. $\frac{K(s\tau_a+1)(s\tau_b+1)}{(s\tau_1+1)(s\tau_2+1)(s\tau_3+1)(s\tau_4+1)}$		
15. $\frac{K(s\tau_a+1)}{s^2(s\tau_1+1)(s\tau_2+1)}$		

$$sX(s) - x(0) = AX(s) + BU(s)$$

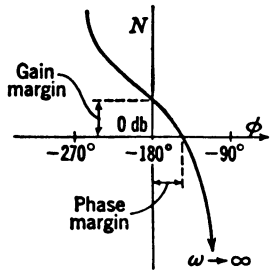
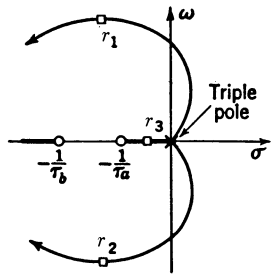
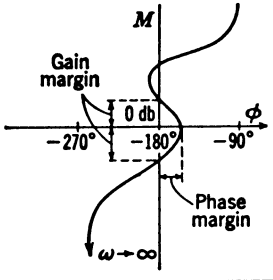
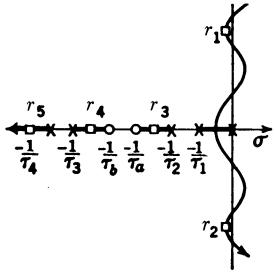
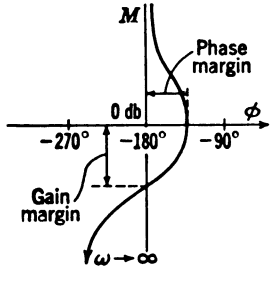
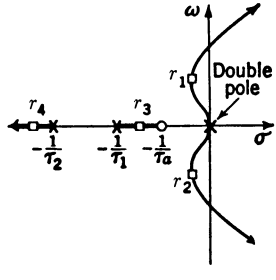
The solution to the fixed linear system therefore can be written as

$$\begin{aligned} x(t) &= \mathcal{L}^{-1}[X(s)] \\ &= \mathcal{L}^{-1}[\Phi(s)]x(0) + \mathcal{L}^{-1}[\Phi(s)BU(s)] \end{aligned}$$

where $\Phi(s)$ is called the *resolvent matrix* and

$$\Phi(t) = \mathcal{L}^{-1}[\Phi(s)] = \mathcal{L}^{-1}[sI - A]^{-1}$$

Table 8 (*Continued*)

Nichols Diagram	Root Locus	Comments
		Conditionally stable; becomes unstable if gain is too low
		Conditionally stable; stable at low gain, becomes unstable as gain is raised, again becomes stable as gain is further increased, and becomes unstable for very high gains
		Conditionally stable; becomes unstable at high gain

6.2 Eigenstructure

The internal structure of a system (and therefore its free response) is defined entirely by the system matrix A. The concept of matrix *eigenstructure,* as defined by the eigenvalues and eigenvectors of the system matrix, can provide a great deal of insight into the fundamental behavior of a system. In particular, the system eigenvectors can be shown to define a special set of first-order subsystems embedded within the system. These subsystems behave independently of one another, a fact that greatly simplifies analysis.

System Eigenvalues and Eigenvectors

For a system with system matrix A, the system *eigenvectors* v_i and associated *eigenvalues* λ_i are defined by the equation

$$Av_i = \lambda_i v_i$$

Note that the eigenvectors represent a set of special directions in the state space. If the state vector is aligned in one of these directions, then the homogeneous state equation becomes $\dot{v}_i = A\dot{v}_i = \lambda v_i$, implying that each of the state variables changes at the *same* rate determined by the eigenvalue λ_i. This further implies that, in the absence of inputs to the system, a state vector that becomes aligned with an eigenvector will remain aligned with that eigenvector.

The system eigenvalues are calculated by solving the nth-order polynomial equation

$$|\lambda I - A| = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 = 0$$

This equation is called the *characteristic equation.* Thus the system eigenvalues are the roots of the characteristic equation, that is, the system eigenvalues are identically the system poles defined in transform analysis.

Each system eigenvector is determined by substituting the corresponding eigenvalue into the defining equation and then solving the resulting set of simultaneous linear equations. Only $n - 1$ of the n components of any eigenvector are independently defined, however. In other words, the magnitude of an eigenvector is arbitrary, and the eigenvector describes a direction in the state space.

Diagonalized Canonical Form

There will be one linearly independent eigenvector for each distinct (nonrepeated) eigenvalue. If all of the eigenvalues of an nth-order system are distinct, then the n independent eigenvectors form a new basis for the state space. This basis represents new coordinate axes defining a set of state variables $z_i(t)$, $i = 1, 2, \ldots, n$, called the *diagonalized canonical variables.* In terms of the diagonalized variables, the homogeneous state equation is

$$\dot{z}(t) = \Lambda z$$

where Λ is a diagonal system matrix of the eigenvectors, that is,

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & \lambda_n \end{bmatrix}$$

The solution to the diagonalized homogeneous system is

$$z(t) = e^{\Lambda t} z(0)$$

where $e^{\Lambda t}$ is the diagonal state transition matrix

$$e^{\Lambda t} = \begin{bmatrix} e^{\lambda_1 t} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2 t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\lambda_n t} \end{bmatrix}$$

Modal Matrix

Consider the state equation of the nth-order system

$$\dot{x}(t) = Ax(t) + Bu(t)$$

which has real, distinct eigenvalues. Since the system has a full set of eigenvectors, the state vector $x(t)$ can be expressed in terms of the canonical state variables as

$$x(t) = v_1 z_1(t) + v_2 z_2(t) + \cdots + v_n z_n(t) = Mz(t)$$

where M is the $n \times n$ matrix whose columns are the eigenvectors of A, called the *modal matrix.* Using the modal matrix, the state transition matrix for the original system can be written as

$$\Phi(t) = e^{\Lambda t} = Me^{\Lambda t}M^{-1}$$

where $e^{\Lambda t}$ is the diagonal state transition matrix. This frequently proves to be an attractive method for determining the state transition matrix of a system with real, distinct eigenvalues.

Jordan Canonical Form

For a system with one or more repeated eigenvalues, there is not in general a full set of eigenvectors. In this case, it is not possible to determine a diagonal representation for the system. Instead, the simplest representation that can be achieved is block diagonal. Let $L_k(\lambda)$ be the $k \times k$ matrix

$$L_k(\lambda) = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & \cdots & 0 \\ \vdots & \vdots & \lambda & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & 0 & \lambda \end{bmatrix}$$

Then for any $n \times n$ system matrix A there is certain to exist a nonsingular matrix T such that

$$T^{-1}AT = \begin{bmatrix} L_{k_1}(\lambda_1) & & & \\ & L_{k_2}(\lambda_2) & & \\ & & \ddots & \\ & & & L_{k_r}(\lambda_r) \end{bmatrix}$$

where $k_1 + k_2 + \cdots + k_r = n$ and λ_i, $i = 1, 2, \ldots, r$, are the (not necessarily distinct) eigenvalues of A. The matrix $T^{-1}AT$ is called the *Jordan canonical form.*

7 SIMULATION

7.1 Simulation—Experimental Analysis of Model Behavior

Closed-form solutions for nonlinear or time-varying systems are rarely available. In addition, while explicit solutions for time-invariant linear systems can always be found, for high-order systems this is often impractical. In such cases it may be convenient to study the dynamic behavior of the system using *simulation.*

Simulation is the *experimental* analysis of model behavior. A *simulation run* is a controlled experiment in which a specific realization of the model is manipulated in order to determine the response associated with that realization. A *simulation study* comprises *multiple runs,* each run for a different combination of model parameter values and/or initial conditions. The generalized solution of the model must then be inferred from a finite number of simulated data points.

Simulation is almost always carried out with the assistance of computing equipment. *Digital simulation* involves the *numerical solution* of model equations using a digital computer. *Analog simulation* involves solving model equations by analogy with the behavior of a physical system using an analog computer. *Hybrid simulation* employs digital and analog simulation together using a hybrid (part digital and part analog) computer.

7.2 Digital Simulation

Digital continuous-system simulation involves the approximate solution of a state-variable model over successive time steps. Consider the general state-variable equation

$$\dot{x}(t) = f[x(t), u(t)]$$

to be simulated over the time interval $t_0 \leq t \leq t_K$. The solution to this problem is based on the repeated solution of the single-variable, single-step subproblem depicted in Fig. 24. The subproblem may be stated formally as follows:

Given:

1. $\Delta t(k) = t_k - t_{k-1}$, the length of the kth *time step.*
2. $x_i(t) = f_i[x(t), u(t)]$ for $t_{k-1} \leq t \leq t_k$, the ith equation of state defined for the state variable $x_i(t)$ over the kth time step.
3. $u(t)$ for $t_{k-1} \leq t \leq t_k$, the input vector defined for the kth time step.
4. $\tilde{x}(k-1) \simeq x(t_{k-1})$, an initial approximation for the state vector at the beginning of the time step.

Find:

5. $\tilde{x}_i(k) \simeq x_i(t_k)$, a final approximation for the state variable $x_i(t)$ at the end of the kth time step.

Solving this single-variable, single-step subproblem for each of the state variables $x_i(t)$, $i = 1, 2, \ldots, n$, yields a final approximation for the state vector $\tilde{x}(k) \simeq x(t_k)$ at the end of the kth time step. Solving the complete single-step problem K times over K time steps, beginning with the initial condition $\tilde{x}(0) = x(t_0)$ and using the final value of $\tilde{x}(t_k)$ from the

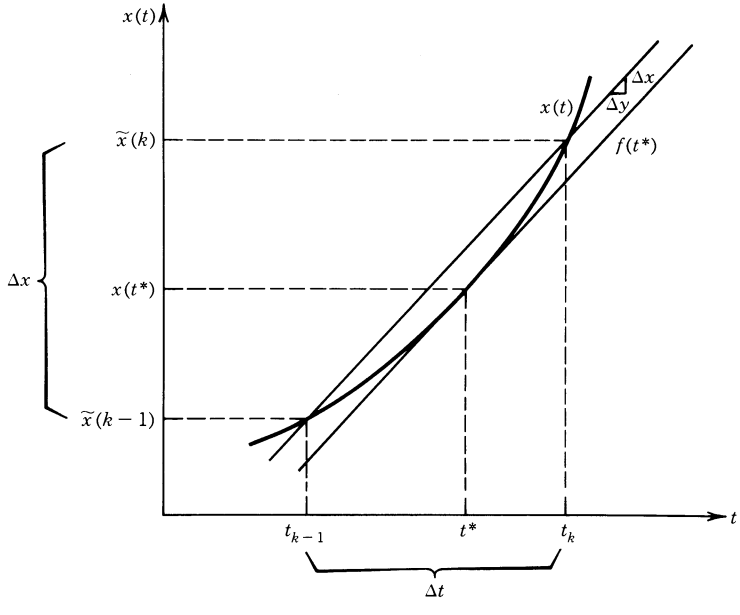


Figure 24 Numerical approximation of a single variable over a single time step.

kth time step as the initial value of the state for the $(k + 1)$st time step, yields a discrete succession of approximations $\tilde{x}(1) \simeq x(t_1)$, $\tilde{x}(2) \simeq x(t_2)$, . . . , $\tilde{x}(K) \simeq x(t_k)$ spanning the solution time interval.

