Appendix WA
A Review of Complex Variables

This appendix is a brief summary of some results on complex variables theory, with emphasis on the facts needed in control theory. For a comprehensive study of basic complex variables theory, see standard textbooks such as Brown and Churchill (1996) or Marsden and Hoffman (1998).

WA.1 Definition of a Complex Number

The complex numbers are distinguished from purely real numbers in that they also contain the **imaginary operator**, which we shall denote j. By definition,

$$j^2 = -1 \quad \text{or} \quad j = \sqrt{-1}. \tag{WA.1}$$

A **complex number** may be defined as

$$A = \sigma + j\omega, \tag{WA.2}$$

where σ is the real part and ω is the imaginary part, denoted, respectively, as

$$\sigma = \mathrm{Re}(A), \quad \omega = \mathrm{Im}(A). \tag{WA.3}$$

Note that the imaginary part of A is itself a real number.

Graphically, we may represent the complex number A in two ways. In the Cartesian coordinate system (Fig. WA.1a), A is represented by a single point in the complex plane. In the polar coordinate system, A is represented by a vector with length r and an angle θ; the angle is measured in radians counterclockwise from the positive real axis (Fig. WA.1b). In polar form the complex number A is denoted by

$$A = |A| \cdot \angle \arg A = r \cdot \angle\theta = re^{j\theta}, \quad 0 \le \theta \le 2\pi, \tag{WA.4}$$

where r—called the **magnitude**, **modulus**, or **absolute value** of A—is the length of the vector representing A, namely,

$$r = |A| = \sqrt{\sigma^2 + \omega^2}, \tag{WA.5}$$

and where θ is given by

$$\tan\theta = \frac{\omega}{\sigma} \tag{WA.6}$$

or

$$\theta = \arg(A) = \tan^{-1}\left(\frac{\omega}{\sigma}\right). \tag{WA.7}$$

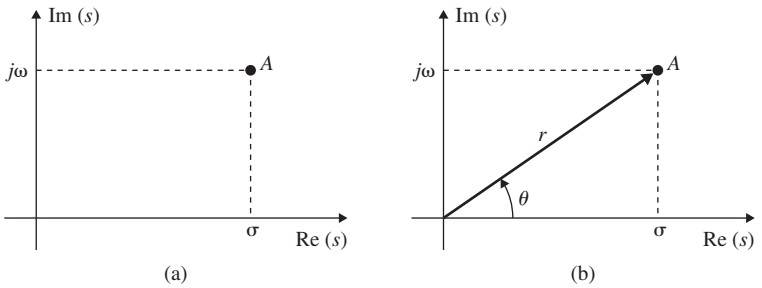


Figure WA.1
The complex number A represented in (a) Cartesian and (b) polar coordinates

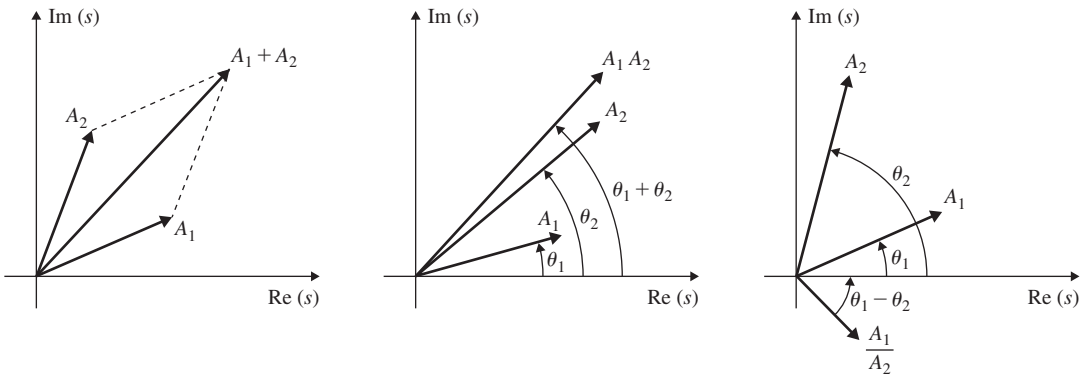


Figure WA.2
Arithmetic of complex numbers: (a) addition; (b) multiplication; (c) division

Care must be taken to compute the correct value of the angle, depending on the sign of the real and imaginary parts (that is, one must find the quadrant in which the complex number lies).

The **conjugate** of A is defined as

$$A^* = \sigma - j\omega. \tag{WA.8}$$

Therefore,

$$(A^*)^* = A, \tag{WA.9}$$

$$(A_1 \pm A_2)^* = A_1^* \pm A_2^*, \tag{WA.10}$$

$$\left(\frac{A_1}{A_2}\right)^* = \frac{A_1^*}{A_2^*}, \tag{WA.11}$$

$$(A_1 A_2)^* = A_1^* A_2^*, \tag{WA.12}$$

$$\mathrm{Re}(A) = \frac{A + A^*}{2}, \quad \mathrm{Im}(A) = \frac{A - A^*}{2j}, \tag{WA.13}$$

$$AA^* = (|A|)^2. \tag{WA.14}$$

WA.2 Algebraic Manipulations

WA.2.1 Complex Addition

If we let

$$A_1 = \sigma_1 + j\omega_1 \quad \text{and} \quad A_2 = \sigma_2 + j\omega_2, \tag{WA.15}$$

then

$$A_1 + A_2 = (\sigma_1 + j\omega_1) + (\sigma_2 + j\omega_2) = (\sigma_1 + \sigma_2) + j(\omega_1 + \omega_2). \tag{WA.16}$$

Because each complex number is represented by a vector extending from the origin, we can add or subtract complex numbers graphically. The sum is obtained by adding the two vectors. This we do by constructing a parallelogram and finding its diagonal, as shown in Fig. WA.2a. Alternatively, we could start at the tail of one vector, draw a vector parallel to the other vector, and then connect the origin to the new arrowhead.

Complex subtraction is very similar to complex addition.

WA.2.2 Complex Multiplication

For two complex numbers defined according to Eq. (WA.15),

$$\begin{aligned} A_1A_2 &= (\sigma_1 + j\omega_1)(\sigma_2 + j\omega_2) \\ &= (\sigma_1\sigma_2 - \omega_1\omega_2) + j(\omega_1\sigma_2 + \sigma_1\omega_2). \end{aligned} \tag{WA.17}$$

The product of two complex numbers may be obtained graphically using polar representations, as shown in Fig. WA.2b.

WA.2.3 Complex Division

The division of two complex numbers is carried out by **rationalization**. This means that both the numerator and denominator in the ratio are multiplied by the conjugate of the denominator:

$$\begin{aligned} \frac{A_1}{A_2} &= \frac{A_1A_2^*}{A_2A_2^*} \\ &= \frac{(\sigma_1\sigma_2 + \omega_1\omega_2) + j(\omega_1\sigma_2 - \sigma_1\omega_2)}{\sigma_2^2 + \omega_2^2}. \end{aligned} \tag{WA.18}$$

From Eq. (WA.4) it follows that

$$A^{-1} = \frac{1}{r}e^{-j\theta}, \quad r \neq 0. \tag{WA.19}$$

Also, if $A_1 = r_1e^{j\theta_1}$ and $A_2 = r_2e^{j\theta_2}$, then

$$A_1A_2 = r_1r_2e^{j(\theta_1+\theta_2)}, \tag{WA.20}$$

where $|A_1A_2| = r_1r_2$ and $\arg(A_1A_2) = \theta_1 + \theta_2$, and

$$\frac{A_1}{A_2} = \frac{r_1}{r_2}e^{j(\theta_1-\theta_2)}, \quad r_2 \neq 0, \tag{WA.21}$$

where $\left|\frac{A_1}{A_2}\right| = \frac{r_1}{r_2}$ and $\arg\left(\frac{A_1}{A_2}\right) = \theta_1 - \theta_2$. The division of complex numbers may be carried out graphically in polar coordinates as shown in Fig. WA.2c.

EXAMPLE WA.1 *Frequency Response of First-Order System*

Find the magnitude and phase of the transfer function $G(s) = \frac{1}{s+1}$, where $s = j\omega$.

Solution. Substituting $s = j\omega$ and rationalizing, we obtain

$$G(j\omega) = \frac{1}{\sigma + 1 + j\omega}\frac{\sigma + 1 - j\omega}{\sigma + 1 - j\omega} = \frac{\sigma + 1 - j\omega}{(\sigma + 1)^2 + \omega^2}.$$

Therefore, the magnitude and phase are

$$|G(j\omega)| = \frac{\sqrt{(\sigma + 1)^2 + \omega^2}}{(\sigma + 1)^2 + \omega^2} = \frac{1}{\sqrt{(\sigma + 1)^2 + \omega^2}},$$

$$\arg(G(j\omega)) = \tan^{-1}\left(\frac{\mathrm{Im}(G(j\omega))}{\mathrm{Re}(G(j\omega))}\right) = \tan^{-1}\left(\frac{-\omega}{\sigma + 1}\right).$$

WA.3 Graphical Evaluation of Magnitude and Phase

Consider the transfer function

$$G(s) = \frac{\prod_{i=1}^{m}(s + z_i)}{\prod_{i=1}^{n}(s + p_i)}. \tag{WA.22}$$

The value of the transfer function for sinusoidal inputs is found by replacing s with $j\omega$. The gain and phase are given by $G(j\omega)$ and may be determined analytically or by a graphical procedure. Consider the pole-zero configuration for such a $G(s)$ and a point $s_0 = j\omega_0$ on the imaginary axis, as shown in Fig. WA.3. Also consider the vectors drawn from the poles and the zero to s_0. The magnitude of the transfer function evaluated at $s_0 = j\omega_0$ is simply the ratio of the distance from the zero to the product of all the distances from the poles:

$$|G(j\omega_0)| = \frac{r_1}{r_2 r_3 r_4}. \tag{WA.23}$$

The phase is given by the sum of the angles from the zero minus the sum of the angles from the poles:

$$\arg G(j\omega_0) = \angle G(j\omega_0) = \theta_1 - (\theta_2 + \theta_3 + \theta_4). \tag{WA.24}$$

This may be explained as follows. The term $s + z_1$ is a vector addition of its two components. We may determine this equivalently as $s - (-z_1)$, which amounts to translation of the vector $s + z_1$ starting at $-z_1$, as shown in

Figure WA.3
Graphical determination of magnitude and phase

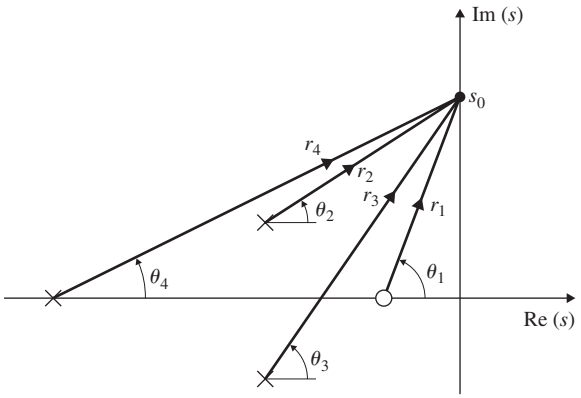


Figure WA.4
Illustration of graphical computation of $s + z_1$

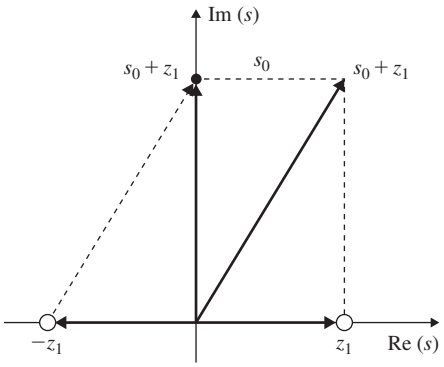


Fig. WA.4. This means that a vector drawn from the zero location to s_0 is equivalent to $s+z_1$. The same reasoning applies to the poles. We reflect p_1, p_2, and p_3 about the origin to obtain the pole locations. Then the vectors drawn from $-p_1$, $-p_2$, and $-p_3$ to s_0 are the same as the vectors in the denominator represented in polar coordinates. Note that this method may also be used to evaluate s_0 at places in the complex plane besides the imaginary axis.

WA.4 Differentiation and Integration

The usual rules apply to complex differentiation. Let $G(s)$ be differentiable with respect to s. Then the derivative at s_0 is defined as

$$G'(s_0) = \lim_{s \to s_0} \frac{G(s) - G(s_0)}{s - s_0}, \tag{WA.25}$$

provided that the limit exists. For conditions on the existence of the derivative, see Brown and Churchill (1996).

The standard rules also apply to integration, except that the constant of integration c is a complex constant:

$$\int G(s)ds = \int \mathrm{Re}[G(s)]ds + j \int \mathrm{Im}[G(s)]ds + c. \tag{WA.26}$$

WA.5 Euler's Relations

Let us now derive an important relationship involving the complex exponential. If we define

$$A = \cos\theta + j\sin\theta, \tag{WA.27}$$

where θ is in radians, then

$$\begin{aligned}\frac{dA}{d\theta} &= -\sin\theta + j\cos\theta = j^2\sin\theta + j\cos\theta \\ &= j(\cos\theta + j\sin\theta) = jA.\end{aligned} \tag{WA.28}$$

We collect the terms involving A to obtain

$$\frac{dA}{A} = jd\theta. \tag{WA.29}$$

Integrating both sides of Eq. (WA.29) yields

$$\ln A = j\theta + c, \tag{WA.30}$$

where c is a constant of integration. If we let $\theta = 0$ in Eq. (WA.30), we find that $c = 0$ or

$$A = e^{j\theta} = \cos\theta + j\sin\theta. \tag{WA.31}$$

Similarly,

$$A^* = e^{-j\theta} = \cos\theta - j\sin\theta. \tag{WA.32}$$

Euler's relations

From Eqs. (WA.31) and (WA.32) it follows that

$$\cos\theta = \frac{e^{j\theta} + e^{-j\theta}}{2}, \tag{WA.33}$$

$$\sin\theta = \frac{e^{j\theta} - e^{-j\theta}}{2j}. \tag{WA.34}$$

WA.6 Analytic Functions

Let us assume that G is a complex-valued function defined in the complex plane. Let s_0 be in the domain of G, which is assumed to be finite within some disk centered at s_0. Thus, $G(s)$ is defined not only at s_0 but also at all points in the disk centered at s_0. The function G is said to be **analytic** if its derivative exists at s_0 and at each point in the neighborhood of s_0.

WA.7 Cauchy's Theorem

A **contour** is a piecewise-smooth arc that consists of a number of smooth arcs joined together. A **simple closed contour** is a contour that does not intersect itself and ends on itself. Let C be a closed contour as shown in Fig. WA.5a, and let G be analytic inside and on C. Cauchy's theorem states that

$$\oint_C G(s)ds = 0. \tag{WA.35}$$

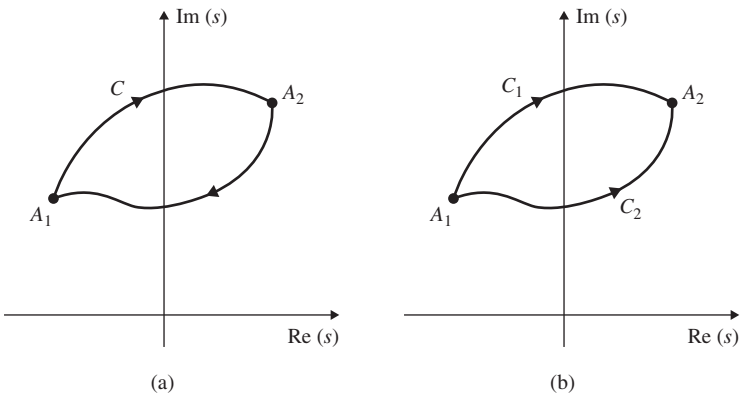


Figure WA.5
Contours in the s-plane: (a) a closed contour; (b) two different paths between A_1 and A_2

There is a corollary to this theorem: Let C_1 and C_2 be two paths connecting the points A_1 and A_2 as in Fig. WA.5b. Then,

$$\int_{C_1} G(s)ds = \int_{C_2} G(s)ds. \tag{WA.36}$$

WA.8 Singularities and Residues

If a function $G(s)$ is not analytic at s_0 but is analytic at some point in every neighborhood of s_0, it is said to be a **singularity**. A singular point is said to be an **isolated singularity** if $G(s)$ is analytic everywhere else in the neighborhood of s_0 except at s_0. Let $G(s)$ be a **rational function** (that is, a ratio of polynomials). If the numerator and denominator are both analytic, then $G(s)$ will be analytic except at the locations of the poles (that is, at roots of the denominator). All singularities of rational algebraic functions are the pole locations.

Let $G(s)$ be analytic except at s_0. Then we may write $G(s)$ in its Laurent series expansion form:

$$G(s) = \frac{A_{-n}}{(s-s_0)^n} + \ldots + \frac{A_{-1}}{(s-s_0)} + B_0 + B_1(s-s_0) + \ldots. \tag{WA.37}$$

The coefficient A_{-1} is called the **residue** of $G(s)$ at s_0 and may be evaluated as

$$A_{-1} = \text{Res}[G(s); s_0] = \frac{1}{2\pi j}\oint_C G(s)\, ds, \tag{WA.38}$$

where C denotes a closed arc within an analytic region centered at s_0 that contains no other singularity, as shown in Fig. WA.6. When s_0 is not repeated with $n = 1$, we have

$$A_{-1} = \text{Res}[G(s); s_0] = (s-s_0)G(s)|_{s=s_0}. \tag{WA.39}$$

This is the familiar cover-up method of computing residues.

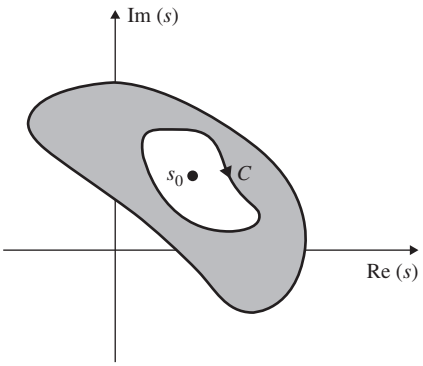


Figure WA.6
Contour around an isolated singularity

WA.9 Residue Theorem

If the contour C contains l singularities, then Eq. (WA.39) may be generalized to yield **Cauchy's residue theorem**:

$$\frac{1}{2\pi j}\oint G(s)\,ds = \sum_{i=1}^{l} \text{Res}[G(s); s_i]. \tag{WA.40}$$

WA.10 The Argument Principle

Before stating the argument principle, we need a preliminary result from which the principle follows readily.

Number of Poles and Zeros

Let $G(s)$ be an analytic function inside and on a closed contour C, except for a finite number of poles inside C. Then, for C described in the positive sense (clockwise direction),

$$\frac{1}{2\pi j}\oint \frac{G'(s)}{G(s)}\,ds = N - P \tag{WA.41}$$

or

$$\frac{1}{2\pi j}\oint d(\ln G) = N - P, \tag{WA.42}$$

where N and P are the total number of zeros and poles of G inside C, respectively. A pole or zero of multiplicity k is counted k times.

