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 $\log \omega$. Bode diagrams for normalized first- and second-order systems are given in Fig. 27.23. 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 $\log \omega$. Polar plots and Bode and Nichols diagrams for common transfer functions are given in Table 27.8.

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.

27.6 STATE-VARIABLE METHODS

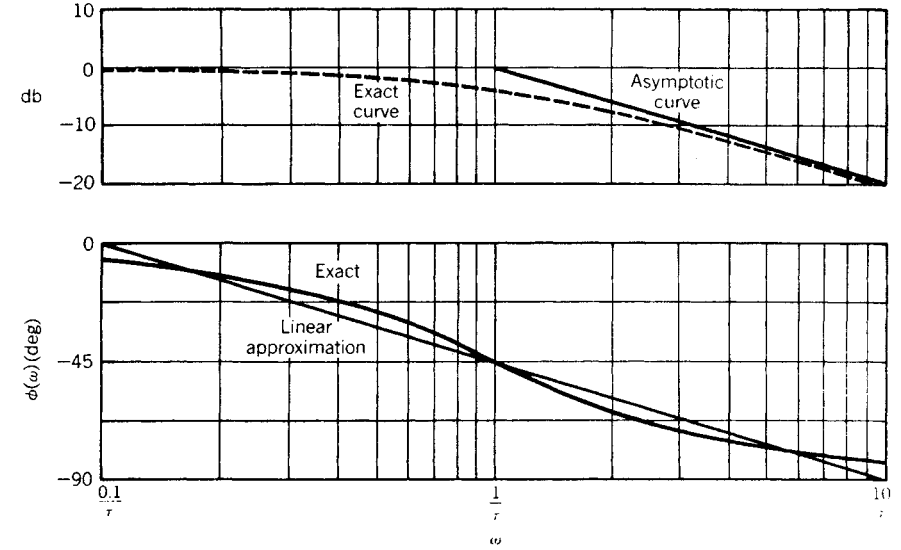
State-variable methods use the vector state and output equations introduced in Section 27.4 for analysis of dynamic systems directly in the time domain. These methods have several advantages over transform methods. First, state-variable methods are particularly advantageous for the study of multivariable (multiple input/multiple output) systems. Second, state-variable methods are more naturally extended for the study of linear time-varying and nonlinear systems. Finally, state-variable methods are readily adapted to computer simulation studies.

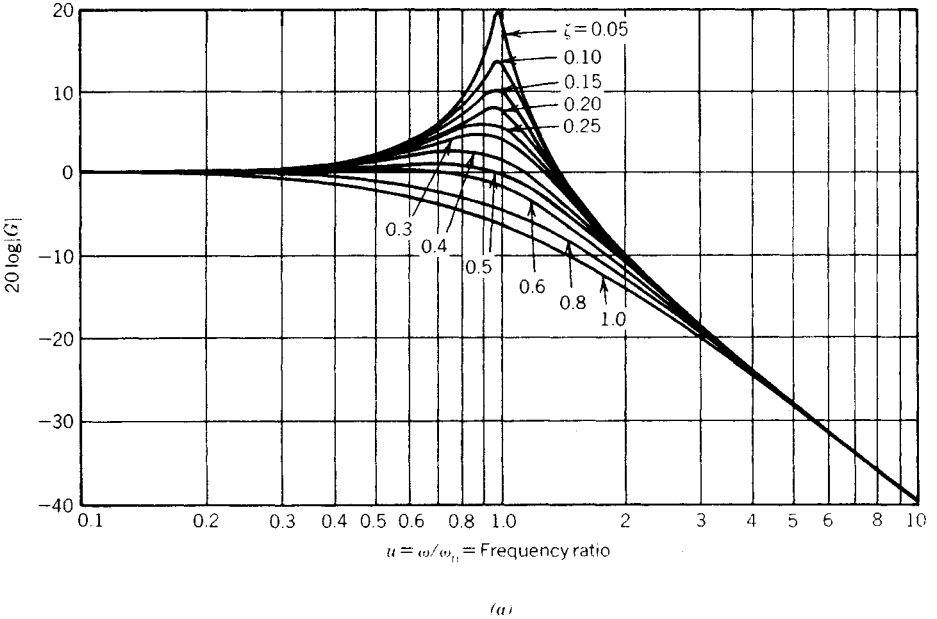
27.6.1 Solution of the State Equation

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

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

The solution to this system is



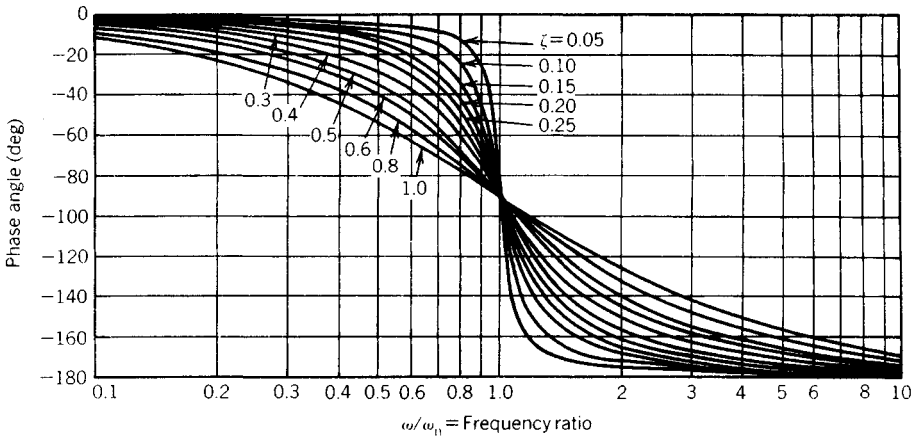


(a)

Fig. 27.23 Bode diagrams for normalized (*a*) first-order and (*b*) second-order systems.

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

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



(b)

Fig. 27.23 *(Continued)*

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

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

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

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

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

27.6.2 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

Table 27.8 Transfer Function Plots for Representative Transfer Functions[5]

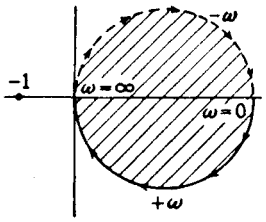
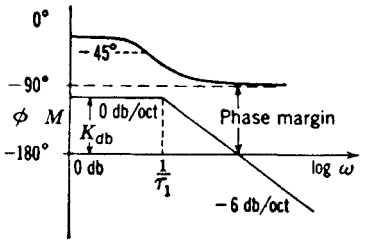
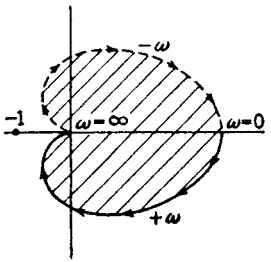
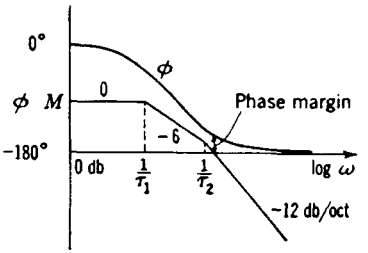
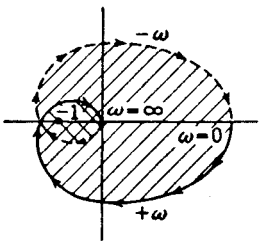
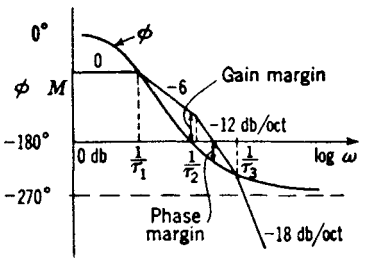
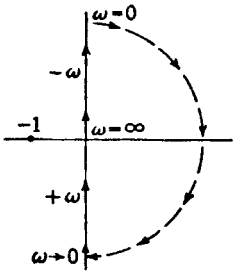
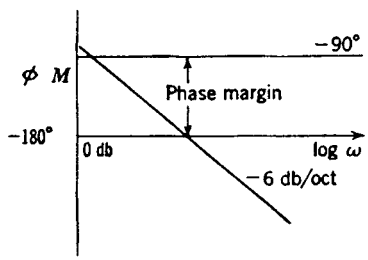
$G(s)$	Polar plot	Bode diagram
1. $\frac{K}{s\tau_1 + 1}$		
2. $\frac{K}{(s\tau_1 + 1)(s\tau_2 + 1)}$		
3. $\frac{K}{(s\tau_1 + 1)(s\tau_2 + 1)(s\tau_3 + 1)}$		
4. $\frac{K}{s}$		