The basic procedure for completing the single-variable, single-step problem is the same regardless of the particular integration method chosen. It consists of two parts: (1) calculation of the average value of the ith derivative over the time step as

$$\dot{x}_i(t^*) = f_i[x(t^*), u(t^*)] = \frac{\Delta x_i(k)}{\Delta t(k)} \simeq \tilde{f}_i(k)$$

and (2) calculation of the final value of the simulated variable at the end of the time step as

$$\begin{aligned} \tilde{x}_i(k) &= \tilde{x}_i(k-1) + \Delta x_i(k) \\ &\simeq \tilde{x}_i(k-1) + \Delta t(k)\tilde{f}_i(k) \end{aligned}$$

If the function $f_i[x(t), u(t)]$ is continuous, then t^* is guaranteed to be on the time step, that is, $t_{k-1} \leq t^* \leq t_k$. Since the value of t^* is otherwise unknown, however, the value of $x(t^*)$ can only be approximated as $\tilde{f}(k)$.

Different *numerical integration* methods are distinguished by the means used to calculate the approximation $f_i(k)$. A wide variety of such methods is available for digital simulation of dynamic systems. The choice of a particular method depends on the nature of the model being simulated, the accuracy required in the simulated data, and the computing effort available for the simulation study. Several popular classes of integration methods are outlined in the following sections.

Euler Method

The simplest procedure for numerical integration is the Euler method. The standard Euler method approximates the average value of the ith derivative over the kth time step using the derivative evaluated at the beginning of the time step, that is,

$$\tilde{f}_i(k) = f_i[\tilde{x}(k-1), u(t_{k-1})] \simeq f_i(t_{k-1})$$

$i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, K$. This is shown geometrically in Fig. 25 for the scalar single-step case. A modification of this method uses the newly calculated state variables in the derivative calculation as these new values become available. Assuming the state variables are computed in numerical order according to the subscripts, this implies

$$\tilde{f}_i(k) = f_i[\tilde{x}_1(k), \ldots, \tilde{x}_{i-1}(k), \tilde{x}_i(k-1), \ldots, \tilde{x}_n(k-1), u(t_{k-1})]$$

The modified Euler method is modestly more efficient than the standard procedure and, frequently, is more accurate. In addition, since the input vector $u(t)$ is usually known for the entire time step, using an average value of the input, such as

$$u(k) = \frac{1}{\Delta t(k)} \int_{t_{k-1}}^{t_k} u(\tau)\, d\tau$$

frequently leads to a superior approximation of $\tilde{f}_i(k)$.

The Euler method requires the least amount of computational effort per time step of any numerical integration scheme. Local truncation error is proportional to Δt^2, however, which means that the error within each time step is highly sensitive to step size. Because the accuracy of the method demands very small time steps, the number of time steps required to implement the method successfully can be large relative to other methods. This can imply

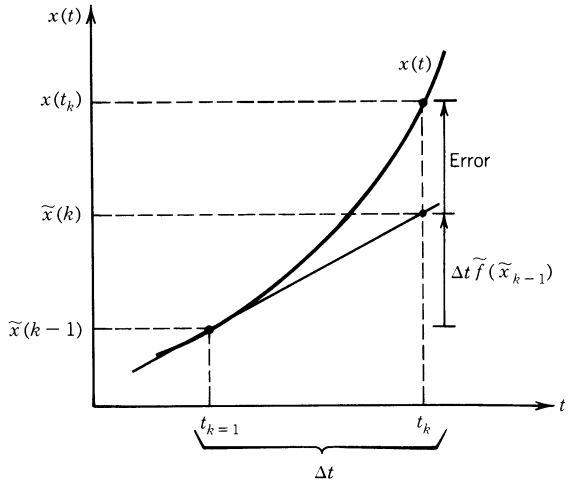


Figure 25 Geometric interpretation of the Euler method for numerical integration.

a large computational overhead and can lead to inaccuracies through the accumulation of roundoff error at each step.

Runge–Kutta Methods

Runge–Kutta methods precompute two or more values of $f_i[x(t), u(t)]$ in the time step $t_{k-1} \leq t \leq t_k$ and use some weighted average of these values to calculate $\tilde{f}_i(k)$. The *order* of a Runge–Kutta method refers to the number of derivative terms (or *derivative calls*) used in the scalar single-step calculation. A Runge–Kutta routine of order N therefore uses the approximation

$$\tilde{f}_i(k) = \sum_{j=1}^{N} w_j f_{ij}(k)$$

where the N approximations to the derivative are

$$f_{i1}(k) = f_i[\tilde{x}(k-1), u(t_{k-1})]$$

(the Euler approximation) and

$$f_{ij} = f_i\left[\tilde{x}(k-1) + \Delta t \sum_{t=1}^{j-1} I b_{jt} f_{il},\ u\left(t_{k-1} + \Delta t \sum_{t=1}^{j-1} b_{jl}\right)\right]$$

where I is the identity matrix. The weighting coefficients w_j and b_{jl} are not unique, but are selected such that the error in the approximation is zero when $x_i(t)$ is some specified Nth-degree polynomial in t. Coefficients commonly used for Runge–Kutta integration are given in Table 9.

Among the most popular of the Runge–Kutta methods is fourth-order Runge–Kutta. Using the defining equations for $N = 4$ and the weighting coefficients from Table 9 yields the derivative approximation

$$\tilde{f}_i(k) = \tfrac{1}{6}[f_{i1}(k) + 2f_{i2}(k) + 2f_{i3}(k) + f_{i4}(k)]$$

based on the four derivative calls

Table 9 Coefficients Commonly Used for Runge–Kutta Numerical Integration[6]

Common Name	N	b_{jl}	w_j
Open or explicit Euler	1	All zero	$w_1 = 1$
Improved polygon	2	$b_{21} = \frac{1}{2}$	$w_1 = 0$ $w_2 = 1$
Modified Euler or Heun's method	2	$b_{21} = 1$	$w_1 = \frac{1}{2}$ $w_2 = \frac{1}{2}$
Third-order Runge–Kutta	3	$b_{21} = \frac{1}{2}$ $b_{31} = -1$ $b_{32} = 2$	$w_1 = \frac{1}{6}$ $w_2 = \frac{2}{3}$ $w_3 = \frac{1}{6}$
Fourth-order Runge–Kutta	4	$b_{21} = \frac{1}{2}$ $b_{31} = 0$ $b_{32} = \frac{1}{2}$ $b_{43} = 1$	$w_1 = \frac{1}{6}$ $w_2 = \frac{1}{3}$ $w_3 = \frac{1}{3}$ $w_4 = \frac{1}{6}$

$$f_{i1}(k) = f_i[\tilde{x}(k-1), u(t_{k-1})]$$

$$f_{i2}(k) = f_i\left[\tilde{x}(k-1) + \frac{\Delta t}{2} If_{i1}, u\left(t_{k-1} + \frac{\Delta t}{2}\right)\right]$$

$$f_{i3}(k) = f_i\left[\tilde{x}(k-1) + \frac{\Delta t}{2} If_{i2}, u\left(t_{k-1} + \frac{\Delta t}{2}\right)\right]$$

$$f_{i4}(k) = f_i[\tilde{x}(k-1) + \Delta t\, If_{i3}, u(t_k)]$$

where I is the identity matrix.

Because Runge–Kutta formulas are designed to be exact for a polynomial of order N, local truncation error is of the order Δt^{N+1}. This considerable improvement over the Euler method means that comparable accuracy can be achieved for larger step sizes. The penalty is that N derivative calls are required for each scalar evaluation within each time step.

Euler and Runge–Kutta methods are examples of *single-step methods* for numerical integration, so-called because the state $x(k)$ is calculated from knowledge of the state $x(k-1)$, without requiring knowledge of the state at any time prior to the beginning of the current time step. These methods are also referred to as *self-starting methods,* since calculations may proceed from any known state.

Multistep Methods

Multistep methods differ from the single-step methods previously described in that multistep methods use the stored values of two or more previously computed states and/or derivatives in order to compute the derivative approximation $\tilde{f}_i(k)$ for the current time step. The advantage of multistep methods over Runge–Kutta methods is that these require only one derivative call for each state variable at each time step for comparable accuracy. The disadvantage is that multistep methods are not self-starting, since calculations cannot proceed from the initial state alone. Multistep methods must be started, or restarted in the case of discontinuous derivatives, using a single-step method to calculate the first several steps.

The most popular of the multistep methods are the *Adams–Bashforth predictor methods* and the *Adams–Moulton corrector methods.* These methods use the derivative approximation

$$\tilde{f}_i(k) = \sum_{j=0}^{N} b_j f_i[\tilde{x}(k-j),\, u(k-j)]$$

where the b_j are weighting coefficients. These coefficients are selected such that the error in the approximation is zero when $x_i(t)$ is a specified polynomial. Table 10 gives the values of the weighting coefficients for several Adams–Bashforth–Moulton rules. Note that the predictor methods employ an *open* or *explicit rule,* since for these methods $b_0 = 0$ and a prior estimate of $x_i(k)$ is not required. The corrector methods use a *closed* or *implicit rule,* since for these methods $b_i \neq 0$ and a prior estimate of $x_i(k)$ is required. Note also that for all of these methods $\Sigma_{j=0}^{N} b_j = 1$, ensuring unity gain for the integration of a constant.

Predictor–Corrector Methods

Predictor–corrector methods use one of the multistep predictor equations to provide an initial estimate (or ''prediction'') of $x(k)$. This initial estimate is then used with one of the multistep corrector equations to provide a second and improved (or ''corrected'') estimate of $x(k)$ before proceeding to the next step. A popular choice is the four-point Adams–Bashforth predictor together with the four-point Adams–Moulton corrector, resulting in a prediction of

$$\tilde{x}_i(k) = \tilde{x}_i(k-1) + \frac{\Delta t}{24}\,[55\tilde{f}_i(k-1) - 59\tilde{f}_i(k-2) + 37\tilde{f}_i(k-3) - 9\tilde{f}_i(k-4)]$$

for $i = 1, 2, \ldots, n$ and a correction of

$$\tilde{x}_i(k) = \tilde{x}_i(k-1) + \frac{\Delta t}{24}\,\{9f_i[\tilde{x}(k), u(k)] + 19\tilde{f}_i(k-1) - 5\tilde{f}_i(k-2) + \tilde{f}_i(k-3)\}$$

Predictor–corrector methods generally incorporate a strategy for increasing or decreasing the size of the time step depending on the difference between the predicted and corrected $x(k)$ values. Such *variable time step methods* are particularly useful if the simulated system possesses local time constants that differ by several orders of magnitude or if there is little a priori knowledge about the system response.

Numerical Integration Errors

An inherent characteristic of digital simulation is that the discrete data points generated by the simulation $x(k)$ are only approximations to the exact solution $x(t_k)$ at the corresponding

Table 10 Coefficients Commonly Used for Adams–Bashforth–Moulton Numerical Integration[6]

Common Name	Predictor or Corrector	Points	b_{-1}	b_0	b_1	b_2	b_3
Open or explicit Euler	Predictor	1	0	1	0	0	0
Open trapezoidal	Predictor	2	0	3/2	−1/2	0	0
Adams three-point predictor	Predictor	3	0	23/12	−16/12	5/12	0
Adams four-point predictor	Predictor	4	0	55/24	−59/24	37/24	−9/24
Closed or implicit Euler	Corrector	1	1	0	0	0	0
Closed trapezoidal	Corrector	2	1/2	1/2	0	0	0
Adams three-point corrector	Corrector	3	5/12	8/12	−1/12	0	0
Adams four-point corrector	Corrector	4	9/24	19/24	−5/24	1/24	0

point in time. This results from two types of errors that are unavoidable in the numerical solutions. *Round-off errors* occur because numbers stored in a digital computer have finite word length (i.e., a finite number of bits per word) and therefore limited precision. Because the results of calculations cannot be stored exactly, round-off error tends to increase with the number of calculations performed. For a given total solution interval $t_0 \leq t \leq t_K$, therefore, round-off error tends to increase (1) with increasing integration-rule order (since more calculations must be performed at each time step) and (2) with decreasing step size Δt (since more time steps are required).