Proof Let s_0 be a zero of G with multiplicity k. Then, in some neighborhood of that point, we may write $G(s)$ as

$$G(s) = (s - s_0)^k f(s), \tag{WA.43}$$

where $f(s)$ is analytic and $f(s_0) \neq 0$. If we differentiate Eq. (WA.43), we obtain

$$G'(s) = k(s - s_0)^{k-1} f(s) + (s - s_0)^k f'(s). \tag{WA.44}$$

Equation (WA.44) may be rewritten as

$$\frac{G'(s)}{G(s)} = \frac{k}{s - s_0} + \frac{f'(s)}{f(s)}. \tag{WA.45}$$

Therefore, $G'(s)/G(s)$ has a pole at $s = s_0$ with residue K. This analysis may be repeated for every zero. Hence, the sum of the residues of $G'(s)/G(s)$ is the number of zeros of $G(s)$ inside C. If s_0 is a pole with multiplicity l, we may write

$$h(s) = (s - s_0)^l G(s), \tag{WA.46}$$

where $h(s)$ is analytic and $h(s_0) \neq 0$. Then Eq. (WA.46) may be rewritten as

$$G(s) = \frac{h(s)}{(s - s_0)^l}. \tag{WA.47}$$

Differentiating Eq. (WA.47) we obtain

$$G'(s) = \frac{h'(s)}{(s - s_0)^l} - \frac{lh(s)}{(s - s_0)^{l+1}}, \tag{WA.48}$$

so that

$$\frac{G'(s)}{G(s)} = \frac{-l}{s - s_0} + \frac{h'(s)}{h(s)}. \tag{WA.49}$$

This analysis may be repeated for every pole. The result is that the sum of the residues of $G'(s)/G(s)$ at all the poles of $G(s)$ is $-P$.

The Argument Principle

Using Eq. (WA.38), we get

$$\frac{1}{2\pi j} \oint_C d[\ln G(s)] = N - P, \tag{WA.50}$$

where $d[\ln G(s)]$ was substituted for $G'(s)/G(s)$. If we write $G(s)$ in polar form, then

$$\begin{aligned} \oint_\Gamma d[\ln G(s)] &= \oint_\Gamma d\{\ln |G(s)| + j \arg[\ln G(s)]\} \\ &= \ln |G(s)||_{s=s_1}^{s=s_2} + j \arg G(s)|_{s=s_1}^{s=s_2}. \end{aligned} \tag{WA.51}$$

Because Γ is a closed contour, the first term is zero, but the second term is 2π times the net encirclements of the origin:

$$\frac{1}{2\pi j} \oint_\Gamma d[\ln G(s)] = N - P. \tag{WA.52}$$

Intuitively, the argument principle may be stated as follows: We let $G(s)$ be a rational function that is analytic except at possibly a finite number of points. We select an arbitrary contour in the s-plane so that $G(s)$ is analytic at every point on the contour (the contour does not pass through any of the singularities). The corresponding mapping into the $G(s)$-plane may encircle

the origin. The number of times it does so is determined by the difference between the number of zeros and the number of poles of $G(s)$ encircled by the s-plane contour. The direction of this encirclement is determined by which is greater, N (clockwise) or P (counterclockwise). For example, if the contour encircles a single zero, the mapping will encircle the origin once in the clockwise direction. Similarly, if the contour encloses only a single pole, the mapping will encircle the origin, this time in the counterclockwise direction. If the contour encircles no singularities, or if the contour encloses an equal number of poles and zeros, there will be no encirclement of the origin. A contour evaluation of $G(s)$ will encircle the origin if there is a nonzero net difference between the encircled singularities. The mapping is **conformal** as well, which means that the magnitude and sense of the angles between smooth arcs is preserved. Chapter 6 provides a more detailed intuitive treatment of the argument principle and its application to feedback control in the form of the Nyquist stability theorem.

WA.11 Bilinear Transformation

A bilinear transformation is of the form

$$w = \frac{as + b}{cs + d}, \tag{WA.53}$$

where a, b, c, d are complex constants, and it is assumed that $ad - bc \neq 0$. The bilinear transformation always transforms circles in the w-plane into circles in the s-plane. This can be shown in several ways. If we solve for s, we obtain

$$s = \frac{-dw + b}{cw - a}. \tag{WA.54}$$

The equation for a circle in the w-plane is of the form

$$\frac{|w - \sigma|}{|w - \rho|} = R. \tag{WA.55}$$

If we substitute for w in terms of s in Eq. (WA.53), we get

$$\frac{|s - \sigma'|}{|s - \rho'|} = R', \tag{WA.56}$$

where

$$\sigma' = \frac{\sigma d - b}{a - \sigma c}, \quad \rho' = \frac{\rho d - b}{a - \rho c}, \quad R' = \left|\frac{a - \rho c}{a - \sigma c}\right| R, \tag{WA.57}$$

which is the equation for a circle in the s-plane. For alternative proofs the reader is referred to Brown and Churchill (1996) and Marsden and Hoffman (1998).

Appendix WB
Summary of Matrix Theory

In the text, we assume you are already somewhat familiar with matrix theory and with the solution of linear systems of equations. However, for the purposes of review we present here a brief summary of matrix theory with an emphasis on the results needed in control theory. For further study, see Strang (2006) and Gantmacher (1959).

WB.1 Matrix Definitions

An array of numbers arranged in rows and columns is referred to as a **matrix**. If $\mathbf{A}$ is a matrix with m rows and n columns, an $m \times n$ (read "m by n") matrix, it is denoted by

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \tag{WB.1}$$

where the entries a_{ij} are its elements. If $m = n$, then the matrix is **square**; otherwise it is **rectangular**. Sometimes a matrix is simply denoted by $\mathbf{A} = [a_{ij}]$. If $m = 1$ or $n = 1$, then the matrix reduces to a **row vector** or a **column vector**, respectively. A **submatrix** of $\mathbf{A}$ is the matrix with certain rows and columns removed.

WB.2 Elementary Operations on Matrices

If $\mathbf{A}$ and $\mathbf{B}$ are matrices of the same dimension, then their sum is defined by

$$\mathbf{C} = \mathbf{A} + \mathbf{B}, \tag{WB.2}$$

where

$$c_{ij} = a_{ij} + b_{ij}. \tag{WB.3}$$

That is, the addition is done element by element. It is easy to verify the following properties of matrices:

Commutative law for addition

$$\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}, \tag{WB.4}$$

Associative law for addition

$$(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C}). \tag{WB.5}$$

Two matrices can be multiplied if they are compatible. Let $\mathbf{A} = m \times n$ and $\mathbf{B} = n \times p$. Then the $m \times p$ matrix

$$\mathbf{C} = \mathbf{AB} \tag{WB.6}$$

is the product of the two matrices, where

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}. \tag{WB.7}$$

Associative law for multiplication

Matrix multiplication satisfies the associative law

$$\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}, \tag{WB.8}$$

but not the commutative law; that is, in general,

$$\mathbf{AB} \neq \mathbf{BA}. \tag{WB.9}$$

WB.3 Trace

The **trace** of a square matrix is the sum of its diagonal elements:

$$\text{trace } \mathbf{A} = \sum_{i=1}^{n} a_{ii}. \tag{WB.10}$$

WB.4 Transpose

The $n \times m$ matrix obtained by interchanging the rows and columns of $\mathbf{A}$ is called the **transpose of matrix A**:

$$\mathbf{A}^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix}.$$

A matrix is said to be **symmetric** if

$$\mathbf{A}^T = \mathbf{A}. \tag{WB.11}$$

Transposition

It is easy to show that

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T, \tag{WB.12}$$

$$(\mathbf{ABC})^T = \mathbf{C}^T\mathbf{B}^T\mathbf{A}^T, \tag{WB.13}$$

$$(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T. \tag{WB.14}$$

WB.5 Determinant and Matrix Inverse

The **determinant** of a square matrix is defined by Laplace's expansion

$$\det \mathbf{A} = \sum_{j=1}^{n} a_{ij}\gamma_{ij} \quad \text{for any} \quad i = 1, 2, \dots, n, \tag{WB.15}$$

where γ_{ij} is called the **cofactor** and

$$\gamma_{ij} = (-1)^{i+j} \det M_{ij}, \tag{WB.16}$$

where the scalar $\det M_{ij}$ is called a **minor**. M_{ij} is the same as the matrix $\mathbf{A}$ except that its ith row and jth column have been removed. Note that M_{ij} is always an $(n-1) \times (n-1)$ matrix, and that the minors and cofactors are identical except possibly for a sign.

The **adjugate** of a matrix is the transpose of the matrix of its cofactors:

$$\text{adj } \mathbf{A} = [\gamma_{ij}]^T. \tag{WB.17}$$

It can be shown that

$$\mathbf{A} \text{ adj } \mathbf{A} = (\det \mathbf{A})\mathbf{I}, \tag{WB.18}$$

Identity matrix

where $\mathbf{I}$ is called the **identity matrix**:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix};$$

that is, $\mathbf{I}$ has ones along the diagonal and zeros elsewhere. If $\det \mathbf{A} \neq 0$, then the **inverse** of a matrix $\mathbf{A}$ is defined by

$$\mathbf{A}^{-1} = \frac{\text{adj } \mathbf{A}}{\det \mathbf{A}}, \tag{WB.19}$$

and has the property

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}. \tag{WB.20}$$

Note that a matrix has an inverse—that is, it is **nonsingular**—if its determinant is nonzero.

The inverse of the product of two matrices is the product of the inverse of the matrices in reverse order:

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \tag{WB.21}$$

Inversion

and

$$(\mathbf{ABC})^{-1} = \mathbf{C}^{-1}\mathbf{B}^{-1}\mathbf{A}^{-1}. \tag{WB.22}$$

WB.6 Properties of the Determinant

When dealing with determinants of matrices, the following elementary (row or column) operations are useful:

1. If any row (or column) of $\mathbf{A}$ is multiplied by a scalar α, the resulting matrix $\bar{\mathbf{A}}$ has the determinant

$$\det \bar{\mathbf{A}} = \alpha \det \mathbf{A}. \tag{WB.23}$$

Hence

$$\det(\alpha \mathbf{A}) = \alpha^n \det \mathbf{A}. \tag{WB.24}$$

2. If any two rows (or columns) of $\mathbf{A}$ are interchanged to obtain $\bar{\mathbf{A}}$, then

$$\det \bar{\mathbf{A}} = -\det \mathbf{A}. \tag{WB.25}$$

3. If a multiple of a row (or column) of $\mathbf{A}$ is added to another to obtain $\bar{\mathbf{A}}$, then

$$\det \bar{\mathbf{A}} = \det \mathbf{A}. \tag{WB.26}$$

4. It is also easy to show that

$$\det \mathbf{A} = \det \mathbf{A}^T \tag{WB.27}$$

and

$$\det \mathbf{AB} = \det \mathbf{A} \det \mathbf{B}. \tag{WB.28}$$

Applying Eq. (WB.28) to Eq. (WB.20), we have

$$\det \mathbf{A} \det \mathbf{A}^{-1} = 1. \tag{WB.29}$$

If $\mathbf{A}$ and $\mathbf{B}$ are square matrices, then the determinant of the block triangular matrix is the product of the determinants of the diagonal blocks:

$$\det \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ 0 & \mathbf{B} \end{bmatrix} = \det \mathbf{A} \det \mathbf{B}. \tag{WB.30}$$

If $\mathbf{A}$ is nonsingular, then

$$\det \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \det \mathbf{A} \det(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}). \tag{WB.31}$$

Using this identity, we can write the transfer function of a scalar system in a compact form:

$$G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + J = \frac{\det \begin{bmatrix} (s\mathbf{I} - \mathbf{A}) & \mathbf{B} \\ -\mathbf{C} & D \end{bmatrix}}{\det(s\mathbf{I} - \mathbf{A})}. \tag{WB.32}$$

WB.7 Inverse of Block Triangular Matrices

If $\mathbf{A}$ and $\mathbf{B}$ are square invertible matrices, then

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{C}\mathbf{B}^{-1} \\ \mathbf{0} & \mathbf{B}^{-1} \end{bmatrix}. \tag{WB.33}$$

WB.8 Special Matrices

Diagonal matrix

Some matrices have special structures and are given names. We have already defined the identity matrix, which has a special form. A **diagonal matrix** has (possibly) nonzero elements along the main diagonal and zeros elsewhere:

$$\mathbf{A} = \begin{bmatrix} a_{11} & & \mathbf{0} & & \\ & a_{22} & & & \\ & & a_{33} & & \\ & & & \ddots & \\ \mathbf{0} & & & & a_{nn} \end{bmatrix}. \tag{WB.34}$$

Upper triangular matrix

A matrix is said to be **(upper) triangular** if all the elements below the main diagonal are zeros:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & & \cdots & a_{1n} \\ 0 & a_{22} & & & \\ \vdots & 0 & & & \vdots \\ 0 & \vdots & \ddots & \ddots & \\ 0 & 0 & \cdots & 0 & a_{nn} \end{bmatrix}. \tag{WB.35}$$

The determinant of a diagonal or triangular matrix is simply the product of its diagonal elements.

A matrix is said to be in the **(upper) companion form** if it has the structure

$$\mathbf{A}_c = \begin{bmatrix} -a_1 & -a_2 & & \cdots & -a_n \\ 1 & 0 & & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & \cdots & 1 & 0 \end{bmatrix}. \tag{WB.36}$$

Note that all the information is contained in the first row. Variants of this form are the lower, left, or right companion matrices. A **Vandermonde matrix** has the following structure:

$$\mathbf{A} = \begin{bmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^{n-1} \\ 1 & a_2 & a_2^2 & \cdots & a_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & a_n & a_n^2 & \cdots & a_n^{n-1} \end{bmatrix}. \tag{WB.37}$$

WB.9 Rank

The **rank** of a matrix is the number of its linearly independent rows or columns. If the rank of $\mathbf{A}$ is r, then all $(r+1) \times (r+1)$ submatrices of $\mathbf{A}$ are singular, and there is at least one $r \times r$ submatrix that is nonsingular. It is also true that

$$\text{row rank of } \mathbf{A} = \text{column rank of } \mathbf{A}. \tag{WB.38}$$

WB.10 Characteristic Polynomial

The **characteristic polynomial** of a matrix $\mathbf{A}$ is defined by

$$\begin{aligned} a(s) &\triangleq \det(s\mathbf{I} - \mathbf{A}) \\ &= s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n, \end{aligned} \tag{WB.39}$$

where the roots of the polynomial are referred to as **eigenvalues** of $\mathbf{A}$. We can write

$$a(s) = (s - \lambda_1)(s - \lambda_2) \cdots (s - \lambda_n), \tag{WB.40}$$

where $\{\lambda_i\}$ are the eigenvalues of $\mathbf{A}$. The characteristic polynomial of a companion matrix (for example, Eq. (WB.36)) is

$$\begin{aligned} a(s) &= \det(s\mathbf{I} - \mathbf{A}_c) \\ &= s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n. \end{aligned} \tag{WB.41}$$

WB.11 Cayley–Hamilton Theorem

The Cayley–Hamilton theorem states that every square matrix $\mathbf{A}$ satisfies its characteristic polynomial. This means that if $\mathbf{A}$ is an $n \times n$ matrix with characteristic equation $a(s)$, then

$$a(\mathbf{A}) \triangleq \mathbf{A}^n + a_1\mathbf{A}^{n-1} + \cdots + a_{n-1}\mathbf{A} + a_n\mathbf{I} = 0 \tag{WB.42}$$

WB.12 Eigenvalues and Eigenvectors

Any scalar λ and nonzero vector $\mathbf{v}$ that satisfy

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \tag{WB.43}$$

are referred to as the eigenvalue and the associated **(right) eigenvector** of the matrix $\mathbf{A}$ [because $\mathbf{v}$ appears to the right of $\mathbf{A}$ in Eq. (WB.43)]. By rearranging terms in Eq. (WB.43), we get

$$(\lambda\mathbf{I} - \mathbf{A})\mathbf{v} = 0. \tag{WB.44}$$

Because $\mathbf{v}$ is nonzero,

$$\det(\lambda\mathbf{I} - \mathbf{A}) = 0, \tag{WB.45}$$

so λ is an eigenvalue of the matrix $\mathbf{A}$ as defined in Eq. (WB.43). The normalization of the eigenvectors is arbitrary; that is, if $\mathbf{v}$ is an eigenvector, so is $\alpha\mathbf{v}$. The eigenvectors are usually normalized to have unit length; that is, $\|\mathbf{v}\|^2 = \mathbf{v}^T\mathbf{v} = 1$.

If $\mathbf{w}^T$ is a nonzero row vector such that

$$\mathbf{w}^T\mathbf{A} = \lambda\mathbf{w}^T, \tag{WB.46}$$

then $\mathbf{w}$ is called a **left eigenvector** of $\mathbf{A}$ [because $\mathbf{w}^T$ appears to the left of $\mathbf{A}$ in Eq. (WB.46)]. Note that we can write

$$\mathbf{A}^T\mathbf{w} = \lambda\mathbf{w} \tag{WB.47}$$

so that $\mathbf{w}$ is simply a right eigenvector of $\mathbf{A}^T$.

WB.13 Similarity Transformations

Consider the arbitrary nonsingular matrix $\mathbf{T}$ such that

$$\bar{\mathbf{A}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}. \tag{WB.48}$$

The matrix operation shown in Eq. (WB.48) is referred to as a **similarity transformation**. If $\mathbf{A}$ has a full set of eigenvectors, then we can choose $\mathbf{T}$ to be the set of eigenvectors and $\bar{\mathbf{A}}$ will be diagonal.

Consider the set of equations in state-variable form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u. \tag{WB.49}$$

If we let

$$\mathbf{T}\xi = \mathbf{x}, \tag{WB.50}$$

then Eq. (WB.49) becomes

$$\mathbf{T}\dot{\xi} = \mathbf{A}\mathbf{T}\xi + \mathbf{B}u, \tag{WB.51}$$

and premultiplying both sides by $\mathbf{T}^{-1}$, we get

$$\begin{aligned}\dot{\xi} &= \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\xi + \mathbf{T}^{-1}\mathbf{B}u \\ &= \bar{\mathbf{A}}\xi + \bar{\mathbf{B}}u,\end{aligned} \tag{WB.52}$$

where

$$\begin{aligned}\bar{\mathbf{A}} &= \mathbf{T}^{-1}\mathbf{A}\mathbf{T}, \\ \bar{\mathbf{B}} &= \mathbf{T}^{-1}\mathbf{B}.\end{aligned} \tag{WB.53}$$

The characteristic polynomial of $\bar{\mathbf{A}}$ is

$$\begin{aligned}\det(s\mathbf{I} - \bar{\mathbf{A}}) &= \det(s\mathbf{I} - \mathbf{T}^{-1}\mathbf{A}\mathbf{T}) \\ &= \det(s\mathbf{T}^{-1}\mathbf{T} - \mathbf{T}^{-1}\mathbf{A}\mathbf{T}) \\ &= \det[\mathbf{T}^{-1}(s\mathbf{I} - \mathbf{A})\mathbf{T}] \\ &= \det \mathbf{T}^{-1} \det(s\mathbf{I} - \mathbf{A}) \det \mathbf{T}.\end{aligned} \tag{WB.54}$$

Using Eq. (WB.29), Eq. (WB.54) becomes

$$\det(s\mathbf{I} - \bar{\mathbf{A}}) = \det(s\mathbf{I} - \mathbf{A}). \tag{WB.55}$$

From Eq. (WB.55) we can see that $\bar{\mathbf{A}}$ and $\mathbf{A}$ both have the same characteristic polynomial, giving us the important result that a similarity transformation does not change the eigenvalues of a matrix. From Eq. (WB.50) a new state made up of a linear combination from the old state has the same eigenvalues as the old set.

WB.14 Matrix Exponential

Let $\mathbf{A}$ be a square matrix. The **matrix exponential** of $\mathbf{A}$ is defined as the series

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \frac{1}{2!}\mathbf{A}^2t^2 + \frac{\mathbf{A}^3t^3}{3!} + \cdots. \tag{WB.56}$$

It can be shown that the series converges. If $\mathbf{A}$ is an $n \times n$ matrix, then $e^{\mathbf{A}t}$ is also an $n \times n$ matrix and can be differentiated:

$$\frac{d}{dt}e^{\mathbf{A}t} = \mathbf{A}e^{\mathbf{A}t}. \tag{WB.57}$$

Other properties of the matrix exponential are

$$e^{\mathbf{A}t_1}e^{\mathbf{A}t_2} = e^{\mathbf{A}(t_1+t_2)} \tag{WB.58}$$

and, in general,

$$e^{\mathbf{A}}e^{\mathbf{B}} \neq e^{\mathbf{B}}e^{\mathbf{A}}. \tag{WB.59}$$

(In the exceptional case where $\mathbf{A}$ and $\mathbf{B}$ commute—that is, $\mathbf{AB} = \mathbf{BA}$—then $e^{\mathbf{A}}e^{\mathbf{B}} = e^{\mathbf{B}}e^{\mathbf{A}}$.)