Table 27.8 (*Continued*)

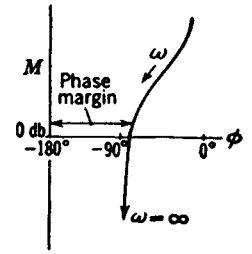
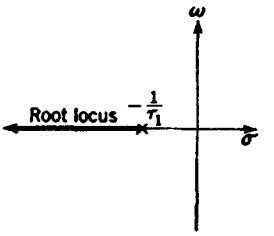
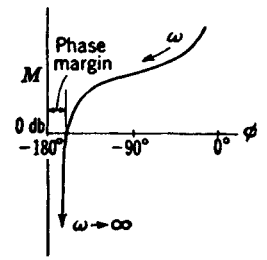
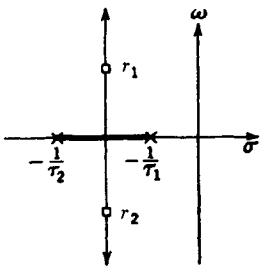
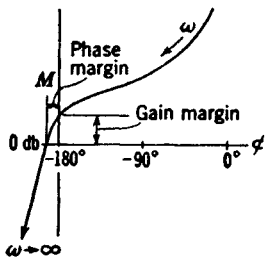
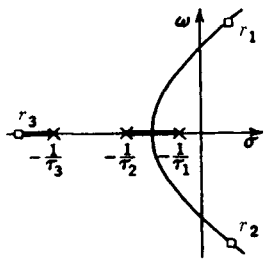
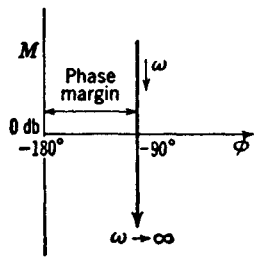
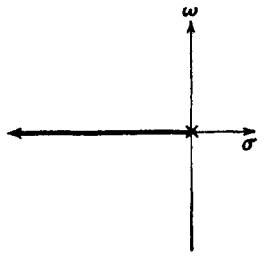
Nichols diagram	Root locus	Comments
		Stable; gain margin = ∞
		Elementary regulator; stable; gain margin = ∞
		Regulator with additional energy-storage component; unstable, but can be made stable by reducing gain
		Ideal integrator; stable

Table 27.8 (*Continued*)

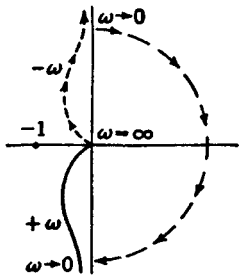
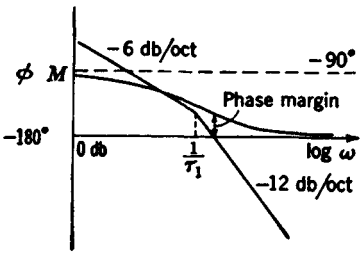
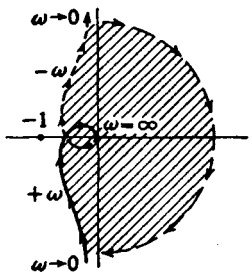
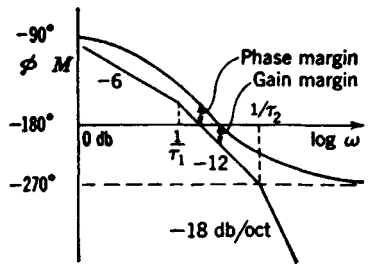
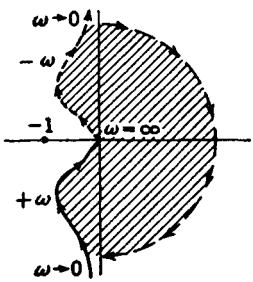
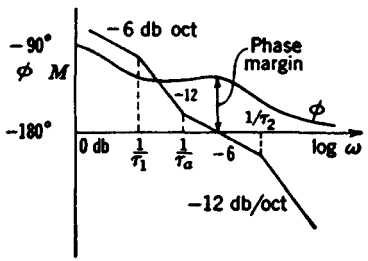
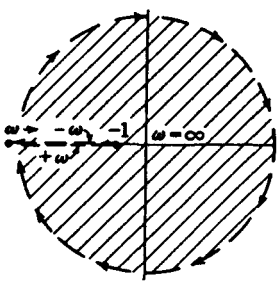
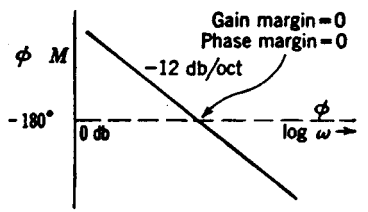
	$G(s)$	Polar plot	Bode diagram
5.	$\frac{K}{s(s\tau_1 + 1)}$		
6.	$\frac{K}{s(s\tau_1 + 1)(s\tau_2 + 1)}$		
7.	$\frac{K(s\tau_a + 1)}{s(s\tau_1 + 1)(s\tau_2 + 1)}$		
8.	$\frac{K}{s^2}$		

Table 27.8 (*Continued*)

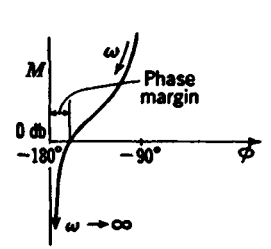
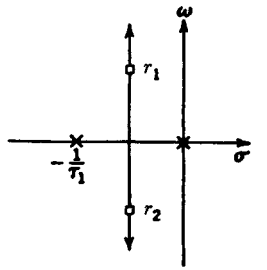
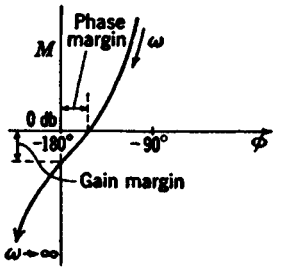
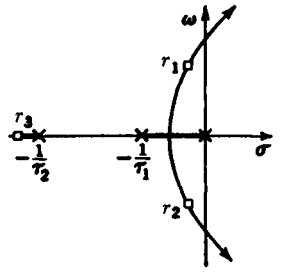
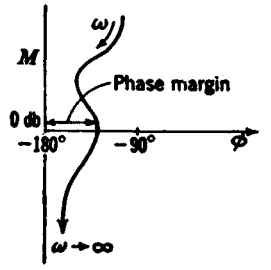
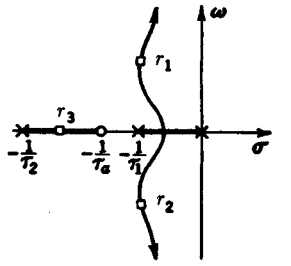
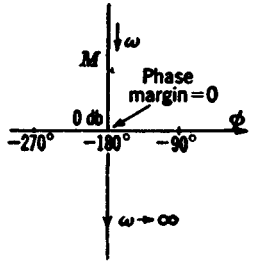
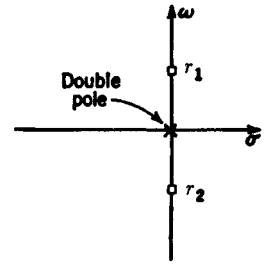
Nichols diagram	Root locus	Comments
		Elementary instrument servo; inherently stable; gain margin = ∞
		Instrument servo with field-control motor or power servo with elementary Ward-Leonard drive; stable as shown, but may become unstable with increased gain
		Elementary instrument servo with phase-lead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