Truncation errors or *numerical approximation errors* occur because of the inherent limitations in the numerical integration methods themselves. Such errors would arise even if the digital computer had infinite precision. *Local* or *per-step truncation error* is defined as

$$e(k) = x(k) - x(t_k)$$

given that $x(k-1) = x(t_{k-1})$ and that the calculation at the kth time step is infinitely precise. For many integration methods, local truncation errors can be approximated at each step. *Global* or *total truncation error* is defined as

$$e(K) = x(K) - x(t_K)$$

given that $x(0) = x(t_0)$ and the calculations for all K time steps are infinitely precise. Global truncation error usually cannot be estimated, neither can efforts to reduce local truncation errors be guaranteed to yield acceptable global errors. In general, however, truncation errors can be decreased by using more sophisticated integration methods and by decreasing the step size Δt.

Time Constants and Time Steps

As a general rule, the step size Δt for simulation must be less than the smallest local time constant of the model simulated. This can be illustrated by considering the simple first-order system

$$\dot{x}(t) = \lambda x(t)$$

and the difference equation defining the corresponding Euler integration

$$x(k) = x(k-1) + \Delta t\, \lambda\, x(k-1)$$

The continuous system is stable for $\lambda < 0$, while the discrete approximation is stable for $|1 + \lambda\, \Delta t| < 1$. If the original system is stable, therefore, the simulated response will be stable for

$$\Delta t \leq 2 \left| \frac{1}{\lambda} \right|$$

where the equality defines the *critical step size.* For larger step sizes, the simulation will exhibit *numerical instability.* In general, while higher order integration methods will provide greater per-step accuracy, the critical step size itself will not be greatly reduced.

A major problem arises when the simulated model has one or more time constants $|1/\lambda_i|$ that are small when compared to the total solution time interval $t_0 \leq t \leq t_K$. Numerical stability will then require very small Δt, even though the transient response associated with the higher frequency (larger λ_i) subsystems may contribute little to the particular solution.

Such problems can be addressed either by neglecting the higher frequency components where appropriate or by adopting special numerical integration methods for *stiff systems.*

Selecting an Integration Method

The best numerical integration method for a specific simulation is the method that yields an acceptable global approximation error with the minimum amount of round-off error and computing effort. No single method is best for all applications. The selection of an integration method depends on the model simulated, the purpose of the simulation study, and the availability of computing hardware and software.

In general, for well-behaved problems with continuous derivatives and no stiffness, a lower order Adams predictor is often a good choice. Multistep methods also facilitate estimating local truncation error. Multistep methods should be avoided for systems with discontinuities, however, because of the need for frequent restarts. Runge–Kutta methods have the advantage that these are self-starting and provide fair stability. For stiff systems where high-frequency modes have little influence on the global response, special stiff-system methods enable the use of economically large step sizes. Variable-step rules are useful when little is known a priori about solutions. Variable-step rules often make a good choice as general-purpose integration methods.

Round-off error usually is not a major concern in the selection of an integration method, since the goal of minimizing computing effort typically obviates such problems. Double-precision simulation can be used where round-off is a potential concern. An upper bound on step size often exists because of discontinuities in derivative functions or because of the need for response output at closely spaced time intervals.

Continuous-System Simulation Languages

Digital simulation can be implemented for a specific model in any high-level language such as FORTRAN or C. The general process for implementing a simulation is shown in Fig. 26. In addition, many special-purpose continuous-system simulation languages are commonly available across a wide range of platforms. Such languages greatly simplify programming tasks and typically provide for good graphical output.

8 MODEL CLASSIFICATIONS

Mathematical models of dynamic systems are distinguished by several criteria which describe fundamental properties of model variables and equations. These criteria in turn prescribe the theory and mathematical techniques that can be used to study different models. Table 11 summarizes these distinguishing criteria. In the following sections, the approaches adopted for the analysis of important classes of systems are briefly outlined.

8.1 Stochastic Systems

Systems in which some of the dependent variables (input, state, output) contain random components are called *stochastic systems.* Randomness may result from environmental factors, such as wind gusts or electrical noise, or simply from a lack of precise knowledge of the system model, such as when a human operator is included within a control system. If the randomness in the system can be described by some rule, then it is often possible to derive a model in terms of probability distributions involving, for example, the means and variances of model variables or parameters.

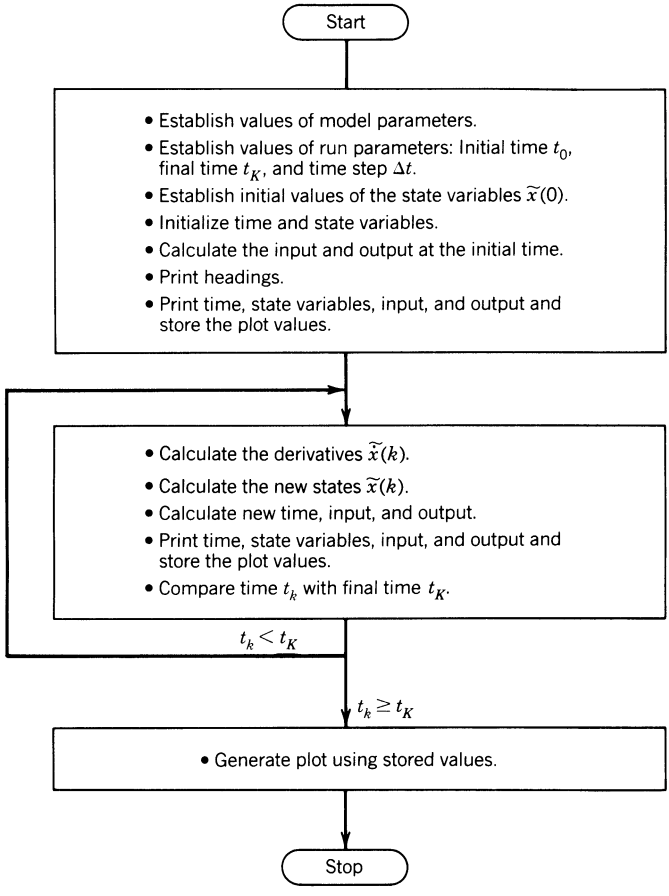


Figure 26 General process for implementing digital simulation (adapted from Close and Frederick[3]).

State-Variable Formulation

A common formulation is the fixed, linear model with additive noise

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t)$$

$$y(t) = Cx(t) + v(t)$$

where $w(t)$ is a zero-mean Gaussian disturbance and $v(t)$ is a zero-mean Gaussian measurement noise. This formulation is the basis for many *estimation problems,* including the problem of *optimal filtering.* Estimation essentially involves the development of a rule or algorithm for determining the best estimate of the past, current, or future values of measured variables in the presence of disturbances or noise.

Random Variables

In the following, important concepts for characterizing random signals are developed. A *random variable x* is a variable that assumes values that cannot be precisely predicted a priori. The likelihood that a random variable will assume a particular value is measured as

Table 11 Classification of Mathematical Models of Dynamic Systems

Criterion	Classification	Description
Certainty	Deterministic	Model parameters and variables can be known with certainty. Common approximation when uncertainties are small.
	Stochastic	Uncertainty exists in the values of some parameters and/or variables. Model parameters and variables are expressed as random numbers or processes and are characterized by the parameters of probability distributions.
Spatial characteristics	Lumped	State of the system can be described by a finite set of state variables. Model is expressed as a discrete set of point functions described by ordinary differential or difference equations.
	Distributed	State depends on both time and spatial location. Model is usually described by variables that are continuous in time and space, resulting in partial differential equations. Frequently approximated by lumped elements. Typical in the study of structures and mass and heat transport.
Parameter variation	Fixed or time invariant	Model parameters are constant. Model described by differential or difference equations with constant coefficients. Model with same initial conditions and input delayed by t_d has the same response delayed by t_d.
	Time varying	Model parameters are time dependent.
Superposition property	Linear	Superposition applies. Model can be expressed as a system of linear difference or differential equations.
	Nonlinear	Superposition does not apply. Model is expressed as a system of nonlinear difference or differential equations. Frequently approximated by linear systems for analytical ease.
Continuity of independent variable (time)	Continuous	Dependent variables (input, output, state) are defined over a continuous range of the independent variable (time), even though the dependence is not necessarily described by a mathematically continuous function. Model is expressed as differential equations. Typical of physical systems.
	Discrete	Dependent variables are defined only at distinct instants of time. Model is expressed as difference equations. Typical of digital and nonphysical systems.
	Hybrid	System with continuous and discrete subsystems, most common in computer control and communication systems. Sampling and quantization typical in A/D (analog-to-digital) conversion; signal reconstruction for D/A conversion. Model frequently approximated as entirely continuous or entirely discrete.
Quantization of dependent variables	Nonquantized	Dependent variables are continuously variable over a range of values. Typical of physical systems at macroscopic resolution.
	Quantized	Dependent variables assume only a countable number of different values. Typical of computer control and communication systems (sample data systems).

the *probability* of that value. The probability *distribution function* $F(x)$ of a continuous random variable x is defined as the probability that x assumes a value no greater than x, that is,

$$F(x) = \Pr(X \leq x) = \int_{-\infty}^{x} f(x)\, dx$$

The probability *density function* $f(x)$ is defined as the derivative of $F(x)$.

The *mean* or *expected value* of a probability distribution is defined as

$$E(X) = \int_{-\infty}^{\infty} xf(x)\, dx = \overline{X}$$

The mean is the first moment of the distribution. The *nth moment* of the distribution is defined as

$$E(X^n) = \int_{-\infty}^{\infty} x^n f(x)\, dx$$

The mean square of the difference between the random variable and its mean is the *variance* or *second central moment* of the distribution,

$$\sigma^2(X) = E(X - \overline{X})^2 = \int_{-\infty}^{\infty} (x - \overline{X})^2 f(x)\, dx = E(X^2) - [E(X)]^2$$

The square root of the variance is the *standard deviation* of the distribution:

$$\sigma(X) = \sqrt{E(X^2) - [E(X)]^2}$$

The mean of the distribution therefore is a measure of the average magnitude of the random variable, while the variance and standard deviation are measures of the variability or dispersion of this magnitude.

The concepts of probability can be extended to more than one random variable. The *joint distribution* function of two random variables x and y is defined as

$$F(x,y) = \Pr(X < x \text{ and } Y < y) = \int_{-\infty}^{x}\int_{-\infty}^{y} f(x,y)\, dy\, dx$$

where $f(x,y)$ is the joint distribution. The ijth moment of the joint distribution is

$$E(X^iY^j) = \int_{-\infty}^{\infty} x^i \int_{-\infty}^{\infty} y^j f(x,y)\, dy\, dx$$

The *covariance* of x and y is defined to be

$$E[(X - \overline{X})(Y - \overline{Y})]$$

and the normalized covariance or *correlation coefficient* as

$$\rho = \frac{E[(X - \overline{X})(Y - \overline{Y})]}{\sqrt{\sigma^2(X)\sigma^2(Y)}}$$

Although many distribution functions have proven useful in control engineering, far and away the most useful is the *Gaussian* or *normal distribution*

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{(-x - \mu)^2}{2\sigma^2}\right]$$

where μ is the mean of the distribution and σ is the standard deviation. The Gaussian distribution has a number of important properties. First, if the input to a linear system is Gaussian, the output also will be Gaussian. Second, if the input to a linear system is only approximately Gaussian, the output will tend to approximate a Gaussian distribution even more closely. Finally, a Gaussian distribution can be completely specified by two parameters, μ and σ, and therefore a zero-mean Gaussian variable is completely specified by its variance.