WB.15 Fundamental Subspaces

The **range space** of $\mathbf{A}$, denoted by $\mathcal{R}(\mathbf{A})$ and also called the **column space** of $\mathbf{A}$, is defined by the set of vectors

$$\mathbf{x} = \mathbf{A}\mathbf{y} \tag{WB.60}$$

for some vector $\mathbf{y}$. The **null space** of $\mathbf{A}$, denoted by $\mathcal{N}(\mathbf{A})$, is defined by the set of vectors $\mathbf{x}$ such that

$$\mathbf{A}\mathbf{x} = \mathbf{0}. \tag{WB.61}$$

If $x \in \mathcal{N}(\mathbf{A})$ and $y \in \mathcal{R}(\mathbf{A}^T)$, then $\mathbf{y}^T\mathbf{x} = 0$; that is, every vector in the null space of $\mathbf{A}$ is **orthogonal** to every vector in the range space of $\mathbf{A}^T$.

WB.16 Singular-Value Decomposition

The **singular-value decomposition (SVD)** is one of the most useful tools in linear algebra and has been widely used in control theory during the last few decades. Let $\mathbf{A}$ be an $m \times n$ matrix. Then there always exist matrices $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$ such that

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{WB.62}$$

Here $\mathbf{U}$ and $\mathbf{V}$ are **orthogonal matrices**; that is,

$$\mathbf{U}\mathbf{U}^T = \mathbf{I}, \mathbf{V}\mathbf{V}^T = \mathbf{I}. \tag{WB.63}$$

$\mathbf{S}$ is a **quasidiagonal matrix** with singular values as its diagonal elements; that is,

$$\mathbf{S} = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}, \tag{WB.64}$$

where Σ is a diagonal matrix of nonzero singular values in descending order:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0. \tag{WB.65}$$

The unique diagonal elements of $\mathbf{S}$ are called the **singular values**. The maximum singular value is denoted by $\bar{\sigma}(A)$, and the minimum singular value is denoted by $\underline{\sigma}(\mathbf{A})$. The rank of the matrix is the same as the number of nonzero singular values. The columns of $\mathbf{U}$ and $\mathbf{V}$,

$$\begin{aligned} \mathbf{U} &= [\ u_1 \quad u_2 \quad \ldots \quad u_m\ ], \\ \mathbf{V} &= [\ \upsilon_1 \quad \upsilon_2 \quad \ldots \quad \upsilon_n\ ], \end{aligned} \tag{WB.66}$$

are called the left and right **singular vectors**, respectively. SVD provides complete information about the fundamental subspaces associated with a matrix:

$$\begin{aligned}
\mathcal{N}(\mathbf{A}) &= \text{span}[\ v_{r+1} \quad v_{r+2} \quad \ldots \quad v_n\ ], \\
\mathcal{R}(\mathbf{A}) &= \text{span}[\ u_1 \quad u_2 \quad \ldots \quad u_r\ ], \\
\mathcal{R}(\mathbf{A}^T) &= \text{span}[\ v_1 \quad v_2 \quad \ldots \quad v_r\ ], \\
\mathcal{N}(\mathbf{A}^T) &= \text{span}[\ u_{r+1} \quad u_{r+2} \quad \ldots \quad u_m\ ].
\end{aligned} \tag{WB.67}$$

Here $\mathcal{N}$ denotes the null space and $\mathcal{R}$, the range space, respectively.

The **norm** of the matrix $\mathbf{A}$, denoted by $\|\mathbf{A}\|_2$, is given by

$$\|\mathbf{A}\|_2 = \bar{\sigma}(\mathbf{A}). \tag{WB.68}$$

If $\mathbf{A}$ is a function of ω, then the infinity norm of $\mathbf{A}$, $\|\mathbf{A}\|_\infty$, is given by

$$\|\mathbf{A}(j\omega)\|_\infty = \max_\omega \bar{\sigma}(\mathbf{A}). \tag{WB.69}$$

WB.17 Positive Definite Matrices

A matrix $\mathbf{A}$ is said to be **positive semidefinite** if

$$\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0 \quad \text{for all } \mathbf{x}. \tag{WB.70}$$

The matrix is said to be **positive definite** if equality holds in Eq. (WB.70) only for $\mathbf{x} = 0$. A symmetric matrix is positive definite if and only if all of its eigenvalues are positive. It is positive semidefinite if and only if all of its eigenvalues are nonnegative.

An alternate method for determining positive definiteness is to test the minors of the matrix. A matrix is positive definite if all the leading principal minors are positive, and positive semidefinite if they are all nonnegative.

WB.18 Matrix Identity

If $\mathbf{A}$ is an $n \times m$ matrix and $\mathbf{B}$ is an $m \times n$ matrix, then

$$\det[\mathbf{I}_n - \mathbf{A}\mathbf{B}] = \det[\mathbf{I}_m - \mathbf{B}\mathbf{A}],$$

where $\mathbf{I}_n$ and $\mathbf{I}_m$ are identity matrices of size n and m, respectively.

Appendix WC
Controllability and Observability

Controllability and observability are important structural properties of dynamic systems. First identified and studied by Kalman (1960) and later by Kalman et al. (1961), these properties have continued to be examined during the last five decades. We will discuss only a few of the known results for linear constant systems with one input and one output. In the text, we discuss these concepts in connection with control law and estimator designs. For example, in Section 7.4 we suggest that, if the square matrix given by

$$\mathcal{C} = [\mathbf{B}\ \mathbf{AB}\ \mathbf{A}^2\mathbf{B}\ \ldots\ \mathbf{A}^{n-1}\mathbf{B}] \tag{WC.1}$$

is nonsingular, by transformation of the state we can convert the given description into control canonical form. We can then construct a control law that will give the closed-loop system an arbitrary characteristic equation.

WC.1 Controllability

We begin our formal discussion of controllability with the first of four definitions.

Definition WC.1 *The system* $(\mathbf{A}, \mathbf{B})$ *is* ***controllable*** *if, for any given nth-order polynomial* $\alpha_c(s)$*, there exists a (unique) control law* $u = -\mathbf{Kx}$ *such that the characteristic polynomial of* $\mathbf{A} - \mathbf{BK}$ *is* $\alpha_c(s)$.

From the results of Ackermann's formula (see Appendix WD), we have the following mathematical test for controllability: $(\mathbf{A}, \mathbf{B})$ is a controllable pair if and only if the rank of $\mathcal{C}$ is n. Definition WC.1 based on pole placement is a frequency-domain concept. Controllability can be equivalently defined in the time domain.

Definition WC.2 *The system* $(\mathbf{A}, \mathbf{B})$ *is* ***controllable*** *if there exists a (piecewise continuous) control signal* $u(t)$ *that will take the state of the system from any initial state* $\mathbf{x}_0$ *to any desired final state* $\mathbf{x}_f$ *in a finite time interval.*

We will now show that the system is controllable by this definition if and only if $\mathcal{C}$ is full rank. We first assume that the system is controllable but that

$$\text{rank}[\mathbf{B}\ \mathbf{AB}\ \mathbf{A}^2\mathbf{B}\ \ldots\ \mathbf{A}^{n-1}\mathbf{B}] < n. \tag{WC.2}$$

We can then find a vector $\mathbf{v}$ such that

$$\mathbf{v}[\mathbf{B}\ \mathbf{AB}\ \mathbf{A}^2\mathbf{B}\ \ldots\ \mathbf{A}^{n-1}\mathbf{B}] = 0 \tag{WC.3}$$

or

$$\mathbf{vB} = \mathbf{vAB} = \mathbf{vA}^2\mathbf{B} = \ldots = \mathbf{vA}^{n-1}\mathbf{B} = 0. \tag{WC.4}$$

The Cayley-Hamilton theorem states that $\mathbf{A}$ satisfies its own characteristic equation, namely,

$$-\mathbf{A}^n = a_1\mathbf{A}^{n-1} + a_2\mathbf{A}^{n-2} + \ldots + a_n\mathbf{I}. \tag{WC.5}$$

Therefore,

$$-\mathbf{vA}^n\mathbf{B} = a_1\mathbf{vA}^{n-1}\mathbf{B} + a_2\mathbf{vA}^{n-2}\mathbf{B} + \ldots + a_n\mathbf{vB} = 0. \tag{WC.6}$$

By induction, $\mathbf{vA}^{n+k}\mathbf{B} = 0$ for $k = 0, 1, 2, \ldots$, or $\mathbf{vA}^m\mathbf{B} = 0$ for $m = 0, 1, 2, \ldots$, and thus

$$\mathbf{v}e^{\mathrm{A}t}\mathbf{B} = \mathbf{v}\left(\mathbf{I} + \mathbf{A}t + \frac{1}{2!}\mathbf{A}^2t^2 + \ldots\right)\mathbf{B} = 0, \tag{WC.7}$$

for all t. However, the zero initial-condition response ($\mathbf{x}_0 = \mathbf{0}$) is

$$\begin{aligned}\mathbf{x}(t) &= \int_0^t \mathbf{v}e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)\,d\tau \\ &= e^{\mathbf{A}t}\int_0^t e^{-A\tau}\mathbf{B}u(\tau)\,d\tau.\end{aligned} \tag{WC.8}$$

Using Eq. (WC.7), Eq. (WC.8) becomes

$$\mathbf{vx}(t) = \int_0^t \mathbf{v}e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)\,d\tau = 0 \tag{WC.9}$$

for all $u(t)$ and $t > 0$. This implies that all points reachable from the origin are orthogonal to $\mathbf{v}$. This restricts the reachable space and therefore contradicts the second definition of controllability. Thus if $\mathcal{C}$ is singular, ($\mathbf{A}$, $\mathbf{B}$) is not controllable by Definition WC.2.

Next we assume that $\mathcal{C}$ is full rank but ($\mathbf{A}$, $\mathbf{B}$) is uncontrollable by Definition WC.2. This means that there exists a nonzero vector $\mathbf{v}$ such that

$$\mathbf{v}\int_0^{t_f} e^{\mathbf{A}^{(t_f-\tau)}}\mathbf{B}u(\tau)\,d\tau = 0, \tag{WC.10}$$

because the whole state-space is not reachable. But Eq. (WC.10) implies that

$$\mathbf{v}e^{\mathbf{A}^{(t_f-\tau)}}\mathbf{B} = 0, \quad 0 \le \tau \le t_f. \tag{WC.11}$$

If we set $\tau = t_f$, we see that $\mathbf{vB} = 0$. Also, differentiating Eq. (WC.11) and letting $\tau = t_f$ gives $\mathbf{vAB} = 0$. Continuing this process, we find that

$$\mathbf{vB} = \mathbf{vAB} = \mathbf{vA}^2\mathbf{B} = \ldots = \mathbf{vA}^{n-1}\mathbf{B} = 0, \tag{WC.12}$$

which contradicts the assumption that $\mathcal{C}$ is full rank.

We have now shown that the system is controllable by Definition WC.2 if and only if the rank of $\mathcal{C}$ is n, which is exactly the same condition we found for pole assignment.

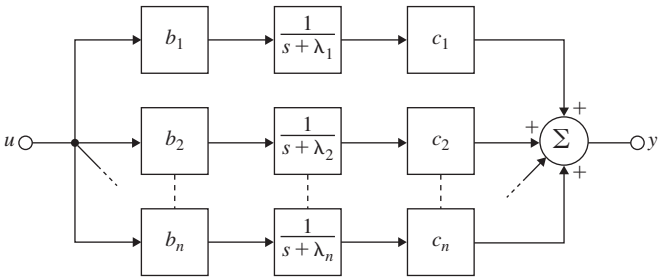


Figure WC.1
Block diagram of a system with a diagonal matrix

Our final definition comes closest to the structural character of controllability.

Definition WC.3 *The system (***A**, **B***) is* ***controllable*** *if every mode of* **A** *is connected to the control input.*

Because of the generality of the modal structure of systems, we will treat only the case of systems for which **A** can be transformed to diagonal form. (The double-integration plant does *not* qualify.) Suppose we have a diagonal matrix $\mathbf{A}_d$ and its corresponding input matrix $\mathbf{B}_d$ with elements b_i. The structure of such a system is shown in Fig. (WC.1). By definition for a controllable system, the input must be connected to each mode so that the b_i are all nonzero. However, this is not enough if the poles (λ_i) are not distinct. Suppose, for instance, that $\lambda_1 = \lambda_2$. The first two state equations are then

$$\begin{aligned} \dot{x}_{1d} &= \lambda_1 x_{1d} + b_1 u, \\ \dot{x}_{2d} &= \lambda_1 x_{2d} + b_2 u. \end{aligned} \tag{WC.13}$$

If we define a new state, $\xi = b_2 x_{1d} - b_1 x_{2d}$, the equation for ξ is

$$\dot{\xi} = b_2\dot{x}_{1d} - b_1\dot{x}_{2d} = b_2\lambda_1 x_{1d} + b_2 b_1 u - b_1\lambda_1 x_{2d} - b_1 b_2 u = \lambda_1 \xi, \tag{WC.14}$$

which does not include the control u; hence, ξ is not controllable. The point is that if any two poles are equal *in a diagonal* $\mathbf{A}_d$ *system with only one input*, we effectively have a hidden mode that is not connected to the control, and the system is not controllable (Fig. WC.2a). This is because the two state variables move together exactly, so we cannot *independently* control x_{1d}

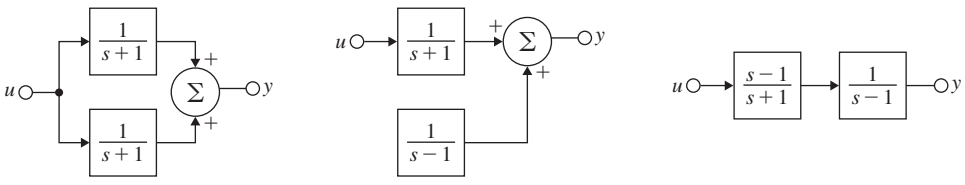


Figure WC.2
Examples of uncontrollable systems

and x_{2d}. Therefore, even in such a simple case, we have two conditions for controllability:

1. All eigenvalues of $\mathbf{A}_d$ are distinct.
2. No element of $\mathbf{B}_d$ is zero.

Now let us consider the controllability matrix of this diagonal system. By direct computation,

$$\begin{aligned}\mathcal{C} &= \begin{bmatrix} b_1 & b_1\lambda_1 & \dots & b_1\lambda_1^{n-1} \\ b_2 & b_2\lambda_2 & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ b_n & b_n\lambda_n & \dots & b_n\lambda_n^{n-1} \end{bmatrix} \\ &= \begin{bmatrix} b_1 & & & \mathbf{0} \\ & b_2 & & \\ & & \ddots & \\ \mathbf{0} & & & b_n \end{bmatrix} \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \lambda_2^2 & \dots & \vdots \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \lambda_n & \lambda_n^2 & \dots & \lambda_n^{n-1} \end{bmatrix}. \end{aligned} \tag{WC.15}$$

Note that the controllability matrix $\mathcal{C}$ is the product of two matrices and is nonsingular if and only if both of these matrices are invertible. The first matrix has a determinant that is the product of b_i, and the second matrix (called a Vandermonde matrix) is nonsingular if and only if the λ_i are distinct. Thus Definition WC.3 is equivalent to having a nonsingular $\mathcal{C}$ also.

Important to the subject of controllability is the **Popov–Hautus–Rosenbrock (PHR) test** (see Rosenbrock, 1970, and Kailath, 1980), which is an alternate way to test the rank (or determinant) of $\mathcal{C}$. The system $(\mathbf{A}, \mathbf{B})$ is controllable if the system of equations

$$\mathbf{v}^T[s\mathbf{I} - \mathbf{A} \quad \mathbf{B}] = \mathbf{0}^T, \tag{WC.16}$$

has only the trivial solution $\mathbf{v}^T = \mathbf{0}^T$—that is, if the **matrix pencil**

$$[s\mathbf{I} - \mathbf{A} \quad \mathbf{B}] = n, \tag{WC.17}$$

is full rank for all s, or if there is no nonzero $\mathbf{v}^T$ such that[1]

$$\mathbf{v}^T\mathbf{A} = s\mathbf{v}^T, \tag{WC.18}$$

$$\mathbf{v}^T\mathbf{B} = 0. \tag{WC.19}$$

This test is equivalent to the rank-of-$\mathcal{C}$ test. It is easy to show that, if such a vector $\mathbf{v}$ exists, $\mathcal{C}$ is singular. For if a nonzero $\mathbf{v}$ exists such that $\mathbf{v}^T\mathbf{B} = 0$ then by Eqs. (WC.18) and (WC.19), we have

$$\mathbf{v}^T\mathbf{A}\mathbf{B} = s\mathbf{v}^T\mathbf{B} = 0. \tag{WC.20}$$

Then, multiplying by $\mathbf{AB}$, we find that

$$\mathbf{v}^T\mathbf{A}^2\mathbf{B} = s\mathbf{v}^T\mathbf{A}\mathbf{B} = 0, \tag{WC.21}$$

[1] $\mathbf{v}^T$ is a left eigenvector of $\mathbf{A}$.

and so on. Thus we determine that $\mathbf{v}^T\mathcal{C} = \mathbf{0}^T$ has a nontrivial solution, that $\mathcal{C}$ is singular, and that the system is not controllable. To show that a nontrivial $\mathbf{v}^T$ exists if $\mathcal{C}$ is singular requires more development, which we will not give here (see Kailath, 1980).

We have given two pictures of uncontrollability. Either a mode is physically disconnected from the input (Fig. WC.2b), or else two parallel subsystems have identical characteristic roots (Fig. WC.2a). The control engineer should be aware of the existence of a third simple situation, illustrated in Fig. WC.2c, namely, a **pole-zero cancellation**. Here the problem is that the mode at $s = 1$ appears to be connected to the input but is masked by the zero at $s = 1$ in the preceding subsystem; the result is an uncontrollable system. This can be confirmed in several ways. First let us look at the controllability matrix. The system matrices are

$$\mathbf{A} = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -2 \\ 1 \end{bmatrix},$$

so the controllability matrix is

$$\mathcal{C} = [\ \mathbf{B} \quad \mathbf{AB}\ ] = \begin{bmatrix} -2 & 2 \\ 1 & -1 \end{bmatrix}, \tag{WC.22}$$

which is clearly singular. The controllability matrix may be computed using the `ctrb` command in Matlab: `[cc]=ctrb(A,B)`. If we compute the transfer function from u to x_2, we find that

$$\mathbf{H}(s) = \frac{s-1}{s+1}\left(\frac{1}{s-1}\right) = \frac{1}{s+1}. \tag{WC.23}$$

Because the natural mode at $s = 1$ disappears from the input–output description, it is not connected to the input. Finally, if we consider the **PHR** test,

$$[s\mathbf{I} - \mathbf{A} \quad \mathbf{B}] = \begin{bmatrix} s+1 & 0 & -2 \\ -1 & s-1 & 1 \end{bmatrix}, \tag{WC.24}$$

and let $s = 1$, then we must test the rank of

$$\begin{bmatrix} 2 & 0 & -2 \\ -1 & 0 & 1 \end{bmatrix},$$

which is clearly less than 2. This result means, again, that the system is uncontrollable.