Table 27.8 (*Continued*)

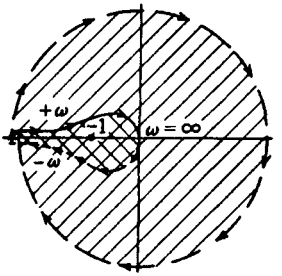
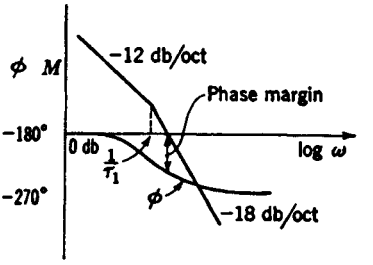
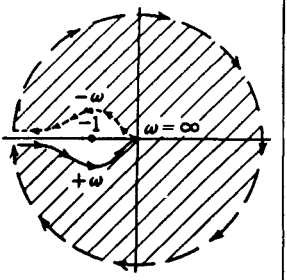
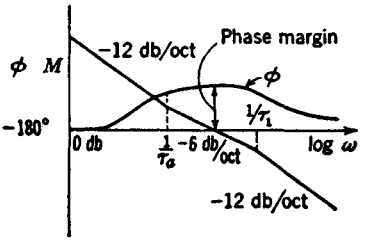
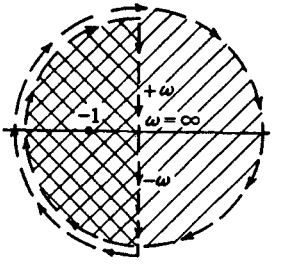
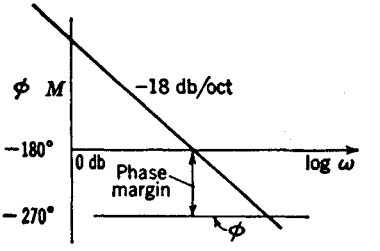
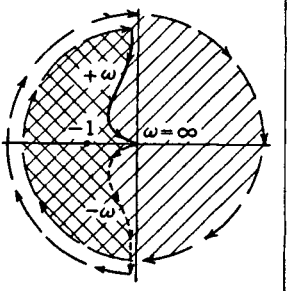
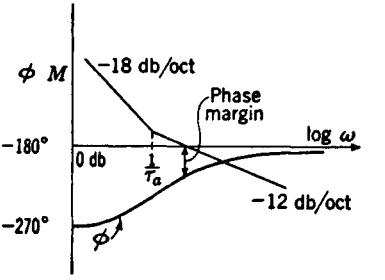
	$G(s)$	Polar plot	Bode diagram
9.	$\frac{K}{s^2(s\tau_1+1)}$		
10.	$\frac{K(s\tau_a+1)}{s^2(s\tau_1+1)}$ $\tau_a > \tau_1$		
11.	$\frac{K}{s^3}$		
12.	$\frac{K(s\tau_a+1)}{s^3}$		

Table 27.8 (*Continued*)

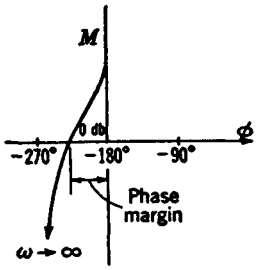
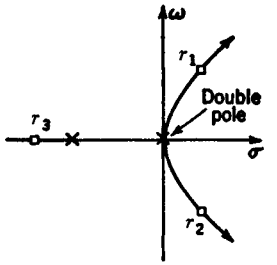
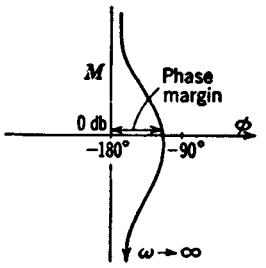
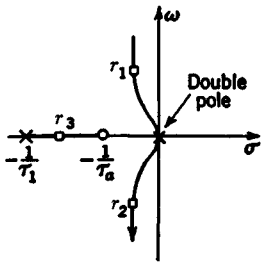
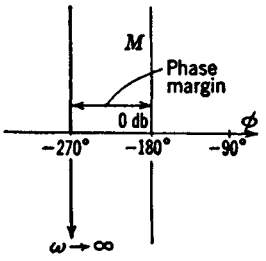
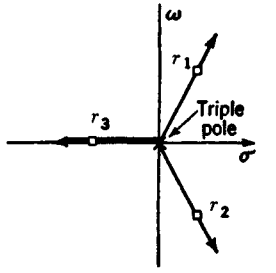
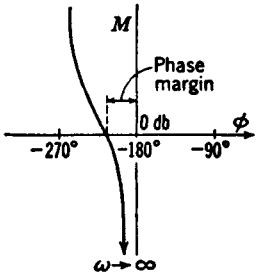
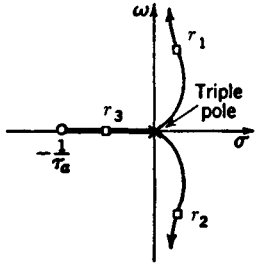
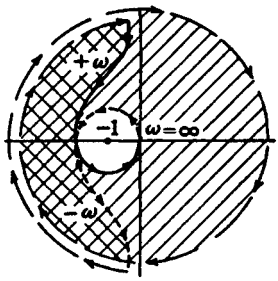
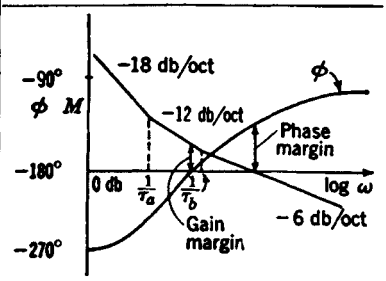
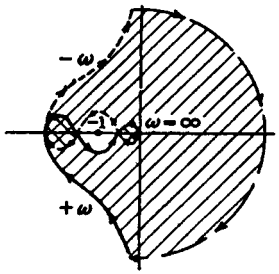
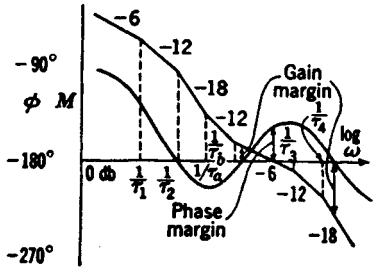
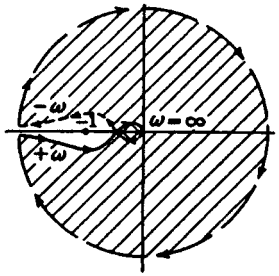
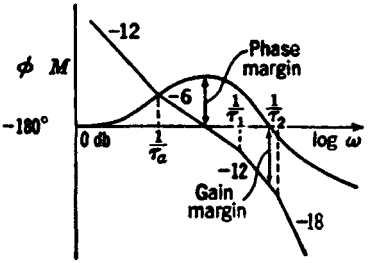
Nichols diagram	Root locus	Comments
		Inherently unstable; must be compensated
		Stable for all gains
		Inherently unstable
		Inherently unstable

Table 27.8 (*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. $\frac{K(s\tau_a+1)}{s^2(s\tau_1+1)(s\tau_2+1)}$		

$$Av_i = \lambda_i v_i$$

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 a eigenvector will remain aligned with that eigenvector.