Random Processes

A *random process* is a set of random variables with time-dependent elements. If the statistical parameters of the process (such as σ for the zero-mean Gaussian process) do not vary with time, the process is *stationary.* The *autocorrelation function* of a stationary random variable $x(t)$ is defined by

$$\phi_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)x(t + \tau)\, dt$$

a function of the fixed time interval τ. The autocorrelation function is a quantitative measure of the sequential dependence or time correlation of the random variable, that is, the relative effect of prior values of the variable on the present or future values of the variable. The autocorrelation function also gives information regarding how rapidly the variable is changing and about whether the signal is in part deterministic (specifically, periodic). The autocorrelation function of a zero-mean variable has the properties

$$\sigma^2 = \phi_{xx}(0) \geq \phi_{xx}(\tau) \qquad \phi_{xx}(\tau) = \phi_{xx}(-\tau)$$

In other words, the autocorrelation function for $\tau = 0$ is identically the variance and the variance is the maximum value of the autocorrelation function. From the definition of the function, it is clear that (1) for a purely random variable with zero mean, $\phi_{xx}(\tau) = 0$ for $\tau \neq 0$, and (2) for a deterministic variable, which is periodic with period T, $\phi_{xx}(k2\pi T) = \sigma^2$ for k integer. The concept of time correlation is readily extended to more than one random variable. The *cross-correlation function* between the random variables $x(t)$ and $y(t)$ is

$$\phi_{xy}(\tau) = \lim_{T\to\infty} \int_{-\infty}^{\infty} x(t)y(t + \tau)\, dt$$

For $\tau = 0$, the cross-correlation between two zero-mean variables is identically the covariance. A final characterization of a random variable is its *power spectrum,* defined as

$$G(\omega, x) = \lim_{T\to\infty} \frac{1}{2\pi T} \left| \int_{-T}^{T} x(t)e^{-j\omega t}\, dt \right|^2$$

For a stationary random process, the power spectrum function is identically the Fourier transform of the autocorrelation function

$$G(\omega, x) = \frac{1}{\pi} \int_{-\infty}^{\infty} \phi_{xx}(\tau)e^{-j\omega t}\, dt$$

with

$$\phi_{xx}(0) = \int_{-\infty}^{\infty} G(\omega,x)\, d\omega$$

8.2 Distributed-Parameter Models

There are many important applications in which the state of a system cannot be defined at a finite number of points in space. Instead, the system state is a continuously varying function of both time and location. When continuous spatial dependence is explicitly accounted for in a model, the independent variables must include spatial coordinates as well as time. The resulting *distributed-parameter model* is described in terms of *partial differential equations,* containing partial derivatives with respect to each of the independent variables.

Distributed-parameter models commonly arise in the study of mass and heat transport, the mechanics of structures and structural components, and electrical transmission. Consider as a simple example the unidirectional flow of heat through a wall, as depicted in Fig. 27. The temperature of the wall is not in general uniform but depends on both the time t and position within the wall x, that is, $\theta = \theta(x, t)$. A distributed-parameter model for this case might be the first-order partial differential equation

$$\frac{d}{dt}\,\theta(x,t) = \frac{1}{C_t}\frac{\partial}{\partial x}\left[\frac{1}{R_t}\frac{\partial}{\partial x}\,\theta(x,t)\right]$$

where C_t is the thermal capacitance and R_t is the thermal resistance of the wall (assumed uniform).

The complexity of distributed-parameter models is typically such that these models are avoided in the analysis and design of control systems. Instead, distributed-parameter systems are approximated by a finite number of spatial "lumps," each lump being characterized by some average value of the state. By eliminating the independent spatial variables, the result is a *lumped-parameter* (*or lumped-element*) *model* described by coupled ordinary differential equations. If a sufficiently fine-grained representation of the lumped microstructure can be achieved, a lumped model can be derived that will approximate the distributed model to any desired degree of accuracy. Consider, for example, the three temperature lumps shown in Fig. 28, used to approximate the wall of Fig. 27. The corresponding third-order lumped approximation is

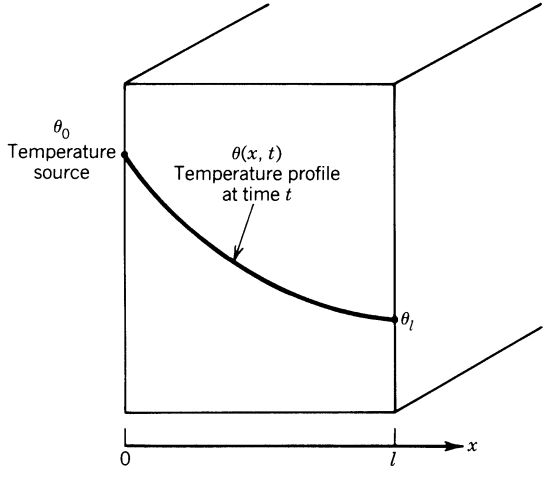


Figure 27 Uniform heat transfer through a wall.

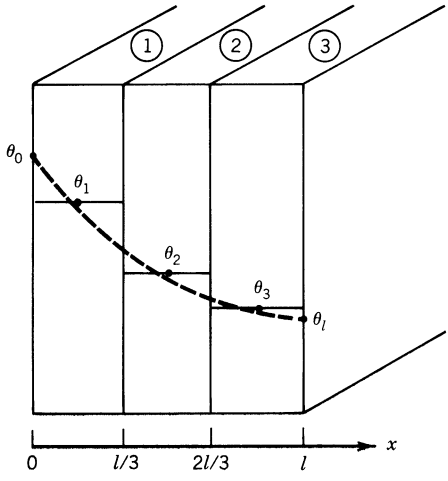


Figure 28 Lumped-parameter model for uniform heat transfer through a wall.

$$\frac{d}{dt}\begin{bmatrix}\theta_1(t)\\ \theta_2(t)\\ \theta_3(t)\end{bmatrix} = \begin{bmatrix}-\dfrac{9}{C_tR_t} & \dfrac{3}{C_tR_t} & 0\\ \dfrac{3}{C_tR_t} & -\dfrac{6}{C_tR_t} & \dfrac{3}{C_tR_t}\\ 0 & \dfrac{3}{C_tR_t} & -\dfrac{6}{C_tR_t}\end{bmatrix}\begin{bmatrix}\theta_1(t)\\ \theta_2(t)\\ \theta_3(t)\end{bmatrix} + \begin{bmatrix}\dfrac{6}{C_tR_t}\\ 0\\ 0\end{bmatrix}\theta_0(t)$$

If a more detailed approximation is required, this can always be achieved at the expense of adding additional, smaller lumps.

8.3 Time-Varying Systems

Time-varying systems are those with characteristics that change as a function of time. Such variation may result from environmental factors, such as temperature or radiation, or from factors related to the operation of the system, such as fuel consumption. While in general a model with variable parameters can be either linear or nonlinear, the name time varying is most frequently associated with linear systems described by the following state equation:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

For this linear time-varying model, the superposition principle still applies. Superposition is a great aid in model formulation but unfortunately does not prove to be much help in determining the model solution.

Paradoxically, the form of the solution to the linear time-varying equation is well known[7]:

$$x(t) = \Phi(t,t_0)x(t_0) + \int_{t_0}^{t} \Phi(t,\tau)B(\tau)u(\tau)\,d\tau$$

where $\Phi(t,t_0)$ is the time-varying state transition matrix. This knowledge is typically of little value, however, since it is not usually possible to determine the state transition matrix by any straightforward method. By analogy with the first-order case, the relationship

$$\Phi(t,t_0) = \exp\left(\int_{t_0}^{t} A(\tau)\, d\tau\right)$$

can be proven valid *if and only if*

$$A(t)\int_{t_0}^{t} A(\tau)\, d\tau = \int_{t_0}^{t} A(\tau)\, d\tau\, A(t)$$

that is, if and only if $A(t)$ and its integral commute. This is a very stringent condition for all but a first-order system and, as a rule, it is usually easiest to obtain the solution using simulation.

Most of the properties of the fixed transition matrix extend to the time-varying case:

$$\Phi(t,t_0) = I$$

$$\Phi^{-1}(t,t_0) = \Phi(t_0,t)$$

$$\Phi(t_2,t_1)\Phi(t_1,t_0) = \Phi(t_2,t_0)$$

$$\Phi(t,t_0) = A(t)\Phi(t,t_0)$$

8.4 Nonlinear Systems

The theory of fixed, linear, lumped-parameter systems is highly developed and provides a powerful set of techniques for control system analysis and design. In practice, however, all physical systems are nonlinear to some greater or lesser degree. The linearity of a physical system is usually only a convenient approximation, restricted to a certain range of operation. In addition, nonlinearities such as dead zones, saturation, or on–off action are sometimes introduced into control systems intentionally, either to obtain some advantageous performance characteristic or to compensate for the effects of other (undesirable) nonlinearities.

Unfortunately, while nonlinear systems are important, ubiquitous, and potentially useful, the theory of nonlinear differential equations is comparatively meager. Except for specific cases, closed-form solutions to nonlinear systems are generally unavailable. The only universally applicable method for the study of nonlinear systems is *simulation.* As described in Section 7, however, simulation is an experimental approach, embodying all of the attending limitations of experimentation.

A number of special techniques are available for the analysis of nonlinear systems. All of these techniques are in some sense approximate, assuming, for example, either a restricted range of operation over which nonlinearities are mild or the relative isolation of lower order subsystems. When used in conjunction with more complex simulation models, however, these techniques often provide insights and design concepts that would be difficult to discover through the use of simulation alone.[8]

Linear versus Nonlinear Behaviors

There are several fundamental differences between the behavior of linear and nonlinear systems that are especially important. These differences not only account for the increased

difficulty encountered in the analysis and design of nonlinear systems, but also imply entirely new types of behavior for nonlinear systems that are not possible for linear systems.

The fundamental property of linear systems is *superposition.* This property states that if $y_1(t)$ is the response of the system to $u_1(t)$ and $y_2(t)$ is the response of the system to $u_2(t)$, then the response of the system to the linear combination $a_1u_1(t) + a_2u_2(t)$ is the linear combination $a_1y_1(t) + a_2y_2(t)$. An immediate consequence of superposition is that the responses of a linear system to inputs differing only in amplitude is qualitatively the same. Since superposition does not apply to nonlinear systems, the responses of a nonlinear system to large and small changes may be fundamentally different.

This fundamental difference in linear and nonlinear behaviors has a second consequence. For a linear system, interchanging two elements connected in series does not affect the overall system behavior. Clearly, this cannot be true in general for nonlinear systems.

A third property peculiar to nonlinear systems is the potential existence of *limit cycles.* A linear oscillator oscillates at an amplitude that depends on its initial state. A limit cycle is an oscillation of fixed amplitude and period, independent of the initial state, that is unique to the nonlinear system.

A fourth property concerns the response of nonlinear systems to sinusoidal inputs. For a linear system, the response to sinusoidal input is a sinusoid of the same frequency, potentially differing only in magnitude and phase. For a nonlinear system, the output will in general contain other frequency components, including possibly harmonics, subharmonics, and aperiodic terms. Indeed, the response need not contain the input frequency at all.