Definition WC.4 *The asymptotically stable system* $(\mathbf{A}, \mathbf{B})$ *is* ***controllable*** *if the controllability Gramian, the square symmetric matrix* $\mathcal{C}_g$*, given by the solution to the Lyapunov equation*

$$\mathbf{A}\mathcal{C}_g + \mathcal{C}_g\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = \mathbf{0} \tag{WC.25}$$

is nonsingular. The controllability ***Gramian*** *is also the solution to the following integral equation:*

$$\mathcal{C}_g = \int_0^\infty e^{\tau\mathbf{A}}\mathbf{B}\mathbf{B}^T e^{\tau\mathbf{A}^T} d\tau. \tag{WC.26}$$

One physical interpretation of the controllability Gramian is that, if the input to the system is white Gaussian noise, $\mathcal{C}_g$ is the covariance of the state. The controllability Gramian (for an asymptotically stable system) can be computed with the following command in Matlab: [cg]=gram(A,B).

In conclusion, the four definitions for controllability—pole assignment (Definition WC.1), state reachability (Definition WC.2), mode coupling to the input (Definition WC.3), and controllability Gramian (Definition WC.4)—are equivalent. The tests for any of these four properties are found in terms of the rank of the controllability, controllability Gramian matrices, or the rank of the **matrix pencil** $[s\mathbf{I}-\mathbf{A}\ \mathbf{B}]$. If $\mathcal{C}$ is nonsingular, we can assign the closed-loop poles arbitrarily by state feedback, we can move the state to any point in the state space in a finite time, and every mode is connected to the control input.[2]

WC.2 Observability

So far we have discussed only controllability. The concept of observability is parallel to that of controllability, and all of the results we have discussed thus far may be transformed to statements about observability by invoking the property of duality, as discussed in Section 7.7.2. The observability definitions are analogous to those for controllability.

Definition WC.1: The system (**A**, **C**) is **observable** if, for any nth-order polynomial $\alpha_e(s)$, there exists an estimator gain **L** such that the characteristic equation of the state estimator error is $\alpha_e(s)$.
Definition WC.2: The system (**A**, **C**) is **observable** if, for any **x(0)**, there is a finite time τ such that **x(0)** can be determined (uniquely) from $u(t)$ and $y(t)$ for $0 \leq t \leq \tau$.
Definition WC.3: The system (**A**, **C**) is **observable** if every dynamic mode in **A** is connected to the output through **C**.
Definition WC.4: The asymptotically stable system (**A**, **C**) is **observable** if the observability Gramian is nonsingular.

As we saw in the discussion for controllability, mathematical tests can be developed for observability. The system is observable if the observability matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{WC.27}$$

is nonsingular. If we take the transpose of $\mathcal{O}$ and let $\mathbf{C}^T = \mathbf{B}$ and $\mathbf{A}^T = \mathbf{A}$, then we find the controllability matrix of (**A**, **B**), which is another manifestation of duality. The observability matrix $\mathcal{O}$ may be computed

[2]We have shown the latter for diagonal **A** only, but the result is true in general.

using the obsv command in Matlab: [oo]=obsv(A,C). The system ($\mathbf{A}$, $\mathbf{C}$) is observable if the following **matrix pencil** is full rank for all s:

$$\text{rank}\begin{bmatrix} s\mathbf{I}-\mathbf{A} \\ \mathbf{C} \end{bmatrix} = n. \tag{WC.28}$$

The observability Gramian $\mathcal{O}_g$, which is a symmetric matrix, and the solution to the integral equation

$$\mathcal{O}_g = \int_0^\infty e^{\tau\mathbf{A}^T}\mathbf{C}^T\mathbf{C}e^{\tau\mathbf{A}}\,d\tau, \tag{WC.29}$$

as well as the Lyapunov equation

$$\mathbf{A}^T\mathcal{O}_g + \mathcal{O}_g\mathbf{A} + \mathbf{C}^T\mathbf{C} = \mathbf{0}, \tag{WC.30}$$

also can be computed (for an asymptotically stable system) using the gram command in Matlab: [og]=gram(A',C'). The observability Gramian has an interpretation as the "information matrix" in the context of estimation.

Appendix WD
Ackermann's Formula for Pole Placement

Given the plant and state-variable equation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u, \tag{WD.1}$$

our objective is to find a state-feedback control law

$$u = -\mathbf{K}\mathbf{x} \tag{WD.2}$$

such that the closed-loop characteristic polynomial is

$$\alpha_c(s) = \det(s\mathbf{I} - \mathbf{A} + \mathbf{B}\mathbf{K}). \tag{WD.3}$$

First we have to select $\alpha_c(s)$, which determines where the poles are to be shifted; then we have to find $\mathbf{K}$ such that Eq. (WD.3) will be satisfied. Our technique is based on transforming the plant equation into control canonical form.

We begin by considering the effect of an arbitrary nonsingular transformation of the state, as

$$\mathbf{x} = \mathbf{T}\bar{\mathbf{x}}, \tag{WD.4}$$

where $\bar{\mathbf{x}}$ is the new transformed state. The equations of motion in the new coordinates, from Eq. (WD.4), are

$$\dot{\mathbf{x}} = \mathbf{T}\dot{\bar{\mathbf{x}}} = \mathbf{A}\mathbf{x} + \mathbf{B}u = \mathbf{A}\mathbf{T}\bar{\mathbf{x}} + \mathbf{B}u, \tag{WD.5}$$

$$\dot{\bar{\mathbf{x}}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\bar{\mathbf{x}} + \mathbf{T}^{-1}\mathbf{B}u = \bar{\mathbf{A}}\bar{\mathbf{x}} + \bar{\mathbf{B}}u. \tag{WD.6}$$

Now the controllability matrix for the original state,

$$\mathcal{C}_{\mathbf{x}} = [\ \mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}\ ], \tag{WD.7}$$

provides a useful transformation matrix. We can also define the controllability matrix for the transformed state:

$$\mathcal{C}_{\bar{\mathbf{x}}} = [\ \bar{\mathbf{B}} \quad \bar{\mathbf{A}}\bar{\mathbf{B}} \quad \bar{\mathbf{A}}^2\bar{\mathbf{B}} \quad \cdots \quad \mathbf{A}^{n-1}\bar{\mathbf{B}}\ ]. \tag{WD.8}$$

The two controllability matrices are related by

$$\mathcal{C}_{\bar{\mathbf{x}}} = [\ \mathbf{T}^{-1}\mathbf{B} \quad \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\mathbf{T}^{-1}\mathbf{B} \quad \cdots\ ] = \mathbf{T}^{-1}\mathcal{C}_{\mathbf{x}} \tag{WD.9}$$

and the transformation matrix

$$\mathbf{T} = \mathcal{C}_{\mathbf{x}}\mathcal{C}_{\bar{\mathbf{x}}}^{-1}. \tag{WD.10}$$

From Eqs. (WD.9) and (WD.10), we can draw some important conclusions. From Eq. (WD.9), we see that if $\mathcal{C}_{\mathbf{x}}$ is nonsingular, then for any nonsingular $\mathbf{T}$, $\mathcal{C}_{\bar{\mathbf{x}}}$ is also nonsingular. This means that a similarity

transformation on the state does not change the controllability properties of a system. We can look at this in another way. Suppose we would like to find a transformation to take the system $(\mathbf{A}, \mathbf{B})$ into control canonical form. As we shall shortly see, $\mathcal{C}_{\bar{\mathbf{x}}}$ in that case is *always* nonsingular. From Eq. (WD.9), we see that a nonsingular $\mathbf{T}$ will always exist if and only if $\mathcal{C}_{\mathbf{x}}$ is nonsingular. We derive the following theorem.

Theorem WD.1 *We can always transform* $(\mathbf{A}, \mathbf{B})$ *into control canonical form if and only if the system is controllable.*

Let us take a closer look at control canonical form and treat the third-order case, although the results are true for any nth-order case:

$$\bar{\mathbf{A}} = \mathbf{A}_c = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \bar{\mathbf{B}} = \mathbf{B}_c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{WD.11}$$

The controllability matrix, by direct computation, is

$$\mathcal{C}_{\bar{\mathbf{x}}} = \mathcal{C}_c = \begin{bmatrix} 1 & -a_1 & a_1^2 - a_2 \\ 0 & 1 & -a_1 \\ 0 & 0 & 1 \end{bmatrix}. \tag{WD.12}$$

Because this matrix is upper triangular with ones along the diagonal, it is always invertible. Also note that the last row of $\mathcal{C}_{\bar{\mathbf{x}}}$ is the unit vector with all zeros, except for the last element, which is unity. We shall use this fact in what we do next.

As we pointed out in Section 7.5, the design of a control law for the state $\bar{\mathbf{x}}$ is trivial if the equations of motion happen to be in control canonical form. The characteristic equation is

$$s^3 + a_1 s^2 + a_2 s + a_3 = 0, \tag{WD.13}$$

and the characteristic equation for the closed-loop system comes from

$$\mathbf{A}_{cl} = \mathbf{A}_c - \mathbf{B}_c \mathbf{K}_c \tag{WD.14}$$

and has the coefficients shown:

$$s^3 + (a_1 + K_{c1})s^2 + (a_2 + K_{c2})s + (a_3 + K_{c3}) = 0. \tag{WD.15}$$

To obtain the desired closed-loop pole locations, we must make the coefficients of s in Eq. (WD.15) match those in

$$\alpha_c(s) = s^3 + \alpha_1 s^2 + \alpha_2 s + \alpha_3, \tag{WD.16}$$

so

$$a_1 + K_{c1} = \alpha_1, \quad a_2 + K_{c2} = \alpha_2, \quad a_3 + K_{c3} = \alpha_3, \tag{WD.17}$$

or in vector form,

$$\mathbf{a} + \mathbf{K}_c = \alpha, \tag{WD.18}$$

where $\mathbf{a}$ and α are row vectors containing the coefficients of the characteristic polynomials of the open-loop and closed-loop systems, respectively.

We now need to find a relationship between these polynomial coefficients and the matrix $\mathbf{A}$. The requirement is achieved by the Cayley–Hamilton theorem, which states that a matrix satisfies its own characteristic polynomial. For $\mathbf{A}_c$, this means that

$$\mathbf{A}_c^n + a_1\mathbf{A}_c^{n-1} + a_2\mathbf{A}_c^{n-2} + \cdots + a_n\mathbf{I} = \mathbf{0}. \tag{WD.19}$$

Now suppose we form the polynomial $\alpha_c(\mathbf{A})$, which is the *closed-loop* characteristic polynomial with the matrix $\mathbf{A}$ substituted for the complex variable s:

$$\alpha_c(\mathbf{A}_c) = \mathbf{A}_c^n + \alpha_1\mathbf{A}_c^{n-1} + \alpha_2\mathbf{A}_c^{n-2} + \cdots + \alpha_n\mathbf{I}. \tag{WD.20}$$

If we solve Eq. (WD.19) for $\mathbf{A}_c^n$ and substitute into Eq. (WD.20), we find that

$$\alpha_c(\mathbf{A}_c) = (-a_1 + \alpha_1)\mathbf{A}_c^{n-1} + (-a_2 + \alpha_2)\mathbf{A}_c^{n-2} + \cdots + (-\alpha_n + \alpha_n)\mathbf{I}. \tag{WD.21}$$

But, because $\mathbf{A}_c$ has such a special structure, we observe that if we multiply it by the transpose of the nth unit vector, $\mathbf{e}_n^T = [\ 0 \quad \cdots \quad 0 \quad 1\ ]$, we get

$$\mathbf{e}_n^T\mathbf{A}_c = [\ 0 \quad \cdots \quad 0 \quad 1 \quad 0\ ] = \mathbf{e}_{n-1}^T, \tag{WD.22}$$

as we can see from Eq. (WD.11). If we multiply this vector again by $\mathbf{A}_c$, getting

$$\begin{aligned}(\mathbf{e}_n^T\mathbf{A}_c)\mathbf{A}_c &= [\ 0 \quad \cdots \quad 0 \quad 1 \quad 0\ ]\mathbf{A}_c \\ &= [\ 0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad 0\ ] = \mathbf{e}_{n-2}^T,\end{aligned} \tag{WD.23}$$

and continue the process, successive unit vectors are generated until

$$\mathbf{e}_n^T\mathbf{A}_c^{n-1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} = \mathbf{e}_1^T. \tag{WD.24}$$

Therefore, if we multiply Eq. (WD.21) by $\mathbf{e}_n^T$, we find that

$$\begin{aligned}\mathbf{e}_n^T\alpha_c(\mathbf{A}_c) &= (-a_1 + \alpha_1)\mathbf{e}_1^T + (-a_2 + \alpha_2)\mathbf{e}_2^T + \cdots + (-a_n + \alpha_n)\mathbf{e}_n^T \\ &= [\ K_{c1} \quad K_{c2} \quad \cdots \quad K_{cn}\ ] = \mathbf{K}_c,\end{aligned} \tag{WD.25}$$

where we use Eq. (WD.18), which relates $\mathbf{K}_c$ to $\mathbf{a}$ and α.

We now have a compact expression for the gains of the system in control canonical form as represented in Eq. (WD.25). However, we still need the expression for $\mathbf{K}$, which is the gain on the original state. If $u = -\mathbf{K}_c\bar{\mathbf{x}}$, then $u = -\mathbf{K}_c\mathbf{T}^{-1}\mathbf{x}$, so that

$$\begin{aligned}\mathbf{K} &= \mathbf{K}_c\mathbf{T}^{-1} = \mathbf{e}_n^T\alpha_c(\mathbf{A}_c)\mathbf{T}^{-1} \\ &= \mathbf{e}_n^T\alpha_c(\mathbf{T}^{-1}\mathbf{A}\mathbf{T})\mathbf{T}^{-1} \\ &= \mathbf{e}_n^T\mathbf{T}^{-1}\alpha_c(\mathbf{A}).\end{aligned} \tag{WD.26}$$

In the last step of Eq. (WD.26), we used the fact that $(\mathbf{T}^{-1}\mathbf{A}\mathbf{T})^k = \mathbf{T}^{-1}\mathbf{A}^k\mathbf{T}$ and that α_c is a polynomial, that is, a sum of the powers of $\mathbf{A}_c$. From Eq. (WD.9), we see that

$$\mathbf{T}^{-1} = \mathcal{C}_c\mathcal{C}_{\mathbf{x}}^{-1}. \tag{WD.27}$$

With this substitution, Eq. (WD.26) becomes

$$\mathbf{K} = \mathbf{e}_n^T \mathcal{C}_c \mathcal{C}_{\mathbf{x}}^{-1} \alpha_c(\mathbf{A}). \tag{WD.28}$$

Ackermann's formula

Now, we use the observation made earlier for Eq. (WD.12) that the last row of $\mathcal{C}_c$, which is $\mathbf{e}_n^T \mathcal{C}_c$, is again $\mathbf{e}_n^T$. We finally obtain Ackermann's formula:

$$\mathbf{K} = \mathbf{e}_n^T \mathcal{C}_{\mathbf{x}}^{-1} \alpha_c(\mathbf{A}). \tag{WD.29}$$

We note again that forming the explicit inverse of $\mathcal{C}_{\mathbf{x}}$ is not advisable for numerical accuracy. Thus we need to solve $\mathbf{b}^T$ such that

$$\mathbf{e}_n^T \mathcal{C}_{\mathbf{x}}^{-1} = \mathbf{b}^T. \tag{WD.30}$$

We solve the linear set of equations

$$\mathbf{b}^T \mathcal{C}_{\mathbf{x}} = \mathbf{e}_n^T \tag{WD.31}$$

and then compute

$$\mathbf{K} = \mathbf{b}^T \alpha_c(\mathbf{A}). \tag{WD.32}$$

Ackermann's formula, Eq. (WD.29), even though elegant, is not recommended for systems with a large number of state variables. Even if it is used, Eqs. (WD.31) and (WD.32) are recommended for better numerical accuracy.

Appendix W2.1.4
Complex Mechanical Systems

In some cases, mechanical systems contain both translational and rotational portions. The procedure is the same as that described in Section 2.1: sketch the free-body diagrams, define coordinates and positive directions, determine all forces and moments acting, and apply Eqs. (2.1) and/or (2.14).

EXAMPLE W2.1 *Rotational and Translational Motion: Hanging Crane*

Write the equations of motion for the hanging crane pictured in Fig. W2.1 and shown schematically in Fig. W2.2. Linearize the equations about

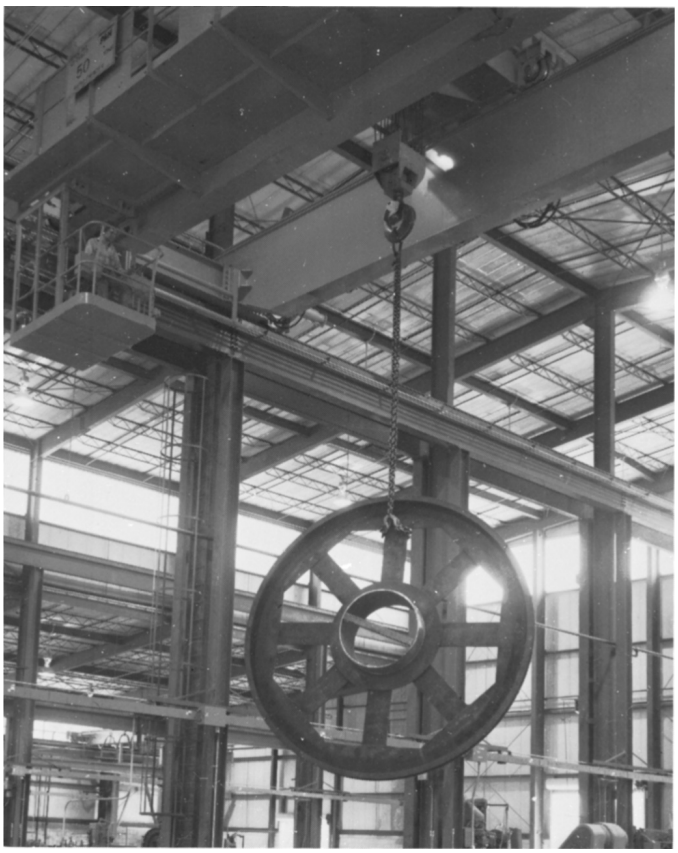


Figure W2.1
Crane with a hanging load
Source: Photo courtesy of Harnischfeger Corporation, Milwaukee, Wisconsin

$\theta = 0$, which would typically be valid for the hanging crane. Also linearize the equations for $\theta = \pi$, which represents the situation for the inverted pendulum shown in Fig. W2.3.

Solution. A schematic diagram of the hanging crane is shown in Fig. W2.2, while the free-body diagrams are shown in Fig. W2.4. In the case of the pendulum, the forces are shown with bold lines, while the components of the inertial acceleration of its center of mass are shown with dashed lines. Because the pivot point of the pendulum is *not* fixed with respect to an inertial reference, the rotation of the pendulum and the motion of its mass center must be considered. The inertial acceleration needs to be determined because the vector **a** in Eq. (2.1) is given with respect to an inertial reference. The inertial acceleration of the pendulum's mass center is the vector sum of the three dashed arrows shown in Fig. W2.4*b*. The derivation of the components of an object's acceleration is called **kinematics** and is usually studied as a prelude to the application of Newton's laws. The results of a kinematic study are shown in Fig. W2.4*b*. The component of acceleration along the pendulum is $l\dot{\theta}^2$ and is called the centripetal acceleration. It is present for any object whose velocity is changing direction. The $\ddot{x}$-component of acceleration is a consequence of the pendulum pivot point accelerating at the trolley's acceleration and will always have the same direction and magnitude as those of the trolley's. The $l\ddot{\theta}$ component is a result of angular acceleration of the pendulum and is always perpendicular to the pendulum.