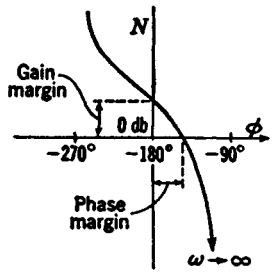
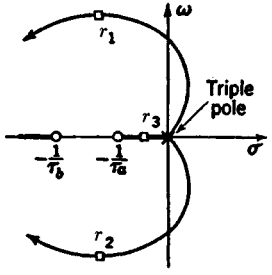
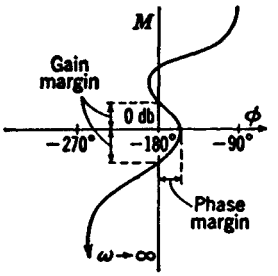
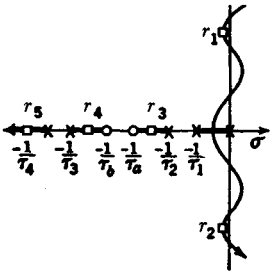
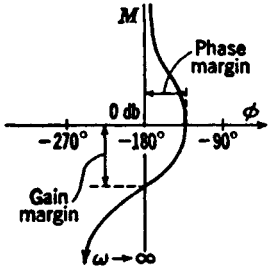
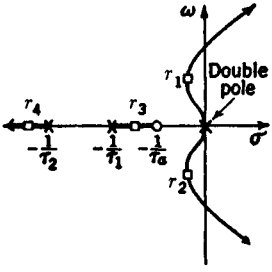
The system eigenvalues are calculated by solving the nth-order polynomial equation

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

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.

Table 27.8 (*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

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.*

27.7 SIMULATION

27.7.1 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.

27.7.2 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. 27.24. 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 kth time step as the initial value of the state for the $(k+1)$st time step, yields a discrete succession of approximations $\tilde{x}(1) \simeq x(t_1)$, $\tilde{x}(2) \simeq x(t_2), \ldots, \tilde{x}(K) \simeq x(t_k)$ spanning the solution time interval.

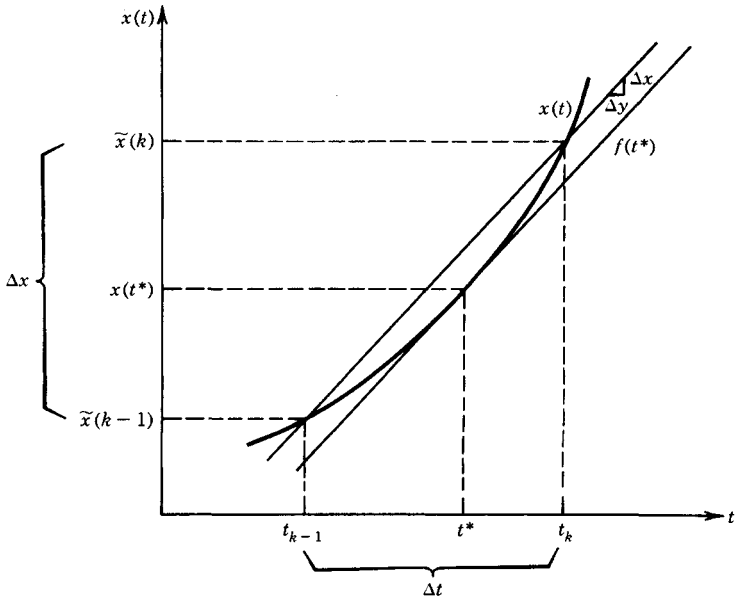


Fig. 27.24 Numerical approximation of a single variable over a single time step.

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 subsections.

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. 27.25 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

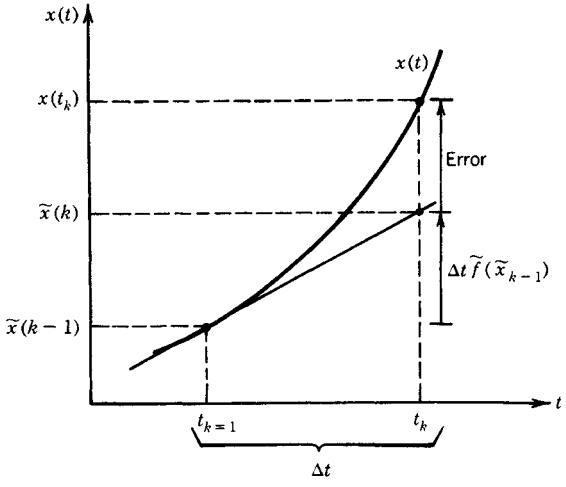


Fig. 27.25 Geometric interpretation of the Euler method for numerical integration.

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

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 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 $t_{k-1} \le t \le t_k$ 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

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

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 27.9.

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 27.9 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 27.9 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 27.10 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

Table 27.10 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

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

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

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

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

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 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

$$\Delta t \leq 2\,|1/\lambda|$$

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. 27.26. 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.

27.8 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 27.11 summarizes these distinguishing criteria. In the following sections, the approaches adopted for the analysis of important classes of systems are briefly outlined.

27.8.1 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.

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

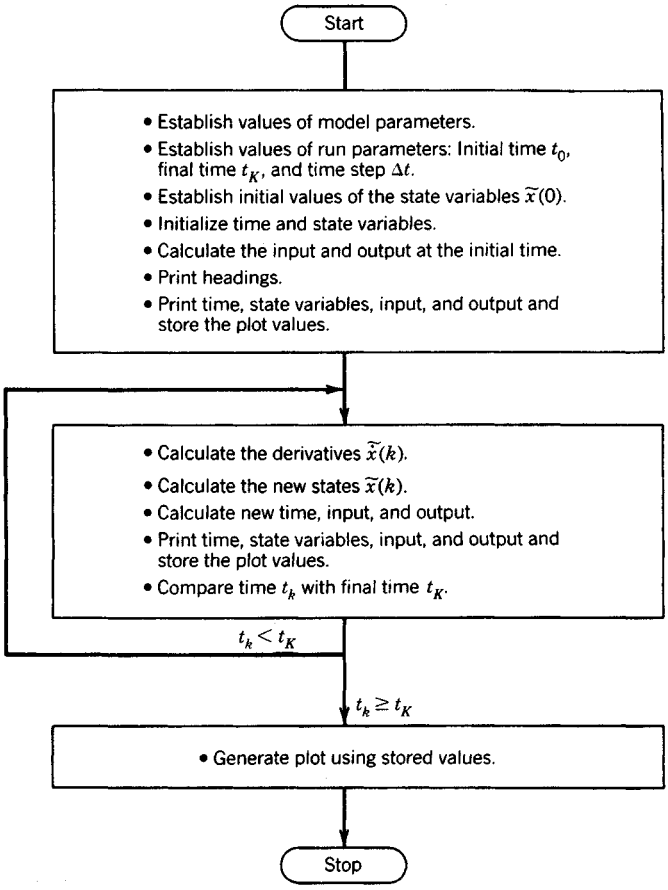


Fig. 27.26 General process for implementing digital simulation (adapted from Close and Frederick[3]).

that a random variable will assume a particular value is measured as 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

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

The mean is the first moment of the distribution. The *n-th 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,

Table 27.11 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).