Linearizing Approximations

Perhaps the most useful technique for analyzing nonlinear systems is to approximate these with linear systems. While many linearizing approximations are possible, linearization can frequently be achieved by considering small excursions of the system state about a reference trajectory. Consider the nonlinear state equation

$$\dot{x}(t) = f[x(t),u(t)]$$

together with a reference trajectory $x^0(t)$ and reference input $u^0(t)$ that together satisfy the state equation

$$\dot{x}^0(t) = f[x^0(t),u^0(t)]$$

Note that the simplest case is to choose a static equilibrium or *operating point* $\bar{x}$ as the reference "trajectory" such that $0 = t(\bar{x},0)$. The actual trajectory is then related to the reference trajectory by the relationships

$$x(t) = x^0(t) + \delta x(t)$$
$$u(t) = u^0(t) + \delta u(t)$$

where $\delta x(t)$ is some small perturbation about the reference state and $\delta u(t)$ is some small perturbation about the reference input. If these perturbations are indeed small, then applying Taylor's series expansion about the reference trajectory yields the linearized approximation

$$\delta\dot{x}(t) = A(t)\ \delta x(t) + B(t)\ \delta u(t)$$

where the state and distribution matrices are the *Jacobian matrices*

$$A(t) = \left[\begin{array}{cccc} \dfrac{\partial f_i}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \cdots & \dfrac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial f_n}{\partial x_1} & \dfrac{\partial f_n}{\partial x_2} & \cdots & \dfrac{\partial f_n}{\partial x_n} \end{array}\right]_{x(t)=x^0(t);\ u(t)=u^0(t)}$$

$$B(t) = \left[\begin{array}{cccc} \dfrac{\partial f_1}{\partial u_1} & \dfrac{\partial f_1}{\partial u_2} & \cdots & \dfrac{\partial f_1}{\partial u_m} \\ \dfrac{\partial f_2}{\partial u_1} & \dfrac{\partial f_2}{\partial u_2} & \cdots & \dfrac{\partial f_2}{\partial u_m} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial f_n}{\partial u_1} & \dfrac{\partial f_n}{\partial u_2} & \cdots & \dfrac{\partial f_n}{\partial u_m} \end{array}\right]_{x(t)=x^0(t);\ u(t)=u^0(t)}$$

If the reference trajectory is a fixed operating point $\bar{x}$, then the resulting linearized system is time invariant and can be solved analytically. If the reference trajectory is a function of time, however, then the resulting system is linear but time varying.

Describing Functions

The describing function method is an extension of the frequency transfer function approach of linear systems, most often used to determine the stability of limit cycles of systems containing nonlinearities. The approach is approximate and its usefulness depends on two major assumptions:

1. All the nonlinearities within the system can be aggregated mathematically into a single block, denoted as $N(M)$ in Fig. 29, such that the equivalent gain and phase associated with this block depend only on the amplitude M_d of the sinusoidal input $m(\omega t) = M \sin(\omega t)$ and are independent of the input frequency ω.

2. All the harmonics, subharmonics, and any dc component of the output of the nonlinear block are filtered out by the linear portion of the system such that the effective output of the nonlinear block is well approximated by a periodic response having the same fundamental period as the input.

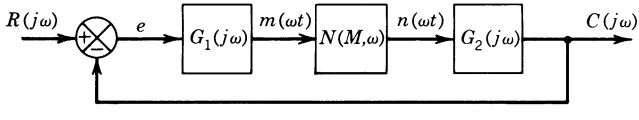


Figure 29 General nonlinear system for describing function analysis.

Although these assumptions appear to be rather limiting, the technique gives reasonable results for a large class of control systems. In particular, the second assumption is generally satisfied by higher order control systems with symmetric nonlinearities, since (a) symmetric nonlinearities do not generate dc terms, (b) the amplitudes of harmonics are generally small when compared with the fundamental term and subharmonics are uncommon, and (c) feedback within a control system typically provides low-pass filtering to further attenuate harmonics, especially for higher order systems. Because the method is relatively simple and can be used for systems of any order, describing functions have enjoyed wide practical application.

The describing function of a nonlinear block is defined as the ratio of the fundamental component of the output to the amplitude of a sinusoidal input. In general, the response of the nonlinearity to the input

$$m(\omega t) = M \sin \omega t$$

is the output

$$n(\omega t) = N_1 \sin(\omega t + \phi_1) + N_2 \sin(2\omega t + \phi_2) + N_3 \sin(3\omega t + \phi_3) + \cdots$$

and, hence, the describing function for the nonlinearity is defined as the complex quantity

$$N(M) = \frac{N_1}{M} e^{j\phi_1}$$

Derivation of the approximating function typically proceeds by representing the fundamental frequency by the Fourier series coefficients

$$A_1(M) = \frac{2}{T} \int_{-T/2}^{T/2} n(\omega t) \cos \omega t \, d(\omega t)$$

$$B_1(M) = \frac{2}{T} \int_{-T/2}^{T/2} n(\omega t) \sin \omega t \, d(\omega t)$$

The describing function is then written in terms of these coefficients as

$$N(M) = \frac{B_1(M)}{M} + j \frac{A_1(M)}{M} = \left[\left(\frac{B_1(M)}{M} \right)^2 + \left(\frac{A_1(M)}{M} \right)^2 \right]^{1/2} \exp \left[j \tan^{-1} \left(\frac{A_1(M)}{B_1(M)} \right) \right]$$

Note that if $n(\omega t) = -n(-\omega t)$, then the describing function is odd, $A_1(M) = 0$, and there is no phase shift between the input and output. If $n(\omega t) = n(-\omega t)$, then the function is even, $B_1(M) = 0$, and the phase shift is $\pi/2$.

The describing functions for a number of typical nonlinearities are given in Fig. 30. Reference 9 contains an extensive catalog. The following derivation for a dead-zone nonlinearity demonstrates the general procedure for deriving a describing function. For the saturation element depicted in Fig. 30*a*, the relationship between the input $m(\omega t)$ and output $n(\omega t)$ can be written as

$$n(\omega t) = \begin{cases} 0 & \text{for} \quad -D < m < D \\ K_1 M(\sin \omega t - \sin \omega_1 t) & \text{for} \quad m > D \\ K_1 M(\sin \omega t + \sin \omega_1 t) & \text{for} \quad m < -D \end{cases}$$

Since the function is odd, $A_1 = 0$. By the symmetry over the four quarters of the response period,

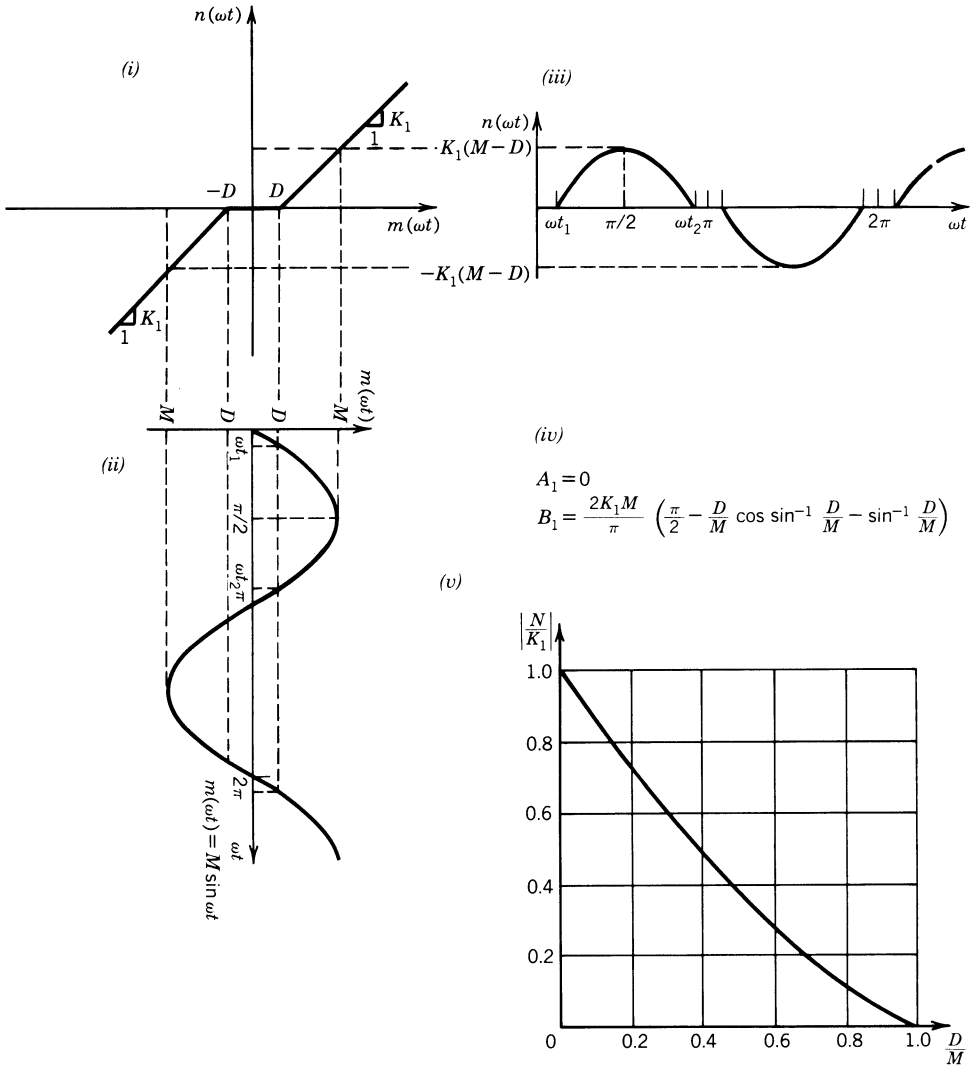


Figure 30*a* Describing functions for typical nonlinearities (after Refs. 9 and 10). Dead-zone nonlinearity: (*i*) nonlinear characteristic; (*ii*) sinusoidal input wave shape; (*iii*) output wave shape; (*iv*) describing-function coefficients; (*v*) normalized describing function.

$$B_1 = 4\left[\frac{2}{\pi/2}\int_0^{\pi/2} n(\omega t)\sin\omega t\, d(\omega t)\right]$$

$$= \frac{4}{\pi}\left[\int_0^{\omega t_1} (0)\sin\omega t\, d(\omega t) + \int_{\omega t_1}^{\pi/2} K_1 M(\sin\omega t - \sin\omega_1 t)\sin\omega t\, d(\omega t)\right]$$

where $\omega t_1 = \sin^{-1}(D/M)$. Evaluating the integrals and dividing by M yields the describing function listed in Fig. 30.