These results can be confirmed by expressing the center of mass of the pendulum as a vector from an inertial reference and then differentiating that vector twice to obtain an inertial acceleration. Figure W2.4*c* shows $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$

Figure W2.2
Schematic of the crane with hanging load

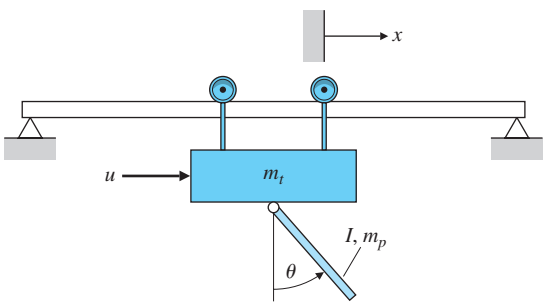
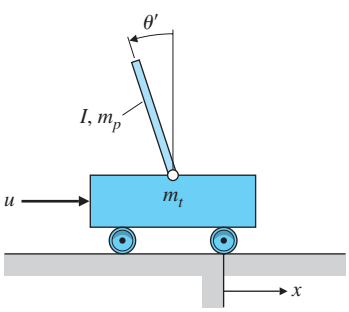


Figure W2.3
Inverted pendulum



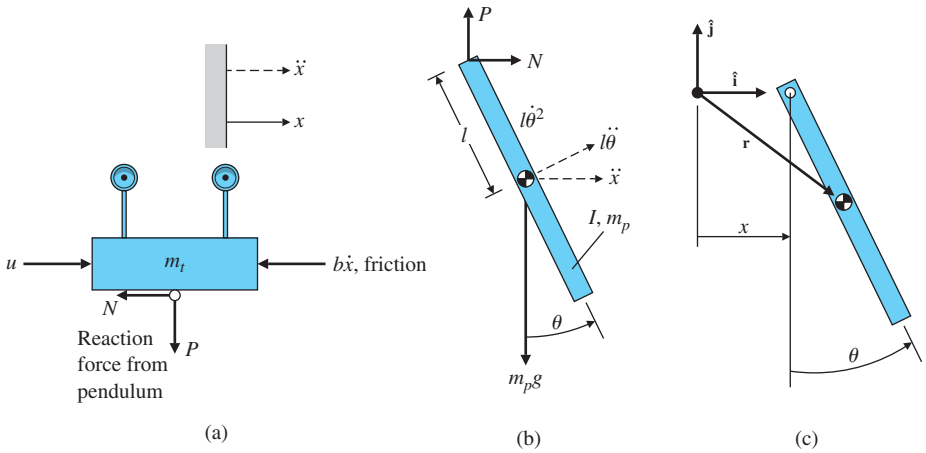


Figure W2.4

Hanging crane: (a) free-body diagram of the trolley; (b) free-body diagram of the pendulum; (c) position vector of the pendulum

axes that are inertially fixed and a vector $\mathbf{r}$ describing the position of the pendulum center of mass. The vector can be expressed as

$$\mathbf{r} = x\hat{\mathbf{i}} + l(\hat{\mathbf{i}}\sin\theta - \hat{\mathbf{j}}\cos\theta).$$

The first derivative of $\mathbf{r}$ is

$$\dot{\mathbf{r}} = \dot{x}\hat{\mathbf{i}} + l\dot{\theta}(\hat{\mathbf{i}}\cos\theta + \hat{\mathbf{j}}\sin\theta).$$

Likewise, the second derivative of $\mathbf{r}$ is

$$\ddot{\mathbf{r}} = \ddot{x}\hat{\mathbf{i}} + l\ddot{\theta}(\hat{\mathbf{i}}\cos\theta + \hat{\mathbf{j}}\sin\theta) - l\dot{\theta}^2(\hat{\mathbf{i}}\sin\theta - \hat{\mathbf{j}}\cos\theta).$$

Note that the equation for $\ddot{\mathbf{r}}$ confirms the acceleration components shown in Fig.W2.4*b*. The $l\dot{\theta}^2$ term is aligned along the pendulum pointing toward the axis of rotation, and the $l\ddot{\theta}$ term is aligned perpendicular to the pendulum pointing in the direction of a positive rotation.

Having all the forces and accelerations for the two bodies, we now proceed to apply Eq. (2.1). In the case of the trolley, Fig.W2.4*a*, we see that it is constrained by the tracks to move only in the x-direction; therefore, application of Eq. (2.1) in this direction yields

$$m_t\ddot{x} + b\dot{x} = u - N, \tag{W2.1}$$

where N is an unknown reaction force applied by the pendulum. Conceptually, Eq. (2.1) can be applied to the pendulum of Fig. W2.4*b* in the vertical and horizontal directions, and Eq. (2.14) can be applied for rotational motion to yield three equations in the three unknowns: N, P, and θ. These three equations then can be manipulated to eliminate the reaction forces N and P so that a single equation results describing the motion of the pendulum—that

is, a single equation in θ. For example, application of Eq. (2.1) for pendulum motion in the x-direction yields

$$N = m_p\ddot{x} + m_p l\ddot{\theta}\cos\theta - m_p l\dot{\theta}^2\sin\theta. \tag{W2.2}$$

However, considerable algebra will be avoided if Eq. (2.1) is applied perpendicular to the pendulum to yield

$$P\sin\theta + N\cos\theta - m_p g\sin\theta = m_p l\ddot{\theta} + m_p\ddot{x}\cos\theta. \tag{W2.3}$$

Application of Eq. (2.14) for the rotational pendulum motion, for which the moments are summed about the center of mass, yields

$$-Pl\sin\theta - Nl\cos\theta = I\ddot{\theta}, \tag{W2.4}$$

where I is the moment of inertia about the pendulum's mass center. The reaction forces N and P can now be eliminated by combining Eqs. (W2.3) and (W2.4). This yields the equation

$$(I + m_p l^2)\ddot{\theta} + m_p gl\sin\theta = -m_p l\ddot{x}\cos\theta. \tag{W2.5}$$

It is identical to a pendulum equation of motion, except that it contains a forcing function that is proportional to the trolley's acceleration.

An equation describing the trolley motion was found in Eq. (W2.1), but it contains the unknown reaction force N. By combining Eqs. (W2.2) and (W2.1), N can be eliminated to yield

$$(m_t + m_p)\ddot{x} + b\dot{x} + m_p l\ddot{\theta}\cos\theta - m_p l\dot{\theta}^2\sin\theta = u. \tag{W2.6}$$

Equations (W2.5) and (W2.6) are the nonlinear differential equations that describe the motion of the crane with its hanging load. For an accurate calculation of the motion of the system, these nonlinear equations need to be solved.

To linearize the equations for small motions about $\theta = 0$, let $\cos\theta \cong 1$, $\sin\theta \cong \theta$, and $\dot{\theta}^2 \cong 0$; thus the equations are approximated by

$$\begin{aligned}(I + m_p l^2)\ddot{\theta} + m_p gl\theta &= -m_p l\ddot{x},\\ (m_t + m_p)\ddot{x} + b\dot{x} + m_p l\ddot{\theta} &= u.\end{aligned} \tag{W2.7}$$

Neglecting the friction term b leads to the transfer function from the control input u to the hanging crane angle θ:

$$\frac{\theta(s)}{U(s)} = \frac{-m_p l}{((I + m_p l^2)(m_t + m_p) - m_p^2 l^2)s^2 + m_p gl(m_t + m_p)}. \tag{W2.8}$$

Inverted pendulum equations

For the inverted pendulum in Fig. W2.3, where $\theta \cong \pi$, assume $\theta = \pi + \theta'$, where θ' represents motion from the vertical *upward* direction. In this case, $\cos\theta \cong -1$, $\sin\theta \cong -\theta'$ in Eqs. (W2.5) and (W2.6), and Eqs. (W2.7) become[1]

$$(I + m_p l^2)\ddot{\theta}' - m_p g l\theta' = m_p l\ddot{x},$$
$$(m_t + m_p)\ddot{x} + b\dot{x} - m_p l\ddot{\theta}' = u. \tag{W2.9}$$

As noted in Example 2.2, a stable system will always have the same signs on each variable, which is the case for the stable hanging crane modeled by Eqs. (W2.7). However, the signs on θ and $\ddot{\theta}$ in the top Eq. (W2.9) are opposite, thus indicating instability, which is the characteristic of the inverted pendulum.

The transfer function, again without friction, is

$$\frac{\theta'(s)}{U(s)} = \frac{m_p l}{((I + m_p l^2) - m_p^2 l^2)s^2 - m_p g l(m_t + m_p)}. \tag{W2.10}$$

W2.1 Additional Problems for Translational and Rotational Systems

Assume the driving force on the hanging crane of Fig. W2.2 is provided by a motor mounted on the cab with one of the support wheels connected directly to the motor's armature shaft. The motor constants are K_e and K_t, and the circuit driving the motor has a resistance R_a and negligible inductance. The wheel has a radius r. Write the equations of motion relating the applied motor voltage to the cab position and load angle.

Solution. The dynamics of the hanging crane are given by Eqs. (W2.5) and (W2.6),

$$\left(I + m_p l^2\right)\ddot{\theta} + m_p g l \sin\theta = -m_p l\ddot{x}\cos\theta,$$
$$\left(m_t + m_p\right)\ddot{x} + b\dot{x} + m_p l\ddot{\theta}\cos\theta - m_p l\dot{\theta}^2\sin\theta = u,$$

where x is the position of the cab, θ is the angle of the load, and u is the applied force that will be produced by the motor. Our task here is to find the force applied by the motor. Normally, the rotational dynamics of a motor is

$$J_1\ddot{\theta}_m + b_1\dot{\theta}_m = T_m = K_t i_a,$$

where the current is found from the motor circuit, which reduces to

$$R_a i_a = V_a - K_e\dot{\theta}_m$$

for the case where the inductance is negligible. However, since the motor is geared directly to the cab, θ_m and x are related kinematically by

$$x = r\theta_m$$

[1] The inverted pendulum is often described with the angle of the pendulum being positive for *clockwise* motion. If defined that way, then reverse the sign on all terms in Eq. (W2.9) in θ' or $\ddot{\theta}'$.

and we can neglect any extra inertia or damping from the motor itself compared to the inertia and damping of the large cab. Therefore we can rewrite the two motor equations in terms of the force applied by the motor on the cab

$$u = T_m/r = K_t i_a/r,$$
$$i_a = (V_a - K_e\dot{\theta}_m)/R_a,$$

where

$$\dot{\theta}_m = \dot{x}/r.$$

These equations, along with

$$\left(I + m_p l^2\right)\ddot{\theta} + m_p g l \sin\theta = -m_p l\ddot{x}\cos\theta,$$
$$\left(m_t + m_p\right)\ddot{x} + b\dot{x} + m_p l\ddot{\theta}\cos\theta - m_p l\dot{\theta}^2\sin\theta = u,$$

constitute the required relations.

Appendix W3.2.3

Δ W3.2.3 Mason's Rule and the Signal-Flow Graph

Signal-flow graph

A compact alternative notation to the block diagram is given by the **signal-flow graph** introduced by S. J. Mason (1953, 1956). As with the block diagram, the signal-flow graph offers a visual tool for representing the causal relationships between the components of the system. The method consists of characterizing the system by a network of directed branches and associated gains (transfer functions) connected at nodes. Several block diagrams and their corresponding signal-flow graphs are shown in Fig. W3.1. The two ways of depicting a system are equivalent, and you can use either diagram to apply Mason's rule (to be defined shortly).

In a signal-flow graph, the internal signals in the diagram, such as the common input to several blocks or the output of a summing junction, are called **nodes**. The system input point and the system output point are also nodes; the input node has outgoing branches only, and the output node has incoming branches only. Mason defined a **path** through a block diagram as a sequence of connected blocks, the route passing from one node to another *in the direction of signal flow of the blocks* without including any block more than once. A **forward path** is a path from the input to output such that no node is included more than once. If the nodes are numbered in a convenient order, a forward path can be identified by the numbers that are included. Any closed path that returns to its starting node without passing through any node more than once is a **loop**, and a path that leads from a given variable back to the same variable is a **loop path**. The **path gain** is the product of component gains (transfer functions) making up the path. Similarly, the **loop gain** is the path gain associated with a loop—that is, the product of gains in a loop. If two paths have a common component, they are said to touch. Notice particularly in this connection that the input and the output of a summing junction are not the same and that the summing junction is a one-way device from its inputs to its output.

Mason's rule relates the graph to the algebra of the simultaneous equations it represents.[1] Consider Fig. W3.1c, where the signal at each node has been given a name and the gains are marked. Then the block diagram (or the signal-flow graph) represents the system of equations:

$$X_1(s) = X_3(s) + U(s),$$
$$X_2(s) = G_1(s)X_1(s) + G_2(s)X_2(s) + G_4(s)X_3(s),$$
$$Y(s) = 1X_3(s).$$

[1]The derivation is based on Cramer's rule for solving linear equations by determinants and is described in Mason's papers.

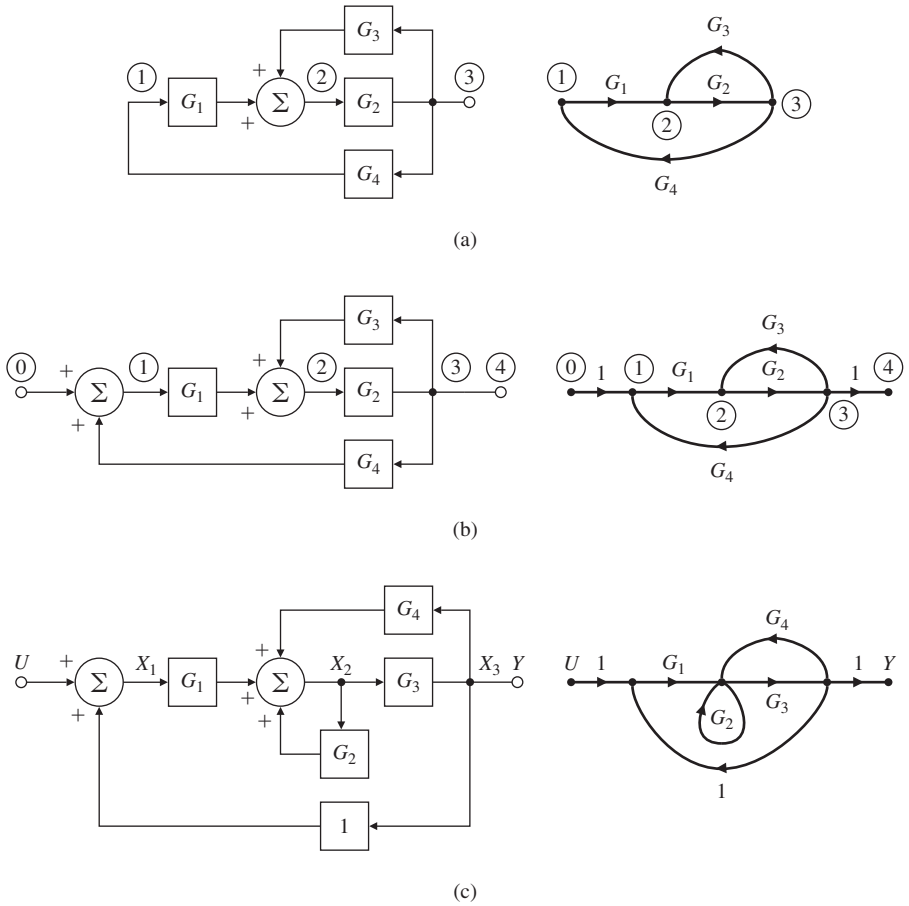


Figure W3.1
Block diagrams and corresponding signal-flow graphs

Mason's rule

Mason's rule states that the input–output transfer function associated with a signal-flow graph is given by

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{\Delta}\sum_i G_i \Delta_i,$$

where

G_i = path gain of the ith forward path,

Δ = the system determinant

= 1 − ∑ (all individual loop gains) + ∑ (gain products of all possible two loops that do not touch) − ∑ (gain products of all possible three loops that do not touch) + . . . ,

Δ_i = ith forward path determinant

= value of Δ for that part of the block diagram that does *not* touch the ith forward path.

We will now illustrate the use of Mason's rule using some examples.

EXAMPLE W3.1 *Mason's Rule in a Simple System*

Find the transfer function for the block diagram in Fig. W3.2.

Solution. From the block diagram shown in Fig. W3.2, we have

Forward Path	*Path Gain*
1236	$G_1 = 1\left(\frac{1}{s}\right)(b_1)(1)$
12346	$G_2 = 1\left(\frac{1}{s}\right)\left(\frac{1}{s}\right)(b_2)(1)$
123456	$G_3 = 1\left(\frac{1}{s}\right)\left(\frac{1}{s}\right)\left(\frac{1}{s}\right)(b_3)(1)$

	Loop Path Gain
232	$l_1 = -a_1/s$
2342	$l_2 = -a_2/s^2$
23452	$l_3 = -a_3/s^3$

and the determinants are

$$\Delta = 1 - \left(-\frac{a_1}{s} - \frac{a_2}{s^2} - \frac{a_3}{s^3}\right) + 0,$$
$$\Delta_1 = 1 - 0,$$
$$\Delta_2 = 1 - 0,$$
$$\Delta_3 = 1 - 0.$$

Applying Mason's rule, we find the transfer function to be

$$G(s) = \frac{Y(s)}{U(s)} = \frac{(b_1/s) + (b_2/s^2) + (b_3/s^3)}{1 + (a_1/s) + (a_2/s^2) + (a_3/s^3)}$$

$$= \frac{b_1 s^2 + b_2 s + b_3}{s^3 + a_1 s^2 + a_2 s + a_3}.$$

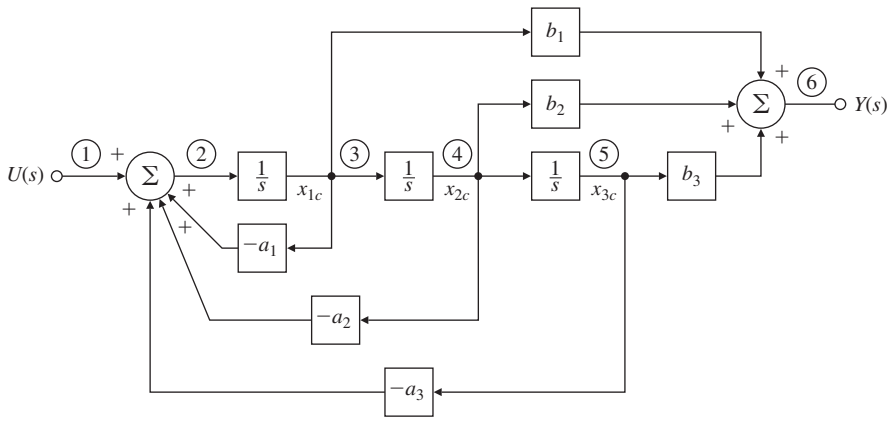


Figure W3.2
Block diagram for Example W3.1

Mason's rule is particularly useful for more complex systems where there are several loops, some of which do not sum into the same point.

EXAMPLE W3.2 *Mason's Rule in a Complex System*

Find the transfer function for the system shown in Fig. W3.3.