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

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

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

and the normalized covariance or *correlation coefficient* as

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

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

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

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

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

27.8.2 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. 27.27. 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).

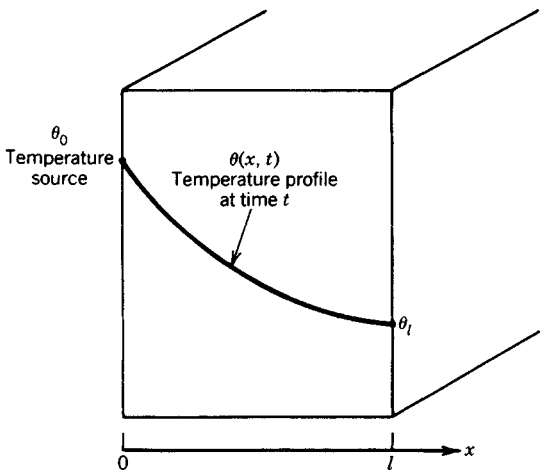


Fig. 27.27 Uniform heat transfer through a wall.

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. 27.28, used to approximate the wall of Fig. 27.27. The corresponding third-order lumped approximation is

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

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

27.8.3 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,

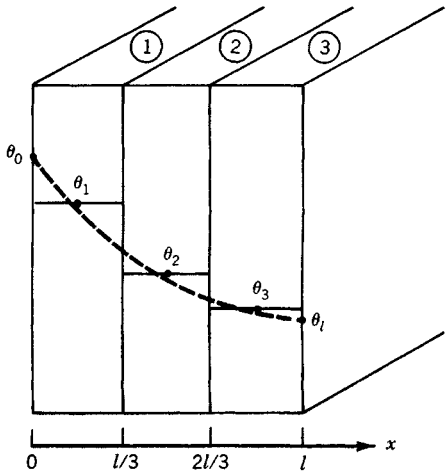


Fig. 27.28 Lumped-parameter model for uniform heat transfer through a wall.

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:

$$\begin{aligned}\Phi(t,t_0) &= I\\ \Phi^{-1}(t,t_0) &= \Phi(t_0,t)\\ \Phi(t_2,t_1)\Phi(t_1,t_0) &= \Phi(t_2,t_0)\\ \Phi(t,t_0) &= A(t)\Phi(t,t_0)\end{aligned}$$

27.8.4 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 27.7, 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 $0 = f(\bar{x},0)$. 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 the 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*

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

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

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. 27.29, such that the equivalent gain and phase associated with this block

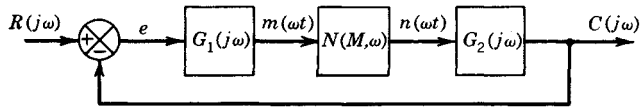


Fig. 27.29 General nonlinear system for describing function analysis.

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.

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. 27.30. 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. 27.30*a*, the relationship between the input $m(\omega t)$ and output $n(\omega t)$ can be written as

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

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

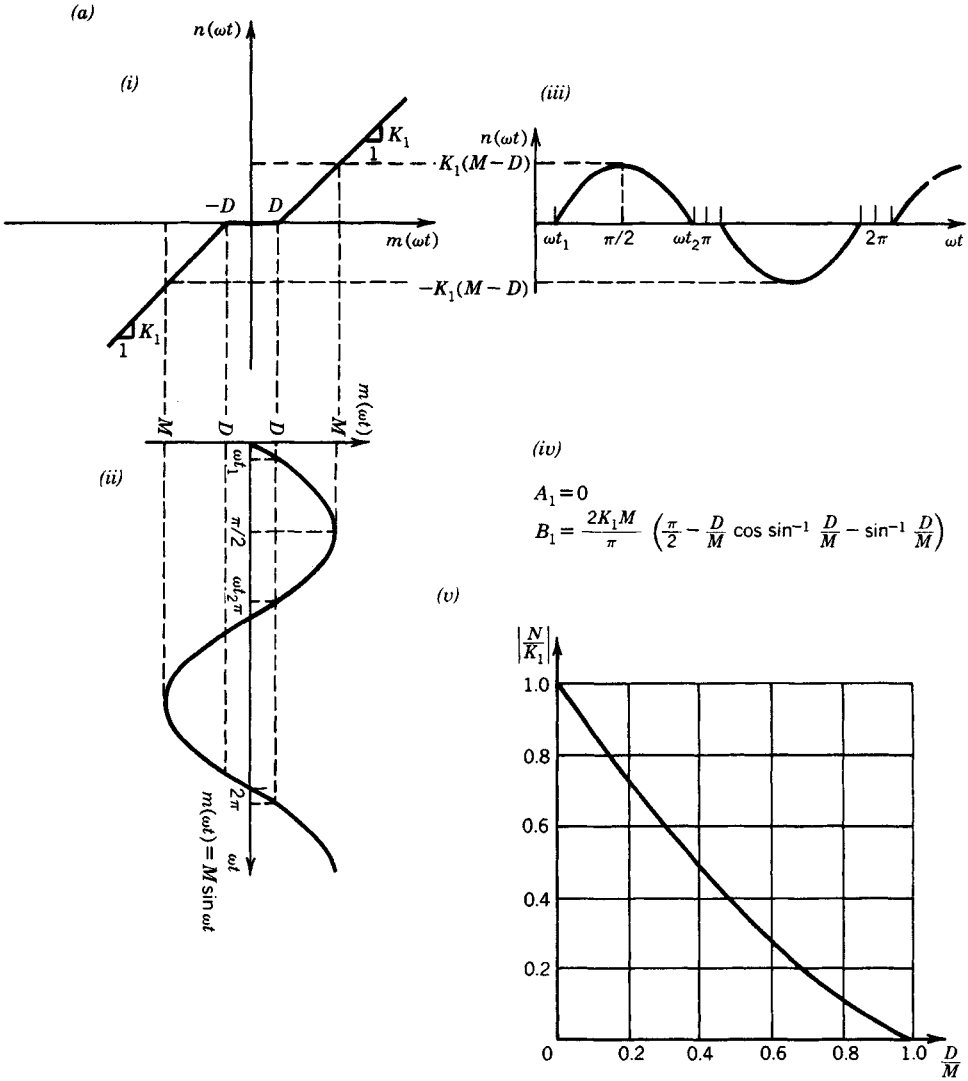


Fig. 27.30a Describing functions for typical nonlinearities (after Refs. 9 and 10). Dead zone nonlinearity: (*i*) nonlinear characteristic; (*ii*) sinusoidal input wave shape; (*iii*) output wave shape; (*iv*) describing-function coefficients; (*v*) normalized describing function.

$$B_1 = 4\left[\frac{2}{\pi/2}\int_0^{\pi/2} n(\omega t)\sin\omega t\, d(\omega t)\right]$$
$$= \frac{4}{\pi}\left[\int_0^{\omega t_1}(0)\sin\omega t\, d(\omega t) + \int_{\omega t_1}^{\pi/2} K_1 M(\sin\omega t - \sin\omega_1 t)\sin\omega t\, d(\omega t)\right]$$

where $\omega t_1 = \sin^{-1}(D/M)$. Evaluating the integrals and dividing by M yields the describing function listed in Fig. 27.30.