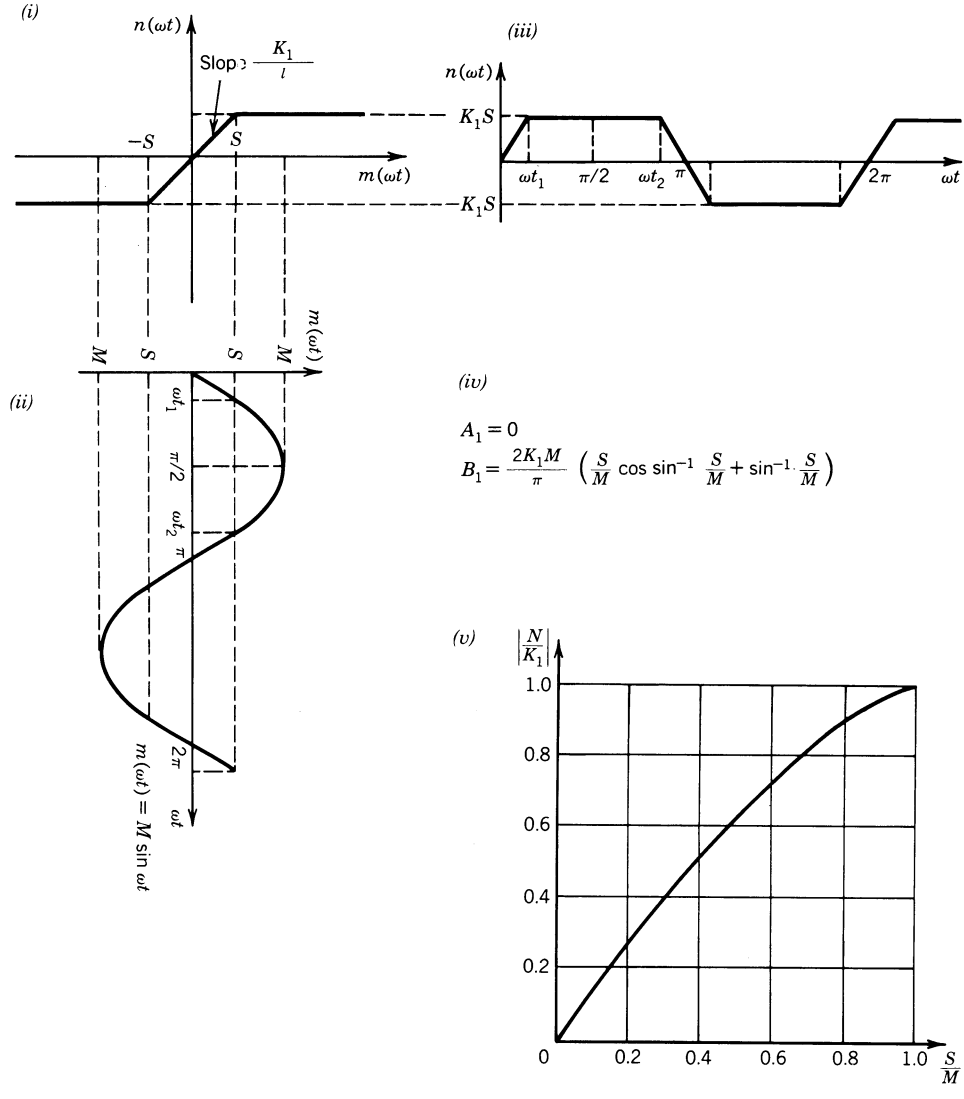


Figure 30*b* Saturation nonlinearity: (*i*) nonlinear characteristic; (*ii*) sinusoidal input wave shape; (*iii*) output wave shape; (*iv*) describing-function coefficients; (*v*) normalized describing function.

Phase-Plane Method

The *phase-plane method* is a graphical application of the state-space approach used to characterize the free response of second-order nonlinear systems. While any convenient pair of state variables can be used, the *phase variables* originally were taken to be the displacement and velocity of the mass of a second-order mechanical system. Using the two state variables as the coordinate axis, the transient response of a system is captured on the *phase plane* as the plot of one variable against the other, with time implicit on the resulting curve. The curve for a specific initial condition is called a *trajectory* in the phase plane; a representative sample of trajectories is called the *phase portrait* of the system. The phase portrait is a

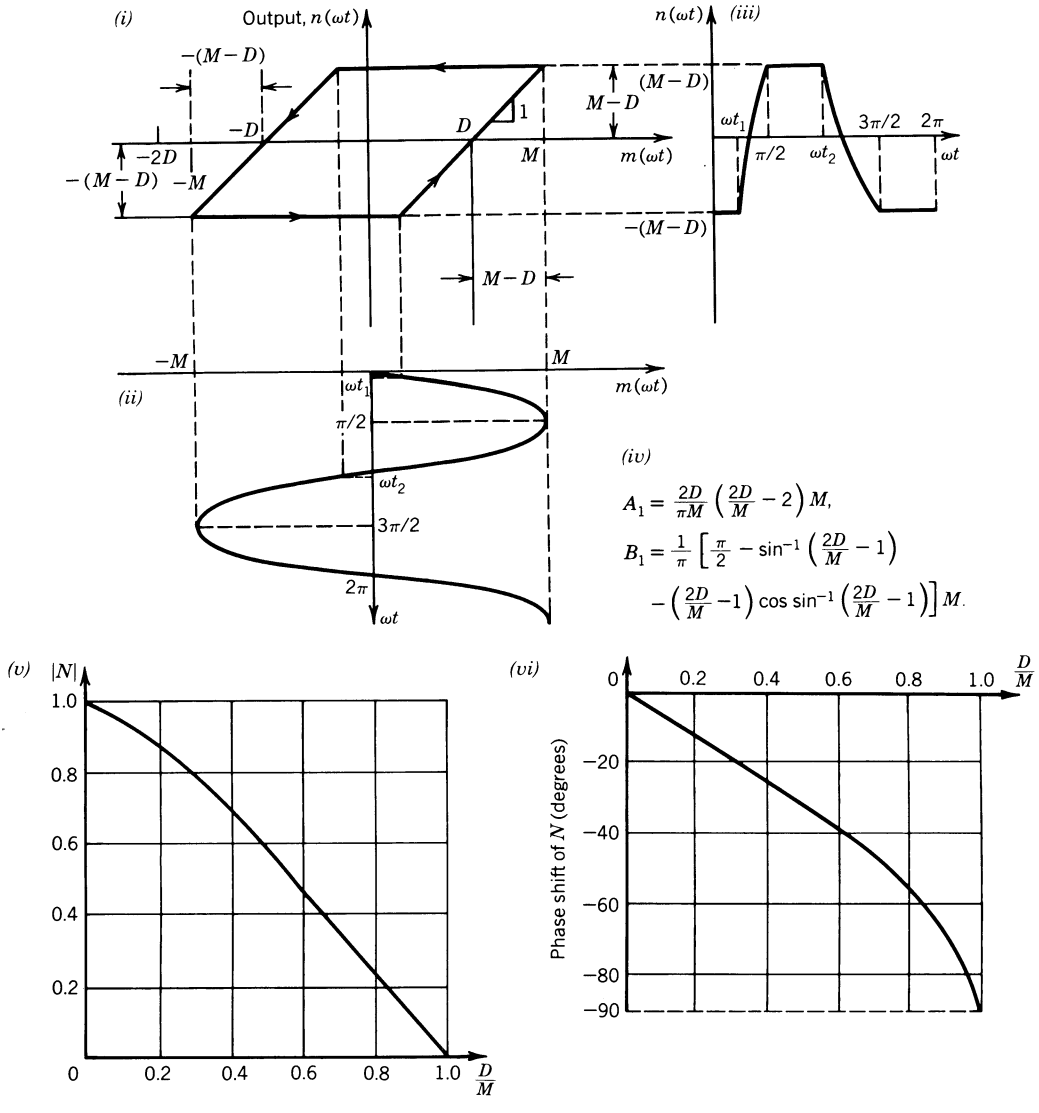


Figure 30*c* Backlash nonlinearity: (*i*) nonlinear characteristic; (*ii*) sinusoidal input wave shape; (*iii*) output wave shape; (*iv*) describing-function coefficients; (*v*) normalized amplitude characteristics for the describing function; (*vi*) normalized phase characteristics for the describing function.

compact and readily interpreted summary of the system response. Phase portraits for a sample of typical nonlinearities are shown in Fig. 31.

Four methods can be used to construct a phase portrait: (1) direct solution of the differential equation, (2) the graphical *method of isoclines,* (3) transformation of the second-order system (with time as the independent variable) into an equivalent first-order system (with one of the phase variables as the independent variable), and (4) numerical solution using simulation. The first and second methods are usually impractical; the third and fourth methods are frequently used in combination. For example, consider the second-order model

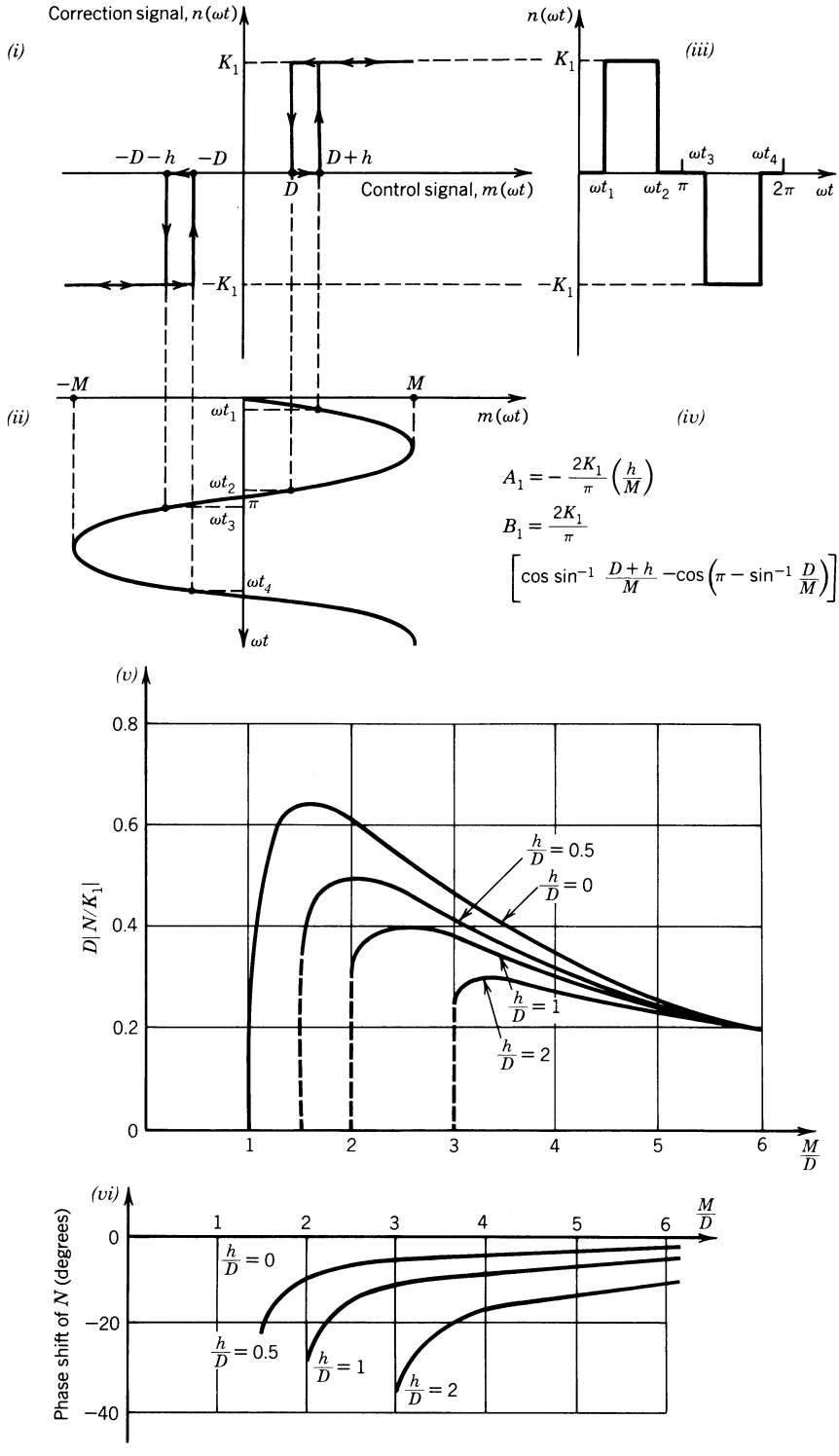


Figure 30*d* Three-position on–off device with hysteresis: (*i*) nonlinear characteristic; (*ii*) sinusoidal input wave shape; (*ii*) output wave shape; (*iv*) describing-function coefficients; (*v*) normalized amplitude characteristics for the describing function; (*vi*) normalized phase characteristics for the describing function.