Solution. From the block diagram, we find that

Forward Path	*Path Gain*
12456	$G_1 = H_1H_2H_3$
1236	$G_2 = H_4$
	Loop Path Gain
242	$l_1 = H_1H_5$ (does not touch l_3)
454	$l_2 = H_2H_6$
565	$l_3 = H_3H_7$ (does not touch l_1)
236542	$l_4 = H_4H_7H_6H_5$

and the determinants are

$$\Delta = 1 - (H_1H_5 + H_2H_6 + H_3H_7 + H_4H_7H_6H_5) + (H_1H_5H_3H_7),$$

$$\Delta_1 = 1 - 0,$$

$$\Delta_2 = 1 - H_2H_6.$$

Therefore,

$$G(s) = \frac{Y(s)}{U(s)} = \frac{H_1H_2H_3 + H_4 - H_4H_2H_6}{1 - H_1H_5 - H_2H_6 - H_3H_7 - H_4H_7H_6H_5 + H_1H_5H_3H_7}.$$

Mason's rule is useful for solving relatively complicated block diagrams by hand. It yields the solution in the sense that it provides an explicit input–output relationship for the system represented by the diagram. The advantage

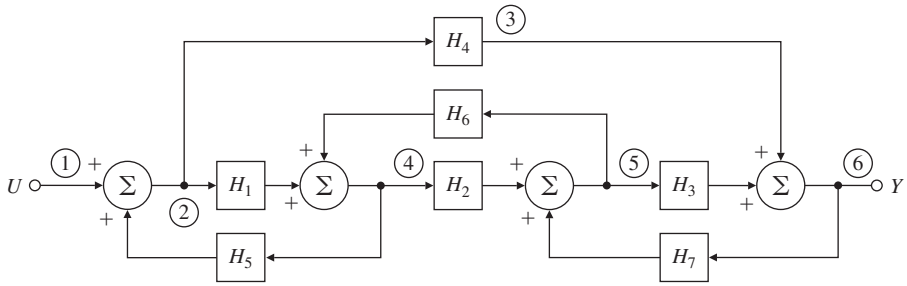


Figure W3.3
Block diagram for Example W3.2

compared with path-by-path block-diagram reduction is that it is systematic and algorithmic rather than problem dependent. MATLAB and other control systems computer-aided software allow you to specify a system in terms of individual blocks in an overall system, and the software algorithms perform the required block-diagram reduction; therefore, Mason's rule is less important today than in the past. However, there are some derivations that rely on the concepts embodied by the rule, so it still has a role in the control designer's toolbox.

△Appendix W.3.6.3.1
Routh Special Cases

Special case I

If only the first element in one of the rows is zero, then we can consider a modified equation with one of the coefficients perturbed by $\epsilon > 0$ and applying the test by taking the limit as $\epsilon \to 0$.

EXAMPLE W3.3 *Routh's Test for Special Case I*

Consider the polynomial

$$a(s) = s^5 + 3s^4 + 2s^3 + 6s^2 + 6s + 9.$$

Determine whether any of the roots are in the RHP.

Solution. In this example, let the coefficient of s^3 be $2+\epsilon$. The test follows from there. The Routh array is

s^5:	1	2	6
s^4:	3	6	9
s^3:	ϵ	3	0
s^2:	$\frac{6\epsilon-9}{\epsilon}$	9	0
s:	$3-\frac{3\epsilon^2}{2\epsilon-3}$	0	0
s^0:	9	0	

There are two sign changes in the first column of the array, which means there are two poles not in the LHP.[1]

Special case II

Another special[2] case occurs when an entire row of the Routh array is zero. This indicates that there are complex conjugate pairs of roots that are mirror images of each other with respect to the imaginary axis. To apply Routh's test correctly, we follow the ensuing procedure. If the ith row is zero, we form an auxiliary equation from the previous (nonzero) row:

$$a_1(s) = \beta_1 s^{i+1} + \beta_2 s^{i-1} + \beta_3 s^{i-3} + \cdots . \tag{W3.1}$$

Here $\{\beta_i\}$ are the coefficients of the $(i+1)$th row in the array. We then replace the ith row by the coefficients of the *derivative* of the auxiliary polynomial and complete the array. However, the roots of the auxiliary polynomial in Eq. (W3.1) are also roots of the characteristic equation, and these must be tested separately.

[1] The actual roots computed with Matlab are at -2.9043, $0.6567 \pm 1.2881j$, $-0.7046 \pm 0.9929j$.

[2] Special case II.

EXAMPLE W3.4 *Routh Test for Special Case II*

For the polynomial

$$a(s) = s^5 + 5s^4 + 11s^3 + 23s^2 + 28s + 12,$$

determine whether there are any roots on the $j\omega$ axis or in the RHP.

Solution. The Routh array is

s^5 :	1	11	28	
s^4 :	5	23	12	
s^3 :	6.4	25.6	0	
s^2 :	3	12		
s :	0	0		$\leftarrow a_1(s) = 3s^2 + 12$
New s :	6	0		$\leftarrow \frac{da_1(s)}{ds} = 6s$
s^0 :	12.			

There are no sign changes in the first column. Hence all the roots have negative real parts except for a pair on the imaginary axis. We may deduce this as follows: When we replace the zero in the first column by $\epsilon > 0$, there are no sign changes. If we let $\epsilon < 0$, then there are two sign changes. Thus, if $\epsilon = 0$, there are two poles on the imaginary axis, which are the roots of

$$a_1(s) = 3s^2 + 12 = 0,$$

or

$$s = \pm j2.$$

This agrees with the fact that the actual roots are at -3, $\pm 2j$, -1, and -1, as computed using the roots command in Matlab.

Appendix W3.7
System Identification

W3.7.1 A Perspective on System Identification

In order to design controls for a dynamic system, it is necessary to have a model that will adequately describe the system's dynamics. The information available to the designer for this purpose is typically of three kinds.

1. **Physical model**: First, there is the knowledge of physics, chemistry, biology, and the other sciences which have over the years developed equations of motion to explain the dynamic response of rigid and flexible bodies, electric circuits and motors, fluids, chemical reactions, and many other constituents of systems to be controlled. The model based on this knowledge is referred to as a "physical" model. There are many advantages to this approach, including ease of controller development and testing. One disadvantage of this approach is that a fairly high-fidelity physical model must be developed.

2. **Black box model**: It is often the case that for extremely complex physical phenomena the laws of science are not adequate to give a satisfactory description of the dynamic plant that we wish to control. Examples include the force on a moving airplane caused by a control surface mounted on a wing and the heat of combustion of a fossil fuel of uncertain composition. In these circumstances, the designer turns to data taken from experiments directly conducted to excite the plant and measure its response. The second approach uses an empirical or heuristic model referred to as the "black box" model. In this approach, the control engineer injects open-loop commands into the system and records the sensor response. The process of constructing models from experimental data is called **system identification**. Standard system identification techniques (for example, linear least-squares) are used to identify a dynamic input/output model. The advantage of this technique is that the control engineer does not need to have a deep understanding of how the system physically behaves, but instead can design a controller solely based on the derived model. There are several major disadvantages to this approach. First, the control engineer must have access to working hardware. Another serious disadvantage of this approach is that it does not provide insight or physical understanding of how specific hardware modifications will affect the control—usually hardware modifications require the control engineer to repeat the full cycle of system identification, control design, and validation. The advantage of this approach is that we use logic and data to model inputs and outputs and the detailed knowledge of the physics is not required.

3. **Grey box model**: The third approach is the use of the combination of physical and empirical models referred to as "grey box" modeling.

In identifying models for control, our motivation is very different from that of modeling as practiced in the sciences. In science, one seeks to develop models of nature as it is; in control, one seeks to develop models of the plant dynamics that will be adequate for the design of a controller that will cause the actual dynamics to be stable and to give good performance. The initial design of a control system typically considers a small signal analysis and is based on models that are linear and time-invariant (LTI). This is referred to as a "control relevant" model. Having accepted that the model is to be linear, we still must choose between several alternative descriptions of linear systems. If we examine the design methods described in the earlier chapters, we find that the required plant models may be grouped in two categories: parametric and nonparametric. For design via root locus or pole assignment, we require a parametric description such as a transfer function or a state-variable description from which we can obtain the poles and zeros of the plant. These equivalent models are completely described by the numbers that specify the coefficients of the polynomials, the elements of the state-description matrices, or the numbers that specify the poles and zeros. In either case, we call these numbers the *parameters* of the model, and the category of such models is a **parametric description** of the plant model.

Parametric model

In contrast to parametric models, the frequency-response methods of Nyquist, Bode, and Nichols require the curves of amplitude and phase of the transfer function $G(j\omega) = Y(j\omega)/U(j\omega)$ as functions of ω. Clearly, if we happen to have a parametric description of the system, we can compute the transfer function and the corresponding frequency response. However, if we are given the frequency response or its inverse transform, the impulse response, without parameters (perhaps obtained from experimental data), we have all we need to design a lead, lag, notch, or other compensation to achieve a desired bandwidth, phase margin, or other frequency response performance objective without ever knowing what the parameters are. We call the functional curves of $G(j\omega)$ a **nonparametric** model because in principle there is no finite set of numbers that describes it exactly.

Nonparametric model

Because of the large data records necessary to obtain effective models and the complexity of many of the algorithms used, the use of computer aids is essential in identification. Developments such as Matlab's System Identification Toolbox are enormous aids to the practical use of the system identification techniques. For detailed discussion on system identification, the reader is referred to Franklin, Powell, and Workman (1998).

W3.7.2 Obtaining Models from Experimental Data

There are several reasons for using experimental data to obtain a model of the dynamic system to be controlled. In the first place, the best theoretical model built from equations of motion is still only an approximation of reality. Sometimes, as in the case of a very rigid spacecraft, the theoretical model is extremely good. Other times, as with many chemical processes such as papermaking or metalworking, the theoretical model is very approximate. In every case, before the final control design is done, it is important and

prudent to verify the theoretical model with experimental data. Second, in situations for which the theoretical model is especially complicated or the physics of the process is poorly understood, the only reliable information on which to base the control design is the experimental data. Finally, the system is sometimes subject to online changes that occur when the environment of the system changes. Examples include when an aircraft changes altitude or speed, a paper machine is given a different composition of fiber, or a nonlinear system moves to a new operating point. On these occasions, we need to "retune" the controller by changing the control parameters. This requires a model for the new conditions, and experimental data are often the most effective, if not the only, information available for the new model.

Our sources of experimental data

There are four kinds of experimental data for generating a model:

1. **Transient response**, such as comes from an impulse or a step;
2. **Frequency-response data**, which result from exciting the system with sinusoidal inputs at many frequencies;
3. **Stochastic steady-state information**, as might come from flying an aircraft through turbulent weather or from some other natural source of randomness; and
4. **Pseudorandom-noise data**, as may be generated in a digital computer.

Each class of experimental data has its properties, advantages, and disadvantages.

Transient response

Transient-response data are quick and relatively easy to obtain. They are also often representative of the natural signals to which the system is subjected. Thus a model derived from such data can be reliable for designing the control system. On the other hand, in order for the signal-to-noise ratio to be sufficiently high, the transient response must be highly noticeable. Consequently, the method is rarely suitable for normal operations, so the data must be collected as part of special tests. A second disadvantage is that the data do not come in a form suitable for standard control systems designs, and some parts of the model, such as poles and zeros, must be computed from the data.[1] This computation can be simple in special cases or complex in the general case.

Frequency response

Frequency-response data (see Chapter 6) are simple to obtain but substantially more time consuming than transient-response information. This is especially so if the time constants of the process are large, as often occurs in chemical processing industries. As with the transient-response data, it is important to have a good signal-to-noise ratio, so obtaining frequency-response data can be very expensive. On the other hand, as we will see in Chapter 6, frequency-response data are exactly in the right form for frequency-response design methods; so once the data have been obtained, the control design can proceed immediately.

[1] Ziegler and Nichols (1943), building on the earlier work of Callender et al. (1936), use the step response directly in designing the controls for certain classes of processes. See Chapter 4 for details.

Stochastic steady-state

Normal operating records from a natural stochastic environment at first appear to be an attractive basis for modeling systems, since such records are by definition nondisruptive and inexpensive to obtain. Unfortunately, the quality of such data is inconsistent, tending to be worse just when the control is best, because then the upsets are minimal and the signals are smooth. At such times, some or even most of the system dynamics are hardly excited. Because they contribute little to the system output, they will not be found in the model constructed to explain the signals. The result is a model that represents only part of the system and is sometimes unsuitable for control. In some instances, as occurs when trying to model the dynamics of the electroencephalogram (brain waves) of a sleeping or anesthetized person to locate the frequency and intensity of alpha waves, normal records are the only possibility. Usually they are the last choice for control purposes.

Pseudorandom noise (PRBS)

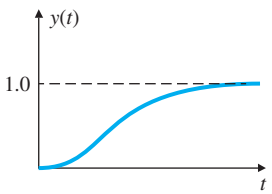
Finally, the pseudorandom signals that can be constructed using digital logic have much appeal. Especially interesting for model making is the pseudorandom binary signal (PRBS). The PRBS takes on the value $+A$ or $-A$ according to the output (1 or 0) of a feedback shift register. The feedback to the register is a binary sum of various states of the register that have been selected to make the output period (which must repeat itself in finite time) as long as possible. For example, with a register of 20 bits, $2^{20} - 1$ (over a million) steps are produced before the pattern repeats. Analysis beyond the scope of this text has revealed that the resulting signal is almost like a broadband random signal. Yet this signal is entirely under the control of the engineer who can set the level (A) and the length (bits in the register) of the signal. The data obtained from tests with a PRBS must be analyzed by computer and both special-purpose hardware and programs for general-purpose computers have been developed to perform this analysis.

W3.7.3 Models from Transient-Response Data

To obtain a model from transient data, we assume that a step response is available. If the transient is a simple combination of elementary transients, then a reasonable low-order model can be estimated using hand calculations. For example, consider the step response shown in Fig. W3.1. The response is monotonic and smooth. If we assume that it is given by a sum of exponentials, we can write

$$y(t) = y(\infty) + Ae^{-\alpha t} + Be^{-\beta t} + Ce^{-\gamma t} + \cdots . \tag{W3.2}$$

Figure W3.4
A step response characteristic of many chemical processes



Subtracting off the final value and assuming that $-\alpha$ is the slowest pole, we write

$$\begin{aligned} y - y(\infty) &\cong Ae^{-\alpha t}, \\ \log_{10}[y - y(\infty)] &\cong \log_{10} A - \alpha t \log_{10} e, \\ &\cong \log_{10} A - 0.4343\alpha t. \end{aligned} \tag{W3.3}$$

This is the equation of a line whose slope determines α and intercept determines A. If we fit a line to the plot of $\log_{10}[y - y(\infty)]$ (or $\log_{10}[y(\infty) - y]$ if A is negative), then we can estimate A and α. Once these are estimated, we plot $y - [y(\infty) + Ae^{-\alpha t}]$, which as a curve approximates $Be^{-\beta t}$ and on the log plot is equivalent to $\log_{10} B - 0.4345\beta t$. We repeat the process, each time removing the slowest remaining term, until the data stop is accurate. Then we plot the final model step response and compare it with data so we can assess the quality of the computed model. It is possible to get a good fit to the step response and yet be far off from the true time constants (poles) of the system. However, the method gives a good approximation for control of processes whose step responses look like Fig. W3.4.

EXAMPLE W3.5 *Determining the Model from Time-Response Data*

Find the transfer function that generates the data given in Table W3.1 and plotted in Fig. W3.5.

Solution. Table W3.1 shows and Fig. W3.5 implies that the final value of the data is $y(\infty) = 1$. We know that A is negative because $y(\infty)$ is greater than $y(t)$. Therefore, the first step in the process is to plot $\log_{10}[y(\infty) - y]$, which is shown in Fig. W3.6. From the line (fitted by eye), the values are

$$\log_{10} |A| = 0.125,$$

$$0.4343\alpha = \frac{1.602 - 1.167}{\Delta t} = \frac{0.435}{1} \Rightarrow \alpha \cong 1.$$

Thus

$$A = -1.33,$$

$$\alpha = 1.0.$$

TABLE W3.1

Step Response Data

t	*y(t)*	*t*	*y(t)*
0.1	0.000	1.0	0.510
0.1	0.005	1.5	0.700
0.2	0.034	2.0	0.817
0.3	0.085	2.5	0.890
0.4	0.140	3.0	0.932
0.5	0.215	4.0	0.975
		∞	1.000

Sinha and Kuszta (1983).

Figure W3.5
Step response data in Table W3.1

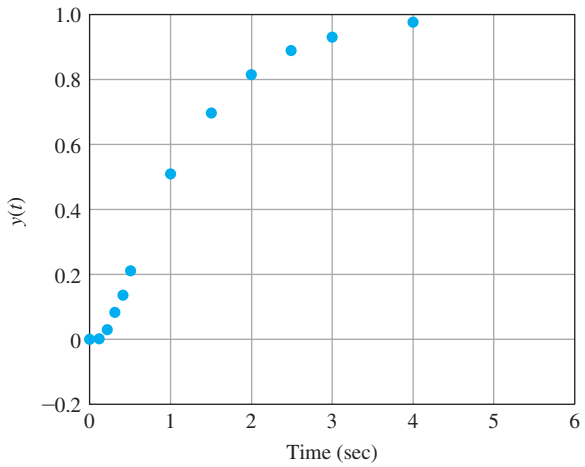
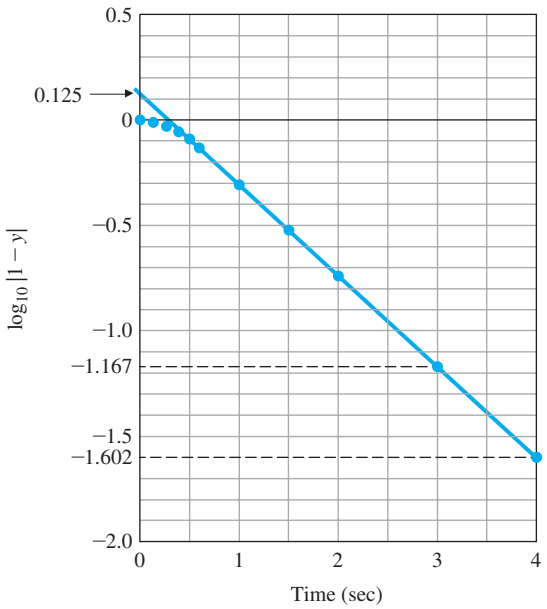


Figure W3.6
$\log_{10}[y(\infty) - y]$ versus t



If we now subtract $1 + Ae^{\alpha t}$ from the data and plot the log of the result, we find the plot of Fig. W3.7. Here we estimate

$$\log_{10} B = -0.48,$$

$$0.4343\beta = \frac{-0.48 - (-1.7)}{0.5} = 2.5,$$

Figure W3.7
$\log_{10}[y - (1 + Ae^{-\alpha t})]$ versus t

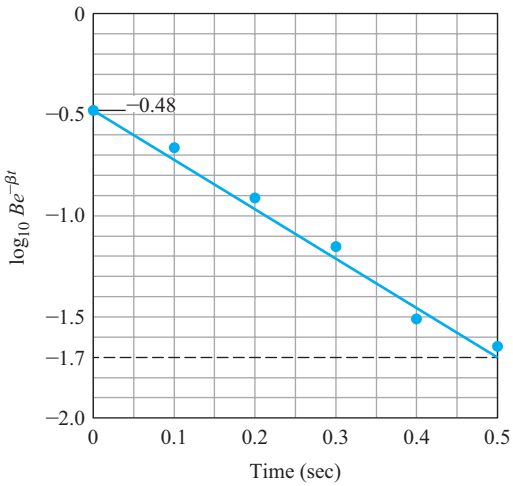
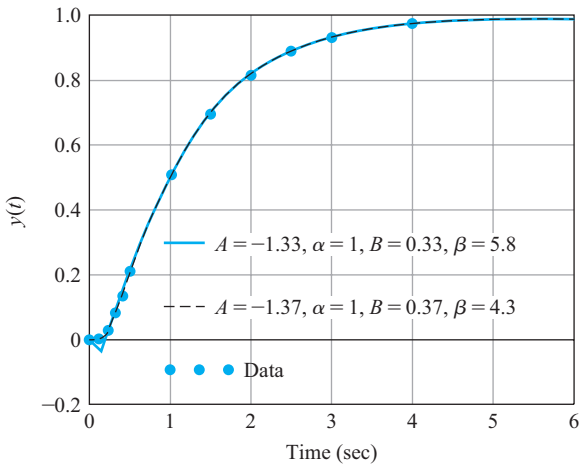


Figure W3.8
Model fits to the experimental data



$$\beta \cong 5.8,$$

$$B = 0.33.$$

Combining these results, we arrive at the y estimate

$$\hat{y}(t) \cong 1 - 1.33e^{-t} + 0.33e^{-5.8t}. \tag{W3.4}$$

Equation (W3.4) is plotted as the colored line in Fig. W3.8 and shows a reasonable fit to the data, although some error is noticeable near $t = 0$.