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

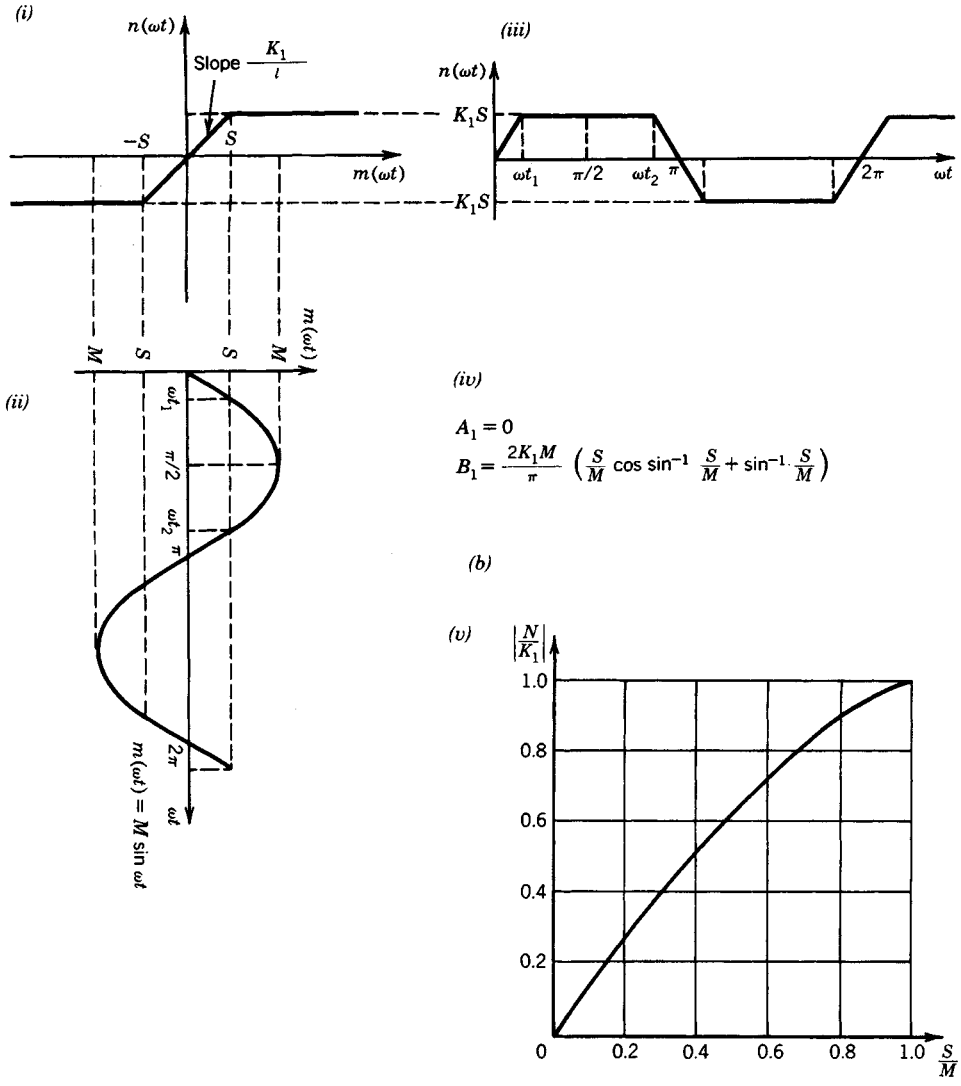


Fig. 27.30*b* Saturation nonlinearity: (*i*) nonlinear characteristic; (*ii*) sinusoidal input wave shape; (*iii*) output wave shape; (*iv*) describing-function coefficients; (*v*) normalized describing function.

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 compact and readily interpreted summary of the system response. Phase portraits for a sample of typical nonlinearities are shown in Fig. 27.31.

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

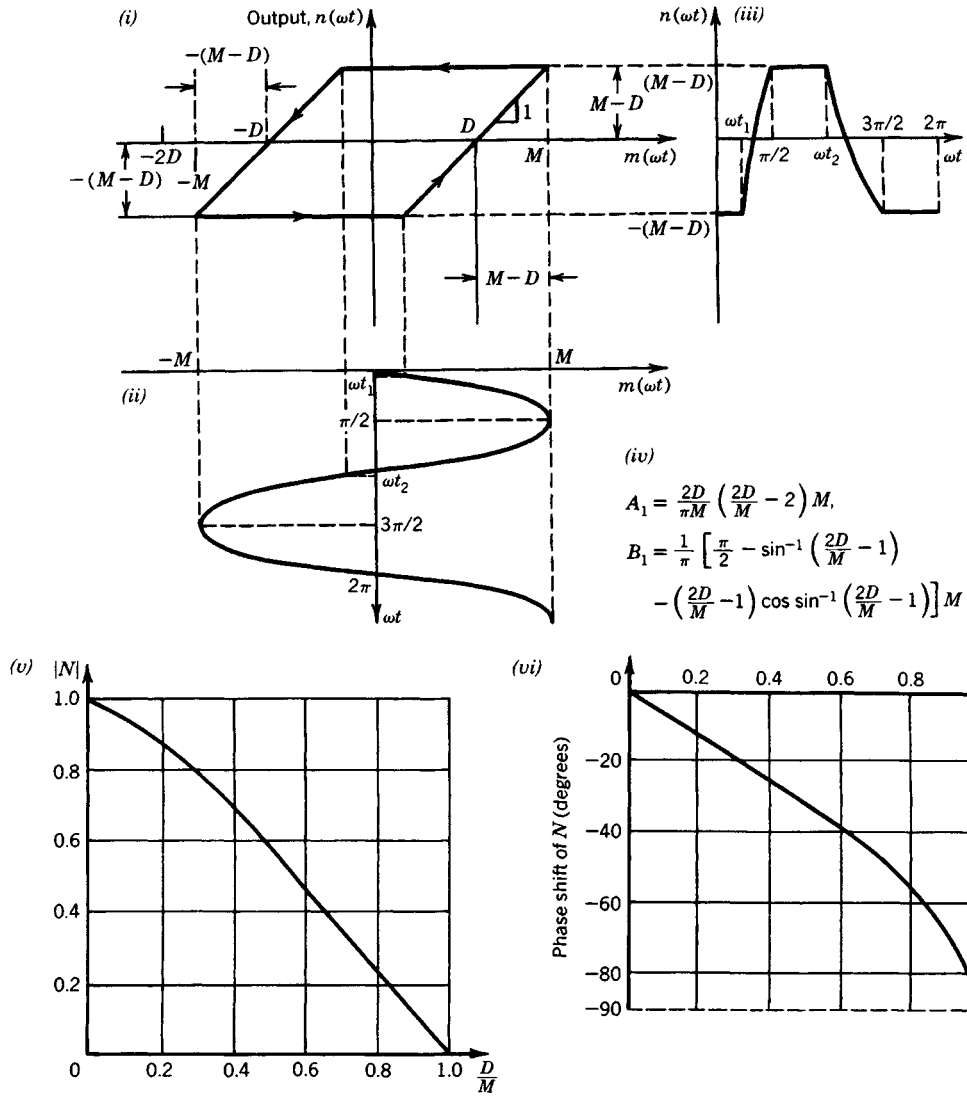


Fig. 27.30c Backlash nonlinearity: (*i*) nonlinear characteristic; (*ii*) sinusoidal input wave shape; (*iii*) output wave shape; (*iv*) describing-function coefficients; (*v*) normalized amplitude characteristics for the describing function; (*vi*) normalized phase characteristics for the describing function.