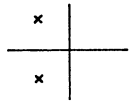
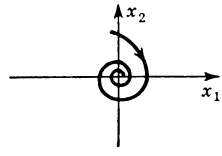
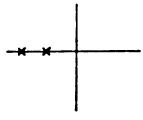
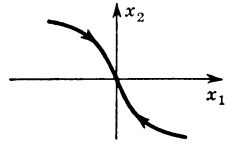
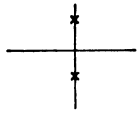
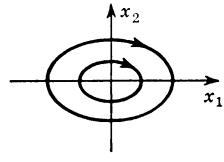
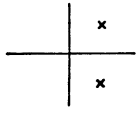
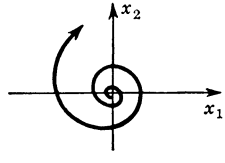
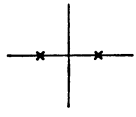
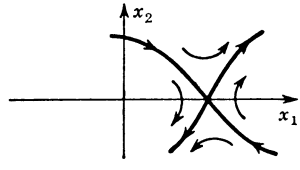
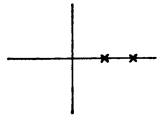
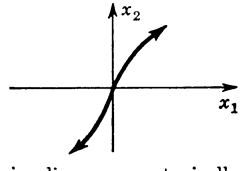
Name	Roots	Sketch
Stable focus or spiral	Damped complex conjugate 	 Trajectories spiral asymptotically to focus
Stable node	Stable real roots 	 Trajectories approach node monotonically
Vortex or center (structurally unstable)	Imaginary roots 	 Conservative system or oscillator
Unstable focus	Complex conjugate with positive real part 	
Saddle point	A point of unstable equilibrium 	
Unstable node	Unstable real roots 	 Trajectories diverge monotonically from node

Figure 31 Typical phase-plane plots for second-order systems.[9]

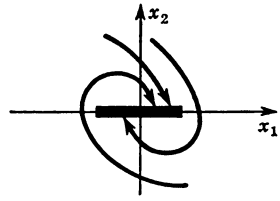
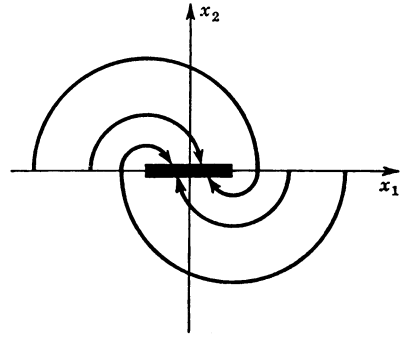
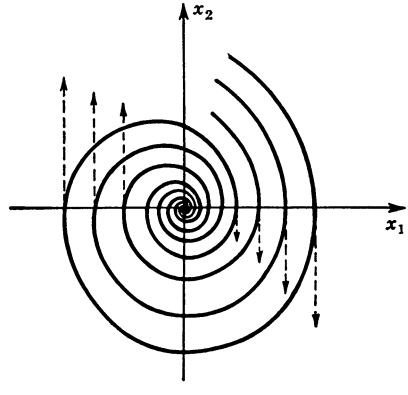
Name	Roots	Sketch
Spiral with nonlinearity viscous damping and coulomb friction		 Spirals are warped by coulomb friction
Nonlinear system with coulomb damping and no viscous damping		 Semicircles center at ends of coulomb line
Spiral with backlash		

Figure 31 (*Continued*)

$$\frac{dx_1}{dt} = f_1(x_1, x_2) \qquad \frac{dx_2}{dt} = f_2(x_1, x_2)$$

Dividing the second equation by the first and eliminating the dt terms yield

$$\frac{dx_2}{dx_1} = \frac{f_2(x_1, x_2)}{f_1(x_1, x_2)}$$

This first-order equation describes the phase-plane trajectories. In many cases it can be solved analytically. If not, it always can be simulated.

The phase-plane method complements the describing-function approach. A describing function is an approximate representation of the sinusoidal response for systems of any order, while the phase plane is an exact representation of the (free) transient response for first- and second-order systems. Of course, the phase-plane method theoretically can be extended for higher order systems, but the difficulty of visualizing the nth-order state space typically makes such a direct extension impractical. An approximate extension of the method has been used with some considerable success,[8] however, in order to explore and validate the relationships among pairs of variables in complex simulation models. The approximation is based on the assumptions that the paired variables define a second-order subsystem which, for the purposes of analysis, is weakly coupled to the remainder of the system.

8.5 Discrete and Hybrid Systems

A *discrete-time system* is one for which the dependent variables are defined only at distinct instants of time. Discrete-time models occur in the representation of systems that are inherently discrete, in the analysis and design of digital measurement and control systems, and in the numerical solution of differential equations (see Section 7). Because most control systems are now implemented using digital computers (especially microprocessors), discrete-time models are extremely important in dynamic systems analysis. The discrete-time nature of a computer's sampling of continuous physical signals also leads to the occurrence of *hybrid systems,* that is, systems that are in part discrete and in part continuous. Discrete-time models of hybrid systems are called *sampled-data systems.*

Difference Equations

Dynamic models of discrete-time systems most naturally take the form of *difference equations.* The I/O form of an nth-order difference equation model is

$$f[y(k+n), y(k+n-1), \ldots, y(k), u(k+n-1), \ldots, u(k)] = 0$$

which expresses the dependence of the $(k+n)$th value of the output, $y(k+n)$, on the n preceding values of the output y and input u. For a linear system, the I/O form can be written as

$$y(k+n) + a_{n-1}(k)y(k+n-1) + \cdots + a_1(k)y(k+1) + a_0(k)y(k)$$
$$= b_{n-1}(k)u(k+n-1) + \cdots + b_0(k)u(k)$$

In state-variable form, the discrete-time model is the vector difference equation

$$x(k+1) = f[x(k), u(k)]$$
$$y(k) = g[x(k), u(k)]$$

where x is the state vector, u is the vector of inputs, and y is the vector of outputs. For a linear system, the discrete state-variable form can be written as

$$x(k+1) = A(k)x(k) + B(k)u(k)$$
$$y(k) = C(k)x(k) + D(k)u(k)$$

The mathematics of difference equations parallels that of differential equations in many important respects. In general, the concepts applied to differential equations have direct analogies for difference equations, although the mechanics of their implementation may vary (see Ref. 11 for a development of dynamic modeling based on difference equations). One important difference is that the general solution of nonlinear and time-varying difference equations can usually be obtained through *recursion.* For example, consider the discrete nonlinear model

$$y(k+1) = \frac{y(k)}{1 + y(k)}$$

Recursive evaluation of the equation beginning with the initial condition $y(0)$ yields

$$y(1) = \frac{y(0)}{1 + y(0)}$$
$$y(2) = \frac{y(1)}{1 + y(1)} = \left[\frac{y(0)}{1 + y(0)}\right] \Big/ \left[1 + \frac{y(0)}{1 + y(0)}\right] = \frac{y(0)}{1 + 2y(0)}$$
$$y(3) = \frac{y(2)}{1 + y(2)} = \frac{y(0)}{1 + 3y(0)}$$
$$\vdots$$

the pattern of which reveals, by induction,

$$y(k) = \frac{y(0)}{1 + ky(0)}$$

as the general solution.

Uniform Sampling

Uniform sampling is the most common mathematical approach to *A/D conversion,* that is, to extracting the discrete-time approximation $y^*(k)$ of the form

$$y^*(k) = y(t = kT)$$

from the continuous-time signal $y(t)$, where T is a constant interval of time called the *sampling period.* If the sampling period is too large, however, it may not be possible to represent the continuous signal accurately. The *sampling theorem* guarantees that $y(t)$ can be reconstructed from the uniformly sampled values $y^*(k)$ if the sampling period satisfies the inequality

$$T \leq \frac{\pi}{\omega_u}$$

where ω_u is the highest frequency contained in the Fourier transform $Y(\omega)$ of $y(t)$, that is, if

$$Y(\omega) = 0 \qquad \text{for all} \quad \omega > \omega_u$$

The Fourier transform of a signal is defined to be

$$\mathfrak{F}[y(t)] = Y(\omega) = \int_{-\infty}^{\infty} y(t)e^{-j\omega t}\,dt$$

Note that if $y(t) = 0$ for $t \geq 0$, and if the region of convergence for the Laplace transform includes the imaginary axis, then the Fourier transform can be obtained from the Laplace transform as

$$Y(\omega) = [Y(s)]_{s=j\omega}$$

For cases where it is impossible to determine the Fourier transform analytically, such as when the signal is described graphically or by a table, numerical solution based on the *fast Fourier transform* (*FFT*) *algorithm* is usually satisfactory.

In general, the condition $T \leq \pi/\omega_u$ cannot be satisfied exactly, since most physical signals have no finite upper frequency ω_u. A useful approximation is to define the upper frequency as the frequency for which 99% of the signal "energy" lies in the frequency spectrum $0 \leq \omega \leq \omega_u$. This approximation is found from the relation

$$\int_0^{\omega_u} |Y(\omega)|^2\,d\omega = 0.99 \int_0^{\infty} |Y(\omega)|^2\,d\omega$$

where the square of the amplitude of the Fourier transform $|Y(\omega)|^2$ is said to be the *power spectrum* and its integral over the entire frequency spectrum is referred to as the "energy" of the signal. Using a sampling frequency 2–10 times this approximate upper frequency (depending on the required factor of safety) and inserting a low-pass filter (called a *guard filter*) before the sampler to eliminate frequencies above the *Nyquist frequency* π/T usually lead to satisfactory results.[4]

The z-Transform

The *z-transform* permits the development and application of transfer functions for discrete-time systems, in a manner analogous to continuous-time transfer functions based on the Laplace transform. A discrete signal may be represented as a series of impulses

$$\begin{aligned} y^*(t) &= y(0)\delta(t) + y(1)\delta(t - T) + y(2)\delta(t - 2T) + \cdots \\ &= \sum_{k=0}^{N} y(k)\delta(t - kT) \end{aligned}$$

where $y(k) = y^*(t = kT)$ are the values of the discrete signal, $\delta(t)$ is the unit impulse function, and N is the number of samples of the discrete signal. The Laplace transform of the series is

$$Y^*(s) = \sum_{k=0}^{N} y(k)e^{-ksT}$$

where the shifting property of the Laplace transform has been applied to the pulses. Defining the *shift* or *advance operator* as $z = e^{sT}$, $Y^*(s)$ may now be written as a function of z:

$$Y^*(z) = \sum_{k=0}^{N} \frac{y(k)}{z^k} = \mathfrak{Z}\,[y(t)]$$

where the transformed variable $Y^*(z)$ is called the z-transform of the function $y^*(t)$. The inverse of the shift operator $1/z$ is called the *delay operator* and corresponds to a time delay of T.

The z-transforms for many sampled functions can be expressed in closed form. A listing of the transforms of several commonly encountered functions is given in Table 12. Properties of the z-transform are listed in Table 13.