From $\hat{y}(t)$, we compute

$$\begin{aligned}\hat{Y}(s) &= \frac{1}{s} - \frac{1.33}{s+1} + \frac{0.33}{s+5.8} \\ &= \frac{(s+1)(s+5.8) - 1.33s(s+5.8) + 0.33s(s+1)}{s(s+1)(s+5.8)} \\ &= \frac{-0.58s + 5.8}{s(s+1)(s+5.8)}.\end{aligned}$$

The resulting transfer function is

$$G(s) = \frac{-0.58(s-10)}{(s+1)(s+5.8)}.$$

Notice that this method has given us a system with a zero in the RHP, even though the data showed no values of y that were negative. Very small differences in the estimated value for A, all of which approximately fit the data, can cause values of β to range from 4 to 6. This illustrates the sensitivity of pole locations to the quality of the data and emphasizes the need for a good signal-to-noise ratio.

By using a computer to perform the plotting, we are better able to iterate the four parameters to achieve the best overall fit. The data presentation in Figs. W3.6 and W3.7 can be obtained directly by using a semilog plot. This eliminates having to calculate $\log_{10}$ and the exponential expression to find the values of the parameters. The equations of the lines to be fit to the data are $y(t) = Ae^{\alpha t}$ and $y(t) = Be^{\beta t}$, which are straight lines on a semilog plot. The parameters A and α, or B and β, are iteratively selected so that the straight line comes as close as possible to passing through the data. This process produces the improved fit shown by the dashed black line in Fig. W3.8. The revised parameters, $A = -1.37$, $B = 0.37$, and $\beta = 4.3$ result in the transfer function

$$G(s) = \frac{-0.22s + 4.3}{(s+1)(s+4.3)}.$$

The RHP zero is still present, but it is now located at $s \cong +20$ and has no noticeable effect on the time response.

This set of data was fitted quite well by a second-order model. In many cases, a higher-order model is required to explain the data and the modes may not be as well separated.

If the transient response has oscillatory modes, then these can sometimes be estimated by comparing them with the standard plots of Fig. 3.18. The period will give the frequency ω_d, and the decay from one period to the next will afford an estimate of the damping ratio. If the response has a mixture of modes not well separated in frequency, then more sophisticated methods need to be used. One such is **least-squares system identification**, in which a numerical optimization routine selects the best combination of system parameters so as to minimize the fit error. The fit error is defined to be a scalar **cost function**

Least-squares system identification

$$J = \sum_i (y_{data} - y_{model})^2, \; i = 1, 2, 3, \cdots, \text{ for each data point,}$$

so that fit errors at all data points are taken into account in determining the best value for the system parameters.

W3.7.3.1 Models from Other Data

As mentioned early in Section 3.1.2, we can also generate a model using frequency-response data, which are obtained by exciting the system with a set of sinusoids and plotting $H(j\omega)$. In Chapter 6, we show how such plots can be used directly for design. Alternatively, we can use the frequency response to estimate the poles and zeros of a transfer function using straight-line asymptotes on a logarithmic plot.

The construction of dynamic models from normal stochastic operating records or from the response to a PRBS can be based either on the concept of cross-correlation or on the least-squares fit of a discrete equivalent model, both topics in the field of **system identification**. They require substantial presentation and background that are beyond the scope of this text. An introduction to system identification can be found in Chapter 8 of Franklin et al. (1998), and a comprehensive treatment is given in Ljüng (1999). Based largely on the work of Professor Ljüng, the MATLAB Toolbox on Identification provides substantial software to perform system identification and to verify the quality of the proposed models.

W3.7.4 Obtaining a Pole-Zero Model from Frequency-Response Data

As we pointed out earlier, it is relatively easy to obtain the frequency-response of a system experimentally. Sometimes it is desirable to obtain an approximate model, in terms of a transfer function, directly from the frequency response. The derivation of such a model can be done to various degrees of accuracy. The method described in this section is usually adequate and is widely used in practice.

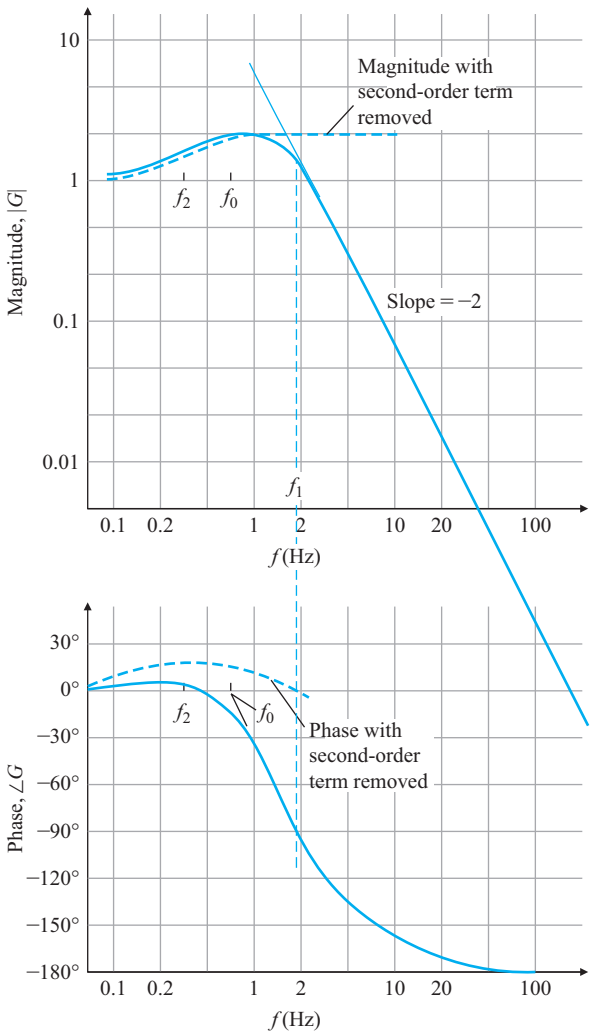
There are two ways to obtain a model from frequency-response data. In the first case, we can introduce a sinusoidal input, measure the gain (logarithm of the amplitude ratio of output to input) and the phase difference between output and input, and accept the curves plotted from this data as the model. Using the methods given in previous sections, we can derive the design directly from this information. In the second case, we wish to use the frequency data to verify a mathematical model obtained by other means. To do so we need to extract an approximate transfer function from the plots, again by fitting straight lines to the data, estimating break points (that is, finding the poles and zeros), and using Fig. 6.3 to estimate the damping ratios of complex factors from the frequency overshoot. The next example illustrates the second case.

EXAMPLE W3.6 *Transfer Function from Measured Frequency Response*

Determine a transfer function from the frequency response plotted in Fig. W3.9, where frequency f is plotted in hertz.

Solution. Drawing an asymptote to the final slope of -2 (or -40 db per decade), we assume a break point at the frequency where the phase is $-90°$. This occurs at $f_1 \cong 1.66$ Hz ($\omega_1 = 2\pi f_1 = 10.4$ rad/sec). We need to know

Figure W3.9
Experimental frequency response



the damping ratio in order to subtract out this second-order pole. For this, the phase curve may be of more help. Since the phase around the break-point frequency is symmetric, we draw a line at the slope of the phase curve at f_1 to find that the phase asymptote intersects the 0° line at $f_0 \cong 0.71$ Hz (or 4.46 rad/sec). This corresponds to $f_1/f_0 \cong 2.34$, which in time corresponds to $\zeta \cong 0.5$, as seen on the normalized response curves in Fig. 6.3b. The magnitude curve with the second-order factor taken out shows an asymptotic amplitude gain of about 6.0 db, or a factor of $10^{6.0/20} = 2.0$. As this is a gain rise, it occurs because of a lead compensation of the form

$$\frac{s/a + 1}{s/b + 1},$$

where $b/a = 2.0$. If we remove the second-order terms in the phase curve, we obtain a phase curve with a maximum phase of about 20°, which also

corresponds to a frequency separation of about 2. To locate the center of the lead compensation, we must estimate the point of maximum phase based on the lead term alone, which occurs at the geometric mean of the two break-point frequencies. The lead center seems to occur at $f_2 \cong 0.3$ Hz (or $\omega_2 = 1.88$ rad/sec).

Thus we have the relations

$$ab(1.88)^2 = 3.55,$$

$$\frac{b}{a} = 2,$$

from which we can solve

$$2a^2 = 3.55,$$

$$a = 1.33,$$

$$b = 2.66.$$

Model from measured response

Our final model is given by

$$\hat{G}(s) = \frac{(s/1.33) + 1}{[(s/2.66) + 1][(s/10.4)^2 + (s/10.4) + 1]}. \tag{W3.5}$$

The actual data were plotted from

$$G(s) = \frac{(s/2) + 1}{[(s/4) + 1][(s/10)^2 + (s/10) + 1]}.$$

As can be seen, we found the second-order term quite easily, but the location of the lead compensation is off in center frequency by a factor of $4/2.66 \cong 1.5$. However, the subtraction of the second-order term from the composite curve was not done with great accuracy, rather, by reading the curves. Again, as with the transient response, we conclude that by a bit of approximate plotting we can obtain a crude model (usually within a factor of 1.4 (± 3 db) in amplitude and $\pm 10°$ in phase) that can be used for control design.

Refinements on these techniques with computer aids are rather obvious, and an interactive program for removing standard first- and second-order terms and accurately plotting the residual function would greatly improve the speed and accuracy of the process. It is also common to have computer tools that can find the parameters of an assumed model structure by minimizing the sum of squares of the difference between the model's frequency response and the experimental frequency response.

Further Reading for System Identification:

[1] L. Ljung, *Perspectives on System Identification,* http://users.isy.liu.se/en/rt/ljung/seoul2dvinew/plenary2.pdf.

[2] L. Ljung, *System Identification: Theory for the User*, 2nd Ed., Prentice-Hall, 1999.

[3] G. F. Franklin, J. D. Powell, M. L. Workman, *Digital Control of Dynamic Systems*, 3rd Ed. Ellis-Kagle Press, 1998.

[4] M. B. Tischler and R. K. Remple, *Aircraft and Rotorcraft System Identification: Engineering Methods with Flight-Test Examples,* AIAA, 2006.

[5] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*, 2nd ed., Wiley-IEEE Press, 2012.

[6] System Identification Toolbox, The Mathworks.

Appendix W3.8
Amplitude and Time Scaling

The magnitude of the values of the variables in a problem is often very different, sometimes so much so that numerical difficulties arise. This was a serious problem years ago when equations were solved using analog computers and it was routine to *scale* the variables so that all had similar magnitudes. Today's widespread use of digital computers for solving differential equations has largely eliminated the need to scale a problem unless the number of variables is very large, because computers are now capable of accurately handling numbers with wide variations in magnitude. Nevertheless, it is wise to understand the principle of scaling for the few cases in which extreme variations in magnitude exist and scaling is necessary or the computer word size is limited.

W3.8.1 Amplitude Scaling

There are two types of scaling that are sometimes carried out: amplitude scaling and time scaling, as we have already seen in Section 3.1.4. **Amplitude scaling** is usually performed unwittingly by simply picking units that make sense for the problem at hand. For the ball levitator, expressing the motion in millimeters and the current in milliamps would keep the numbers within a range that is easy to work with. Equations of motion are sometimes developed in the standard SI units such as meters, kilograms, and amperes, but when computing the motion of a rocket going into orbit, using kilometers makes more sense. The equations of motion are usually solved using computer-aided design software, which is often capable of working in any units. For higher-order systems, it becomes important to scale the problem so that system variables have similar numerical variations. A method for accomplishing the best scaling for a complex system is first to estimate the maximum values for each system variable and then to scale the system so that each variable varies between -1 and 1.

In general, we can perform amplitude scaling by defining the scaled variables for each state element. If

$$x' = S_x x, \tag{W3.6}$$

then

$$\dot{x}' = S_x \dot{x} \quad \text{and} \quad \ddot{x}' = S_x \ddot{x}. \tag{W3.7}$$

We then pick S_x to result in the appropriate scale change, substitute Eqs. (W3.6) and (W3.7) into the equations of motion, and recompute the coefficients.

EXAMPLE W3.7 *Scaling for the Ball Levitator*

The linearized equation of motion for the ball levitator (see Example 9.2, Chapter 9) is

$$\delta\ddot{x} = 1667\delta x + 47.6\delta i, \tag{W3.8}$$

where δx is in units of meters and δi is in units of amperes. Scale the variables for the ball levitator to result in units of millimeters and milliamps instead of meters and amps.

Solution. Referring to Eq. (W3.6), we define

$$\delta x' = S_x \delta x \quad \text{and} \quad \delta i' = S_i \delta i,$$

such that both S_x and S_i have a value of 1000 in order to convert δx and δi in meters and amps to $\delta x'$ and $\delta i'$ in millimeters and milliamps. Substituting these relations into Eq. (W3.8) and taking note of Eq. (W3.7) yields

$$\delta\ddot{x}' = 1667\delta x' + 47.6\frac{S_x}{S_i}\delta i'.$$

In this case, $S_x = S_i$, so Eq. (W3.8) remains unchanged. Had we scaled the two quantities by different amounts, there would have been a change in the last coefficient in the equation.

W3.8.2 Time Scaling

The unit of time when using SI units or English units is seconds. Computer-aided design software is *usually* able to compute results accurately no matter how fast or slow the particular problem at hand. However, if a dynamic system responds in a few microseconds, or if there are characteristic frequencies in the system on the order of several megahertz, the problem may become ill conditioned, so that the numerical routines produce errors. This can be particularly troublesome for high-order systems. The same holds true for an extremely slow system. It is therefore useful to know how to change the units of time should you encounter an ill-conditioned problem.

We define the new scaled time to be

$$\tau = \omega_o t \tag{W3.9}$$

such that, if t is measured in seconds and $\omega_o = 1000$, then τ will be measured in milliseconds. The effect of the **time scaling** is to change the differentiation so that

$$\dot{x} = \frac{dx}{dt} = \frac{dx}{d(\tau/\omega_o)} = \omega_o \frac{dx}{d\tau} \tag{W3.10}$$

and

$$\ddot{x} = \frac{d^2x}{dt^2} = \omega_o^2 \frac{d^2x}{d\tau^2}. \tag{W3.11}$$

EXAMPLE W3.8 *Time Scaling an Oscillator*

The equation for an oscillator was derived in Example 2.5. For a case with a very fast natural frequency $\omega_n = 15{,}000$ rad/sec (about 2 kHz), Eq. (2.23) can be rewritten as

$$\ddot{\theta} + 15{,}000^2 \cdot \theta = 10^6 \cdot T_c.$$

Determine the time-scaled equation so that the unit of time is milliseconds.

Solution. The value of ω_o in Eq. (W3.9) is 1000. Equation (W3.11) shows that

$$\frac{d^2\theta}{d\tau^2} = 10^{-6} \cdot \ddot{\theta},$$

and the time-scaled equation becomes

$$\frac{d^2\theta}{d\tau^2} + 15^2 \cdot \theta = T_c.$$

In practice, we would then solve the equation

$$\ddot{\theta} + 15^2 \cdot \theta = T_c \tag{W3.12}$$

and label the plots in milliseconds instead of seconds.

W3.8.3 Time and Amplitude Scaling in State-Space

We have already discussed time and amplitude scaling in Chapter 3. We now extend the ideas to the state-variable form. Time scaling with $\tau = \omega_o t$ replaces Eq. (7.3) with

$$\frac{d\mathbf{x}}{d\tau} = \frac{1}{\omega_o}\mathbf{A}\mathbf{x} + \frac{1}{\omega_o}\mathbf{B}u = \hat{\mathbf{A}}\mathbf{x} + \hat{\mathbf{B}}u. \tag{W3.13}$$

Amplitude scaling of the state corresponds to replacing $\mathbf{x}$ with $\mathbf{z} = \mathbf{D}_x^{-1}\mathbf{x}$, where $\mathbf{D}_x$ is a diagonal matrix of scale factors. Input scaling corresponds to replacing u with $v = \mathbf{D}_u^{-1}u$. With these substitutions,

$$\mathbf{D}_x\dot{\mathbf{z}} = \frac{1}{\omega_o}\mathbf{A}\mathbf{D}_x\mathbf{z} + \frac{1}{\omega_o}\mathbf{B}\mathbf{D}_u v. \tag{W3.14}$$

Then

$$\dot{\mathbf{z}} = \frac{1}{\omega_o}\mathbf{D}_x^{-1}\mathbf{A}\mathbf{D}_x\mathbf{z} + \frac{1}{\omega_o}\mathbf{D}_x^{-1}\mathbf{B}\mathbf{D}_u v = \hat{\mathbf{A}}\mathbf{z} + \hat{\mathbf{B}}v. \tag{W3.15}$$

Equation (W3.15) compactly expresses the time- and amplitude-scaling operations. Regrettably, it does not relieve the engineer of the responsibility of actually thinking of good scale factors so that scaled equations are in good shape.

EXAMPLE W3.9 *Time Scaling an Oscillator*

The equation for an oscillator was derived in Example 2.5. For a case with a very fast natural frequency $\omega_n = 15{,}000$ rad/sec (about 2 kHz), Eq. (2.23) can be rewritten as

$$\ddot{\theta} + 15{,}000^2 \cdot \theta = 10^6 \cdot T_c.$$

Determine the time-scaled equation so that the unit of time is milliseconds.

Solution. In state-variable form with a state vector $\mathbf{x} = [\theta \ \dot{\theta}]^T$, the unscaled matrices are

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -15{,}000^2 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 10^6 \end{bmatrix}.$$

Applying Eq. (W3.13) results in

$$\hat{\mathbf{A}} = \begin{bmatrix} 0 & \frac{1}{1000} \\ -\frac{15{,}000^2}{1000} & 0 \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{B}} = \begin{bmatrix} 0 \\ 10^3 \end{bmatrix},$$

which yields state-variable equations that are scaled.

Appendix W4.1.4.1
The Filtered Case

Thus far the analysis has been based on the simplest open- and closed-loop structures. A more general case includes a dynamic filter on the input and also dynamics in the sensor. The filtered open-loop structure is shown in Fig. W4.1 having the transfer function $T_{ol} = GD_{ol}F$. In this case, the open-loop controller transfer function has been simply replaced by DF and the discussion given for the unfiltered open-loop case is easily applied to this change.