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

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

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

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

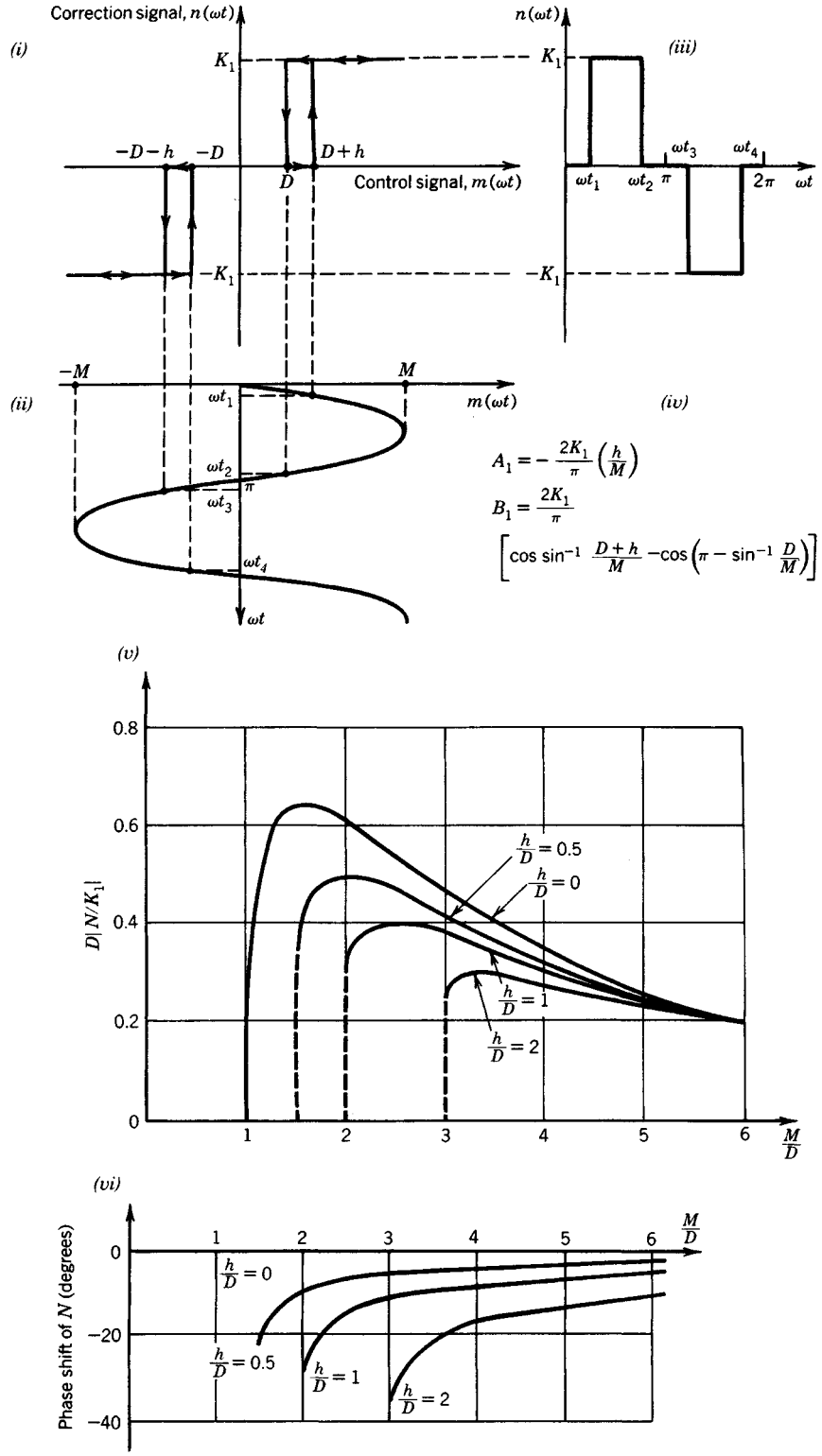


Fig. 27.30*d* Three-position on–off device with hysteresis: (*i*) nonlinear characteristic; (*ii*) sinusoidal input wave shape; (*ii*) output wave shape; (*iv*) describing-function coefficients; (*v*) normalized amplitude characteristics for the describing function; (*vi*) normalized phase characteristics for the describing function.

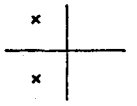
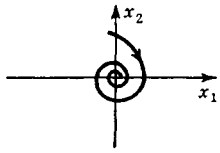
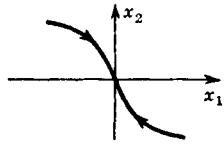
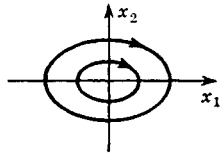
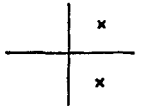
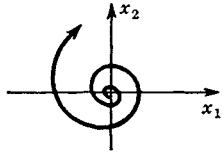
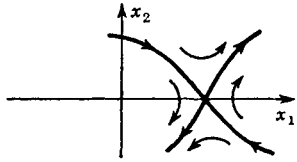
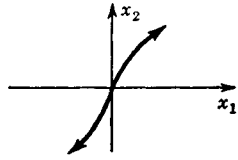
Name	Roots	Sketch
Stable focus or spiral	Damped complex conjugate 	 Trajectories spiral asymptotically to focus
Stable node	Stable real roots 	 Trajectories approach node monotonically
Vortex or center (structurally unstable)	Imaginary roots 	 Conservative system or oscillator
Unstable focus	Complex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

Fig. 27.31 Typical phase-plane plots for second-order systems.[9]

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.

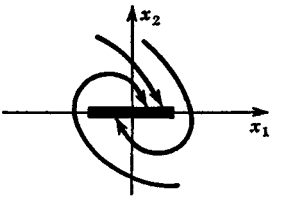
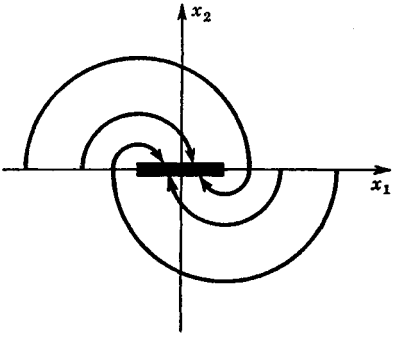
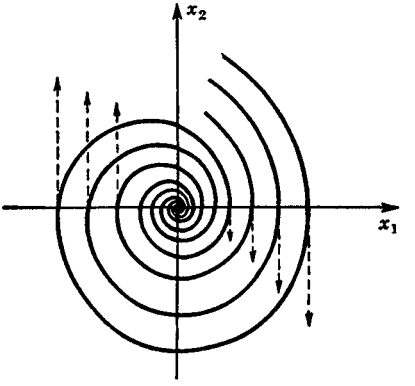
Name	Roots	Sketch
Spiral with nonlinearity viscous damping and coulomb friction		 Spirals are warped by coulomb friction
Nonlinear system with coulomb damping and no viscous damping		 Semicircles center at ends of coulomb line
Spiral with backlash		

Fig. 27.31 *(Continued)*

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.