Pulse Transfer Functions

The transfer function concept developed for continuous systems has a direct analog for sampled-data systems. For a continuous system with sampled output $u(t)$ and sampled input $y(t)$, the *pulse* or *discrete transfer function* $G(z)$ is defined as the ratio of the z-transformed output $Y(z)$ to the z-transformed input $U(z)$, assuming zero initial conditions. In general, the pulse transfer function has the form

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-1} + \cdots + a_n z^{-n}}$$

Zero-Order Hold

The *zero-order data hold* is the most common mathematical approach to *D/A conversion,* that is, to creating a piecewise continuous approximation $u(t)$ of the form

Table 12 z-Transform Pairs

	$X(s)$	$x(t)$ or $x(k)$	$X(z)$
1	1	$\delta(t)$	1
2	e^{-kTs}	$\delta(t - kT)$	z^{-k}
3	$\frac{1}{s}$	$1(t)$	$\frac{z}{z-1}$
4	$\frac{1}{s^2}$	t	$\frac{Tz}{(z-1)^2}$
5	$\frac{1}{s+a}$	e^{-at}	$\frac{z}{z-e^{-aT}}$
6	$\frac{a}{s(s+a)}$	$1 - e^{-at}$	$\frac{(1-e^{-aT})z}{(z-1)(z-e^{-aT})}$
7	$\frac{\omega}{s^2+\omega^2}$	$\sin \omega t$	$\frac{z \sin \omega T}{z^2 - 2z \cos \omega T + 1}$
8	$\frac{s}{s^2+\omega^2}$	$\cos \omega t$	$\frac{z(z - \cos \omega T)}{z^2 - 2z \cos \omega T + 1}$
9	$\frac{1}{(s+a)^2}$	te^{-at}	$\frac{Tze^{-aT}}{(z-e^{-aT})^2}$
10	$\frac{\omega}{(s+a)^2+\omega^2}$	$e^{-at} \sin \omega t$	$\frac{ze^{-aT} \sin \omega T}{z^2 - 2ze^{-aT} \cos \omega T + e^{-2aT}}$
11	$\frac{s+a}{(s+a)^2+\omega^2}$	$e^{-at} \cos \omega t$	$\frac{z^2 - ze^{-aT} \cos \omega T}{z^2 - 2ze^{-aT} \cos \omega T + e^{-2aT}}$
12	$\frac{2}{s^3}$	t^2	$\frac{T^2 z(z+1)}{(z-1)^3}$
13		a	$\frac{z}{z-a}$
14		$a^k \cos k\pi$	$\frac{z}{z+a}$

Table 13 z-Transform Properties

	$x(t)$ or $x(k)$	$\mathfrak{Z}[x(t)]$ or $\mathfrak{Z}[x(k)]$
1	$ax(t)$	$aX(z)$
2	$x_1(t) + x_2(t)$	$X_1(z) + X_2(z)$
3	$x(t + T)$ or $x(k + 1)$	$zX(z) - zx(0)$
4	$x(t + 2T)$	$z^2X(z) - z^2x(0) - zx(T)$
5	$x(k + 2)$	$z^2X(z) - z^2x(0) - zx(1)$
6	$x(t + kT)$	$z^kX(z) - z^kx(0) - z^{k-1}x(T) - \cdots - zx(kT - T)$
7	$x(k + m)$	$z^mX(z) - z^mx(0) - z^{m-1}x(1) - \cdots - zx(m - 1)$
8	$tx(t)$	$-Tz\dfrac{d}{dz}[X(z)]$
9	$kx(k)$	$-z\dfrac{d}{dz}[X(z)]$
10	$e^{-at}x(t)$	$X(ze^{aT})$
11	$e^{-ak}x(k)$	$X(ze^{a})$
12	$a^kx(k)$	$X\left(\dfrac{z}{a}\right)$
13	$ka^kx(k)$	$-z\dfrac{d}{dz}\left[X\left(\dfrac{z}{a}\right)\right]$
14	$x(0)$	$\lim\limits_{z\to\infty} X(z)$ if the limit exists
15	$x(\infty)$	$\lim\limits_{z\to 1}[(z - 1)X(z)]$ if $\dfrac{z - 1}{z}X(z)$ is analytic on and outside the unit circle
16	$\sum\limits_{k=0}^{\infty} x(k)$	$X(1)$
17	$\sum\limits_{k=0}^{n} x(kT)y(nT - kT)$	$X(z)Y(z)$

$$u(t) = u^*(k) \qquad \text{for} \quad kT \le t < (k + 1)T$$

from the discrete-time signal $u^*(k)$, where T is the period of the hold. The effect of the zero-order hold is to convert a sequence of discrete impulses into a staircase pattern, as shown in Fig. 32. The transfer function of the zero-order hold is

$$G(s) = \frac{1}{s}(1 - e^{-Ts}) = \frac{1 - z^{-1}}{s}$$

Using this relationship, the pulse transfer function of the sampled-data system shown in Fig. 33 can be derived as

$$G(z) = (1 - z^{-1})\mathfrak{Z}\left[\mathfrak{L}^{-1}\frac{G(s)}{s}\right]$$

The continuous system with transfer function $G(s)$ has a sampler and a zero-order hold at its input and a sampler at its output. This is a common configuration in many computer control applications.

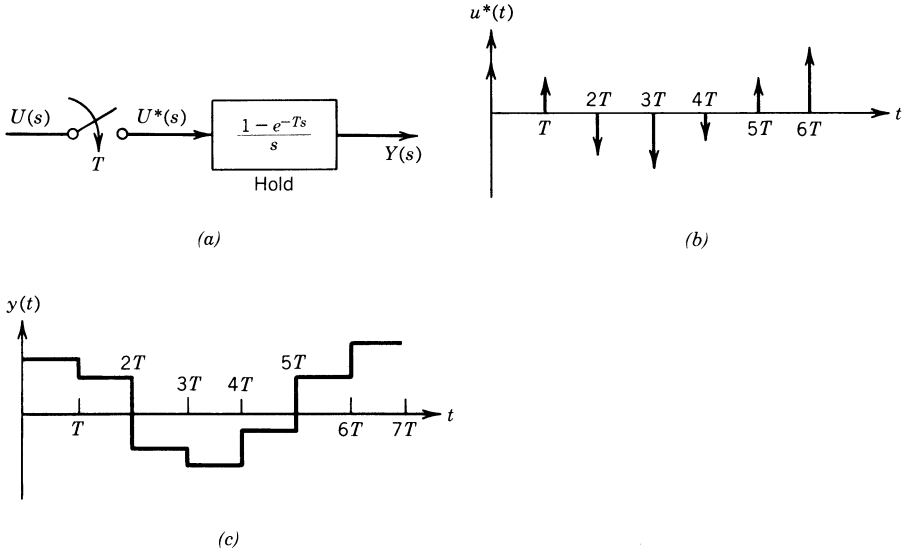


Figure 32 Zero-order hold: (*a*) block diagram of hold with a sampler, (*b*) sampled input sequence, (*c*) analog output for the corresponding input sequence.[4]

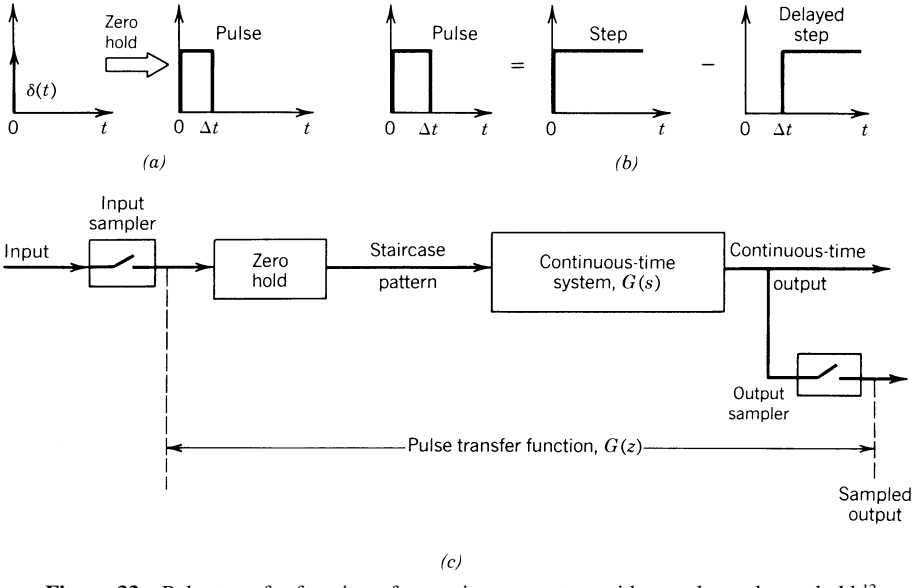


Figure 33 Pulse transfer function of a continuous system with sampler and zero hold.[12]

REFERENCES

1. J. L. Shearer, A. T. Murphy, and H. H. Richardson, *Introduction to System Dynamics,* Addison-Wesley, Reading, MA, 1971.

2. E. O. Doebelin, *System Dynamics: Modeling, Analysis, Simulation, and Design,* Merrill, Columbus, OH, 1998.

3. C. M. Close, D. K. Frederick, and J. C. Newell, *Modeling and Analysis of Dynamic Systems,* 3rd ed., Houghton Mifflin, Boston, 2001.

4. W. J. Palm III, *Modeling, Analysis, and Control of Dynamic Systems,* 2nd ed., Wiley, New York, 2000.

5. G. J. Thaler and R. G. Brown, *Analysis and Design of Feedback Control Systems,* 2nd ed., McGraw-Hill, New York, 1960.

6. G. A. Korn and J. V. Wait, *Digital Continuous System Simulation,* Prentice-Hall, Englewood Cliffs, NJ, 1975.

7. B. C. Kuo and F. Golnaraghi, *Automatic Control Systems,* 8th ed., Prentice-Hall, Englewood Cliffs, NJ, 2002.

8. W. Thissen, ''Investigation into the World3 Model: Lessons for Understanding Complicated Models,'' *IEEE Transactions on Systems, Man, and Cybernetics,* **SMC-8**(3) (1978) pp. 183–193.

9. J. E. Gibson, *Nonlinear Automatic Control,* McGraw-Hill, New York, 1963.

10. S. M. Shinners, *Modern Control System Theory and Design,* 2nd ed., Wiley, New York, 1998.

11. D. G. Luenberger, *Introduction to Dynamic Systems: Theory, Models, and Applications,* Wiley, New York, 1979.

12. D. M. Auslander, Y. Takahashi, and M. J. Rabins, *Introducing Systems and Control,* McGraw-Hill, New York, 1974.

BIBLIOGRAPHY

Bateson, R. N., *Introduction to Control System Technology,* Prentice-Hall, Englewood Cliffs, NJ, 2001.

Bishop, R. H., and R. C. Dorf, *Modern Control Systems,* 10th ed., Prentice-Hall, 2004.

Brogan, W. L., *Modern Control Theory,* 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1991.

Cannon, Jr., R. H., *Dynamics of Physical Systems,* McGraw-Hill, New York, 1967.

DeSilva, C. W., *Control Sensors and Actuators,* Prentice-Hall, Englewood Cliffs, NJ, 1989.

Doebelin, E. O., *Measurement Systems,* 5th ed., McGraw-Hill, New York, 2004.

Franklin, G. F., J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems,* 4th ed., Addison-Wesley, Reading, MA, 2001.

Grace, A., A. J. Laub, J. N. Little, and C. Thompson, *Control System Toolbox Users Guide,* The Mathworks, Natick, MA, 1990.

Hartley, T. T., G. O. Beale, and S. P. Chicatelli, *Digital Simulation of Dynamic Systems: A Control Theory Approach,* Prentice-Hall, Englewood Cliffs, NJ, 1994.

Kheir, N. A., *Systems Modeling and Computer Simulation,* 2nd ed., Marcel Dekker, New York, 1996.

Kelton, W. D., R. P. Sadowski, and D. T. Sturrock, *Simulation with Arena,* 3rd ed., McGraw-Hill, New York, 2004.

Lay, D. C., *Linear Algebra and Its Applications,* 3rd ed., Addison-Wesley, Reading, MA, 2006.

Ljung, L., and T. Glad, *Modeling Simulation of Dynamic Systems,* Prentice-Hall, Englewood Cliffs, NJ, 1994.

The MathWorks on-line at ⟨www.mathworks.com⟩ NJ, 1995.

Phillips, C. L., and R. D. Harbor, *Feedback Control Systems,* 4th ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.

Phillips, C. L., and H. T. Nagle, *Digital Control System Analysis and Design,* 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1995.

Van Loan, C., *Computational Frameworks for the Fast Fourier Transform,* SIAM, Philadelphia, PA, 1992.

Wolfram, S., *Mathematica Book,* 5th ed., Wolfram Media, Inc., 2003.