27.8.5 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 27.7). 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 input–output (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

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

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

$$\begin{aligned} x(k+1) &= f[x(k), u(k)] \\ y(k) &= g[x(k), u(k)] \end{aligned}$$

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

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) \\ y(k) &= C(k)x(k) + D(k)u(k) \end{aligned}$$

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

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

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 *analog-to-digital (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

Table 27.12 *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	$\frac{T^2z(z+1)}{(z-1)^3}$
13		a	$\frac{z}{z-a}$
14		$a^k \cos k\pi$	$\frac{z}{z+a}$

Table 27.13 *z*-Transform Properties

	$x(t)$ or $x(k)$	$\mathcal{Z}[x(t)]$ or $\mathcal{Z}[x(k)]$
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	$x(0)$	$\lim_{z\to\infty} X(z)$ if the limit exists
15	$x(\infty)$	$\lim_{z\to 1}[(z - 1)X(z)]$ if $\dfrac{z - 1}{z}X(z)$ is analytic on and outside the unit circle
16	$\sum_{k=0}^{\infty} x(k)$	$X(1)$
17	$\sum_{k=0}^{n} x(kT)y(nT - kT)$	$X(z)Y(z)$

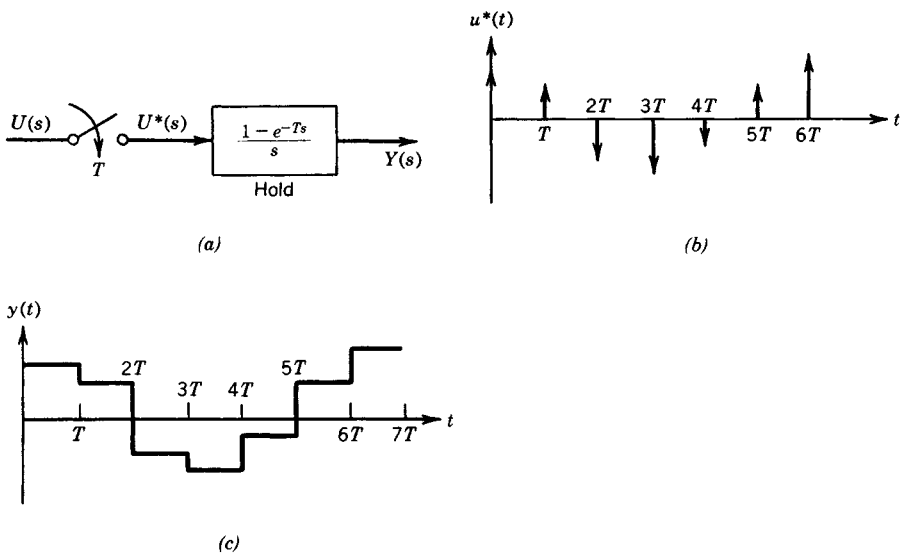


Fig. 27.32 Zero-order hold: (*a*) block diagram of hold with a sampler, (*b*) sampled input sequence, (*c*) analog output for the corresponding input sequence.[4]

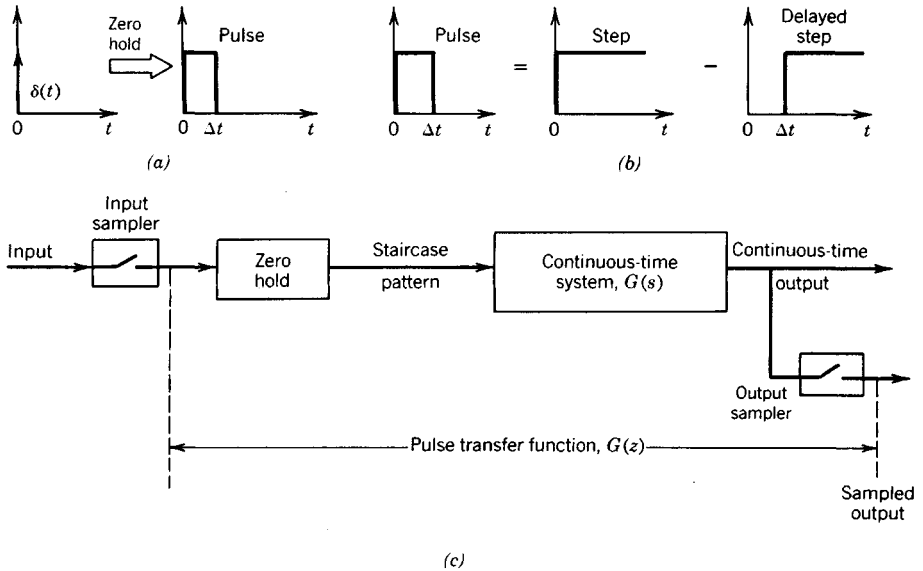


Fig. 27.33 Pulse transfer function of a continuous system with sampler and zero hold.[12]

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

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 leads 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

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

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 27.12. Properties of the z-transform are listed in Table 27.13.

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 *digital-to-analog (D/A) conversion,* that is, to creating a piecewise continuous approximation $u(t)$ of the form

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

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. 27.32. 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. 27.33 can be derived as

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

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.

REFERENCES

1. J. L. Schearer, A. T. Murphy, and H. H. Richardson, *Introduction to System Dynamics,* Addison-Wesley, Reading, MA, 1971.
2. E. O. Doebelin, *System Dynamics: Modeling and Response,* Merrill, Columbus, OH, 1972.
3. C. M. Close and D. K. Frederick, *Modeling and Analysis of Dynamic Systems,* 2nd ed., Houghton Mifflin, Boston, 1993.
4. W. J. Palm III, *Modeling, Analysis, and Control of Dynamic Systems,* Wiley, New York, 1983.
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, *Automatic Control Systems,* 7th ed., Prentice-Hall, Englewood Cliffs, NJ, 1995.
8. W. Thissen, "Investigation into the World3 Model: Lessons for Understanding Complicated Models," *IEEE Transactions on Systems, Man, and Cybernetics,* **SMC-8** (3) (1978).
9. J. E. Gibson, *Nonlinear Automatic Control,* McGraw-Hill, New York, 1963.
10. S. M. Shinners, *Modern Control System Theory and Design,* Wiley, New York, 1992.
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,* Macmillan, New York, 1993.

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,* 4th ed., McGraw-Hill, New York, 1990.

Dorf, R. C., and R. H. Bishop, *Modern Control Systems,* 7th ed., Addison-Wesley, Reading, MA, 1995.

Franklin, G. F., J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems,* 3rd ed., Addison-Wesley, Reading, MA, 1994.

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.

Lay, D. C., *Linear Algebra and Its Applications,* Addison-Wesley, Reading, MA, 1994.

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

Mathworks, The, *The Student Edition of MATLAB 4 User's Guide,* Prentice-Hall, Englewood Cliffs, NJ, 1995.

Mathworks, The, *The Student Edition of SIMULINK User's Guide,* Prentice-Hall, Englewood Cliffs, NJ, 1996.

Pegden, C. D., R. E. Shannon, and R. P. Sadowski, *Introduction to Simulation Using SIMAN,* 2nd ed., McGraw-Hill, New York, 1995.

Phillips, C. L., and R. D. Harbor, *Feedback Control Systems,* 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1996.

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: A System for Doing Mathematics by Computer,* 2nd ed., Addison-Wesley, Reading, MA, 1991.