For the filtered feedback case shown in Fig. W4.2, the changes are more significant. In that case, the transform of the system output is given by

$$Y = \frac{GD_{cl}F}{1 + GD_{cl}H}R + \frac{G}{1 + GD_{cl}H}W - \frac{HGD_{cl}}{1 + GD_{cl}H}V. \qquad \text{(W4.1)}$$

As is evident from this equation, the sensor dynamics, H, is part of the loop transfer function and enters into the question of stability with $D_{cl}H$ replacing the D_{cl} of the unity feedback case. In fact, if $F = H$, then with respect to stability, tracking, and regulation, the filtered case is identical to the unity case with $D_{cl}H$ replacing D_{cl}. On the other hand, the filter transfer function F can play the role of the open-loop controller except that here the filter F would be called on to modify the entire loop transfer function, $\frac{GD_{cl}}{1+GD_{cl}H}$, rather than simply GD_{ol}. Therefore the filtered closed-loop structure can realize the best

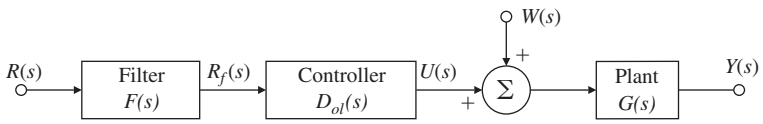


Figure W4.1

Filtered open-loop system

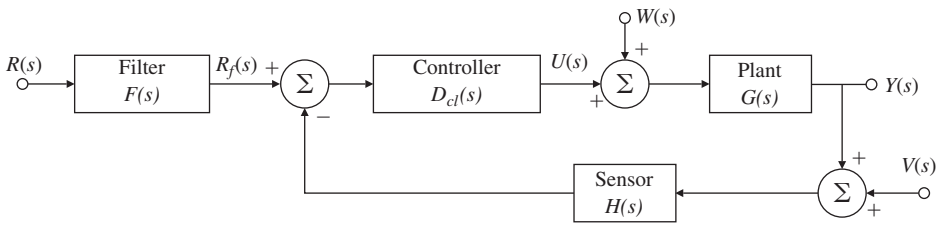


Figure W4.2

Filtered closed-loop. R = reference, U = control, Y = output, V = sensor noise

properties of both the open-loop and the unity feedback closed-loop cases. The controller, D_{cl}, can be designed to effectively regulate the system for the disturbance W and the sensor noise V, while the filter F is designed to improve the tracking accuracy. If the sensor dynamics H are accessible to the designer, this term also can be designed to improve the response to the sensor noise. The remaining issue is sensitivity.

Using the formula given in Eq. (4.18), with changes in the parameter of interest, we can compute

$$\mathcal{S}_F^{T_{cl}} = 1.0, \tag{W4.2}$$

$$\mathcal{S}_G^{T_{cl}} = \frac{1}{1 + GD_{cl}H}, \tag{W4.3}$$

$$\mathcal{S}_H^{T_{cl}} = -\frac{GD_{cl}H}{1 + GD_{cl}H}. \tag{W4.4}$$

Of these, the most interesting is the last. Notice that with respect to H, the sensitivity approaches unity as the loop gain grows. Therefore it is particularly important that the transfer function of the sensor be not only low in noise but also very stable in gain. Money spent on the sensor is money well spent!

EXAMPLE W4.1

If S is the sensitivity of the filtered feedback system to changes in the plant transfer function and T is the transfer function from reference to output, compute the sum of $S + T$. Show that $S + T = 1$ if $F = H$.

(a) Compute the sensitivity of the filtered feedback system shown in Fig. W4.2 with respect to changes in the plant transfer function, G.
(b) Compute the sensitivity of the filtered feedback system shown in Fig. W4.2 with respect to changes in the controller transfer function, D_{cl}.
(c) Compute the sensitivity of the filtered feedback system shown in Fig. W4.2 with respect to changes in the filter transfer function, F.
(d) Compute the sensitivity of the filtered feedback system shown in Fig. W4.2 with respect to changes in the sensor transfer function, H. If S is the sensitivity of the filtered feedback system to changes in the plant transfer function and T is the transfer function from reference to output, compute the sum of $S + T$. Show that $S + T = 1$ if $F = H$.

Solution. To answer the first question, we need the answer to part (a), so let's start there.

(a) Applying the formula for sensitivity of T to changes in G:

$$T = \frac{FDG}{1 + DGH},$$

$$S = G\frac{1 + DGH}{FDG}\frac{(1 + DGH)FD - FDG(DH)}{(1 + DGH)^2}$$

$$= \frac{1}{1 + DGH}.$$

Now we can do

$$\begin{aligned} S+T &= \frac{1}{1+DGH} + \frac{FDG}{1+DGH} \\ &= \frac{1+FDG}{1+DGH} \\ &= 1 \quad \textit{if } F=H. \end{aligned} \tag{W4.5}$$

(b) Applying the formula for sensitivity of T to changes in D:

$$\begin{aligned} S_T^D &= D\frac{1+DGH}{FDG}\frac{(1+DGH)FG - FDG(GH)}{(1+DGH)^2} \\ &= \frac{1}{1+DGH}. \end{aligned}$$

This is not surprising as D and G are in series.

(c) Applying the formula for sensitivity of T to changes in F:

$$\begin{aligned} S_T^F &= F\frac{1+DGH}{FDG}\frac{(1+DGH)(DG)}{(1+DGH)^2} \\ &\quad \frac{1+DGH}{1+DGH} \\ &= 1. \end{aligned}$$

In this case, the F term is in the open loop so it has a sensitivity of unity.

(d) Applying the formula for sensitivity of T to changes in H:

$$\begin{aligned} S_T^H &= H\frac{1+DGH}{FDG}\frac{(1+DGH)0 - FDG(DG)}{(1+DGH)^2} \\ &= -\frac{DGH}{(1+DGH)}. \end{aligned}$$

Appendix W4.2.2.1
Truxal's Formula for the Error Constants

Truxal (1955) derived a formula for the velocity constant of a Type 1 system in terms of the closed-loop poles and zeros, which is a formula that connects the steady-state error to the system's dynamic response. Since control design often requires a trade-off between these two characteristics, Truxal's formula can be useful to know. Its derivation is quite direct. Suppose the closed-loop transfer function $\mathcal{T}(s)$ of a Type 1 system is

$$\mathcal{T}(s) = K\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}. \tag{W4.6}$$

Since the steady-state error in response to a step input in a Type 1 system is zero, the DC gain is unity; thus

$$\mathcal{T}(0) = 1. \tag{W4.7}$$

The system error is given by

$$E(s) \triangleq R(s) - Y(s) = R(s)\left[1 - \frac{Y(s)}{R(s)}\right] = R(s)[1 - \mathcal{T}(s)]. \tag{W4.8}$$

The system error due to a unit ramp input is given by

$$E(s) = \frac{1-\mathcal{T}(s)}{s^2}. \tag{W4.9}$$

Using the Final Value Theorem, we get

$$e_{ss} = \lim_{s\to 0}\frac{1-\mathcal{T}(s)}{s}. \tag{W4.10}$$

Using L'Hôpital's rule, we rewrite Eq. (W4.10) as

$$e_{ss} = -\lim_{s\to 0}\frac{d\mathcal{T}}{ds}, \tag{W4.11}$$

or

$$e_{ss} = -\lim_{s\to 0}\frac{d\mathcal{T}}{ds} = \frac{1}{K_v}. \tag{W4.12}$$

Equation (W4.12) implies that $1/K_v$ is related to the slope of the transfer function at the origin, which is a result that will also be shown in Section 6.1.2. Using Eq. (W4.7), we can rewrite Eq. (W4.12) as

$$e_{ss} = -\lim_{s\to 0}\frac{d\mathcal{T}}{ds}\frac{1}{\mathcal{T}}, \tag{W4.13}$$

or

$$e_{ss} = -\lim_{s\to 0}\frac{d}{ds}[\ln \mathcal{T}(s)]. \tag{W4.14}$$

Substituting Eq. (W4.6) into Eq. (W4.14), we get

$$e_{ss} = -\lim_{s\to 0} \frac{d}{ds}\left\{\ln\left[K\frac{\prod_{i=1}^{m}(s-z_i)}{\prod_{i=1}^{n}(s-p_i)}\right]\right\}, \tag{W4.15}$$

$$= -\lim_{s\to 0} \frac{d}{ds}\left[\ln K + \sum_{i=1}^{m}\ln(s-z_i) - \sum_{i=1}^{m}\ln(s-p_i)\right], \tag{W4.16}$$

or

$$\frac{1}{K_v} = -\frac{d\ln \mathcal{T}}{ds}\bigg|_{s=0} = \sum_{i=1}^{n} -\frac{1}{p_i} + \sum_{i=1}^{m}\frac{1}{z_i}. \tag{W4.17}$$

We observe from Eq. (W4.17) that K_v increases as the closed-loop poles move away from the origin. Similar relationships exist for other error coefficients, and these are explored in the problems.

EXAMPLE W4.2 *Truxal's Formula*

Truxal's formula

A third-order Type 1 system has closed-loop poles at $-2 \pm 2j$ and -0.1. The system has only one closed-loop zero. Where should the zero be if a $K_v = 10\ \text{sec}^{-1}$ is desired?

Solution. From Truxal's formula, we have

$$\frac{1}{K_v} = -\frac{1}{-2+2j} - \frac{1}{-2-2j} - \frac{1}{-0.1} + \frac{1}{z},$$

or

$$0.1 = 0.5 + 10 + \frac{1}{z},$$

$$\frac{1}{z} = 0.1 - 0.5 - 10$$

$$= -10.4.$$

Therefore, the closed-loop zero should be at $z = 1/-10.4 = -0.096$.

Appendix W4.5
Introduction to Digital Control

As a result of the revolution in the cost-effectiveness of digital computers, there has been an increasing use of digital logic in embedded applications such as controllers in feedback systems. A digital controller gives the designer much more flexibility to make modifications to the control law after the hardware design is fixed, because the formula for calculating the control signal is in the software rather than the hardware. In many instances, this means that the hardware and software designs can proceed almost independently, saving a great deal of time. Also, it is relatively easy to include binary logic and nonlinear operations as part of the function of a digital controller as compared to an analog controller. Special processors designed for real-time signal processing and known as digital signal processors (DSPs) are particularly well suited for use as real-time controllers. Chapter 8 includes a more extensive introduction to the math and concepts associated with the analysis and design of digital controllers and digital control systems. However, in order to be able to compare the analog designs of the next three chapters with reasonable digital equivalents, we give here a brief introduction to the most simple techniques for digital designs.

A digital controller differs from an analog controller in that the signals must be **sampled** and **quantized**.[1] A signal to be used in digital logic needs to be sampled first and then the samples need to be converted by an analog-to-digital converter or A/D[2] into a quantized digital number. Once the digital computer has calculated the proper next control signal value, this value needs to be converted back into a voltage and held constant or otherwise extrapolated by a digital-to-analog converter or D/A[3] in order to be applied to the actuator of the process. The control signal is not changed until the next sampling period. As a result of sampling, there are strict limits on the speed and bandwidth of a digital controller. Discrete design methods that tend to minimize these limitations are described in Chapter 8. A reasonable rule of thumb for selecting the sampling period is that, during the rise-time of the response to a step, the input to the discrete controller should be sampled approximately six times. By adjusting the controller for the effects

[1] A controller that operates on signals that are sampled but *not* quantized is called **discrete**, while one that operates on signals that are both sampled and quantized is called **digital**.

[2] Pronounced "A to D."

[3] Often spelled DAC and pronounced as one word to rhyme with quack.

of sampling, the sample period can be as large as two to three times per rise time. This corresponds to a sampling frequency that is 10 to 20 times the system's closed-loop bandwidth. The quantization of the controller signals introduces an equivalent extra noise into the system; to keep this interference at an acceptable level, the A/D converter usually has an accuracy of 10 to 12 bits although inexpensive systems have been designed with only 8 bits. For a first analysis, the effects of the quantization are usually ignored, as they will be in this introduction. A simplified block diagram of a system with a digital controller is shown in Fig. W4.3.

For this introduction to digital control, we will describe a simplified technique for finding a discrete (sampled but not quantized) equivalent to a given continuous controller. The method depends on the sampling period, T_s, being short enough that the reconstructed control signal is close to the signal that the original analog controller would have produced. We also assume that the numbers used in the digital logic have enough accurate bits so that the quantization implied in the A/D and D/A processes can be ignored. While there are good analysis tools to determine how well these requirements are met, here we will test our results by simulation, following the well-known advice that "The proof of the pudding is in the eating."

Finding a discrete equivalent to a given analog controller is equivalent to finding a recurrence equation for the samples of the control, which will approximate the differential equation of the controller. The assumption is that we have the transfer function of an analog controller and wish to replace it with a discrete controller that will accept samples of the controller input $e(kT_s)$ from a sampler and, using past values of the control signal $u(kT_s)$ and present and past samples of the input $e(kT_s)$, will compute the next control signal to be sent to the actuator. As an example, consider a PID controller with the transfer function

$$U(s) = (k_P + \frac{k_I}{s} + k_D s)E(s), \tag{W4.18}$$

which is equivalent to the three terms of the time-domain expression

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau)d\tau + k_D \dot{e}(t), \tag{W4.19}$$

$$= u_P + u_I + u_D. \tag{W4.20}$$

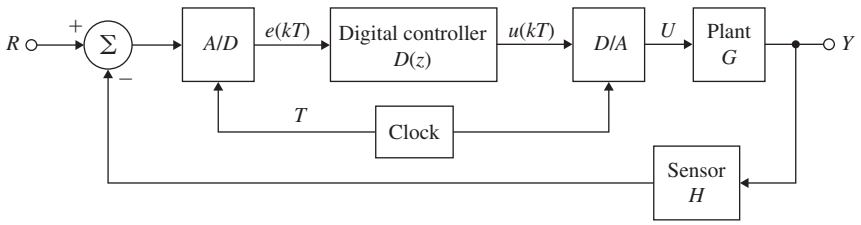


Figure W4.3
Block diagram of a digital controller

Based on these terms and the fact that the system is linear, the next control sample can be computed term-by-term. The proportional term is immediate:

$$u_P(kT_s + T_s) = k_P e(kT_s + T_s). \tag{W4.21}$$

The integral term can be computed by breaking the integral into two parts and approximating the second part, which is the integral over one sample period, as follows.

$$u_I(kT_s + T_s) = k_I \int_0^{kT_s+T_s} e(\tau)d\tau, \tag{W4.22}$$

$$= k_I \int_0^{kT_s} e(\tau)d\tau + k_I \int_{kT_s}^{kT_s+T_s} e(\tau)d\tau, \tag{W4.23}$$

$$= u_I(kT_s) + \{\text{area under } e(\tau) \text{ over one period}\}, \tag{W4.24}$$

$$\cong u_I(kT_s) + k_I \frac{T_s}{2}\{e(kT_s + T_s) + e(kT_s)\}. \tag{W4.25}$$

In Eq. (W4.25), the area in question has been approximated by that of the trapezoid formed by the base T_s and vertices $e(kT_s+T_s)$ and $e(kT_s)$ as shown by the dashed line in Fig. W4.4.

The area also can be approximated by the rectangle of amplitude $e(kT_s)$ and width T_s shown by the solid blue in Fig. W4.4 to give $u_I(kT_s + T_s) = u_I(kT_s) + k_I T_s e(kT_s)$. These and other possibilities are considered in Chapter 8.

In the derivative term, the roles of u and e are reversed from integration and the consistent approximation can be written down at once from Eqs. (W4.25) and (W4.19) as

$$\frac{T_s}{2}\{u_D(kT_s + T_s) + u_D(kT_s)\} = k_D\{e(kT_s + T_s) - e(kT_s)\}. \tag{W4.26}$$

As with linear analog transfer functions, these relations are greatly simplified and generalized by the use of transform ideas. At this time, the discrete transform will be introduced simply as a prediction operator z much as if we

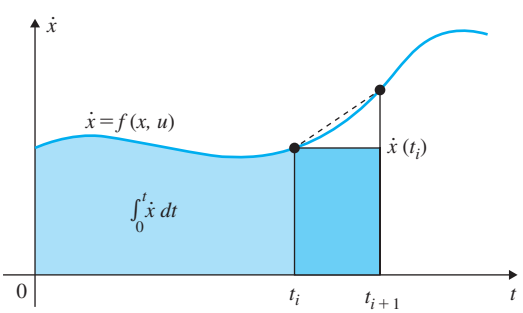


Figure W4.4
Graphical interpretation of numerical integration

described the Laplace transform variable, s, as a differential operator.[4] Here we define the operator z as the forward shift operator in the sense that if $U(z)$ is the transform of $u(kT_s)$ then $zU(z)$ will be the transform of $u(kT_s + T_s)$. With this definition, the integral term can be written as

$$zU_I(z) = U_I(z) + k_I \frac{T_s}{2}\left[zE(z) + E(z)\right], \tag{W4.27}$$

$$U_I(z) = k_I \frac{T_s}{2}\frac{z+1}{z-1}E(z), \tag{W4.28}$$

and from Eq. (W4.26), the derivative term becomes the inverse as

$$U_D(z) = k_D \frac{2}{T_s}\frac{z-1}{z+1}E(z). \tag{W4.29}$$

The complete discrete PID controller is thus described by

$$U(z) = \left(k_P + k_I \frac{T_s}{2}\frac{z+1}{z-1} + k_D \frac{2}{T_s}\frac{z-1}{z+1}\right)E(z). \tag{W4.30}$$

Comparing the two discrete equivalents of integration and differentiation with the corresponding analog terms, it is seen that the effect of the discrete approximation in the z domain is as if everywhere in the analog transfer function the operator s has been replaced by the composite operator $\frac{2}{T_s}\frac{z-1}{z+1}$. This is the trapezoid rule[5] of discrete equivalents.

Trapezoid rule

The discrete equivalent to $D_c(s)$ is

$$D_d(z) = D_c\left(\frac{2}{T_s}\frac{z-1}{z+1}\right). \tag{W4.31}$$

EXAMPLE W4.3 *Discrete Equivalent*

Find the discrete equivalent to the analog controller having transfer function

$$D_c(s) = \frac{U(s)}{E(s)} = \frac{11s+1}{3s+1}, \tag{W4.32}$$

using the sample period $T_s = 1$.

Solution. The discrete operator is $\frac{2(z-1)}{z+1}$ and thus the discrete transfer function is

$$D_d(z) = \frac{U(z)}{E(z)} = D_c(s)\Big|_{s=\frac{2}{T_s}\frac{z-1}{z+1}}, \tag{W4.33}$$

$$= \frac{11\left[\frac{2(z-1)}{z+1}\right]+1}{3\left[\frac{2(z-1)}{z+1}\right]+1}. \tag{W4.34}$$

[4] This is defined as the z-transform in Chapter 8.

[5] The formula is also called Tustin's method after the English engineer who used the technique to study the responses of nonlinear circuits.

Clearing fractions, the discrete transfer function is

$$D_d(z) = \frac{U(z)}{E(z)} = \frac{23z - 21}{7z - 5}. \tag{W4.35}$$

Converting the discrete transfer function to a discrete difference equation using the definition of z as the forward shift operator is done as follows. First we cross-multiply in Eq. (W4.35) to obtain

$$(7z - 5)U(z) = (23z - 21)E(z), \tag{W4.36}$$

and interpreting z as a shift operator, this is equivalent to the difference equation[6]

$$7u(k + 1) - 5u(k) = 23e(k + 1) - 21e(k), \tag{W4.37}$$

where we have replaced $kT_s + T_s$ with $k + 1$ to simplify the notation. To compute the next control at time $kT_s + T_s$, therefore, we solve the difference equation

$$u(k + 1) = \frac{5}{7}u(k) + \frac{23}{7}e(k + 1) - \frac{21}{7}e(k). \tag{W4.38}$$

Now let's apply these results to a control problem. Fortunately, Matlab provides us with the Simulink capability to simulate both continuous and discrete systems allowing us to compare the responses of the systems with continuous and discrete controllers.

EXAMPLE W4.4

Equivalent Discrete Controller for Speed Control

A motor speed control is found to have the plant transfer function

$$\frac{Y}{U} = \frac{45}{(s + 9)(s + 5)}. \tag{W4.39}$$

A PI controller designed for this system has the transfer function

$$D_c(s) = \frac{U}{E} = 1.4\frac{s + 6}{s}. \tag{W4.40}$$

The closed-loop system has a rise time of about 0.2 sec and an overshoot of about 20%. Design a discrete equivalent to this controller and compare the step responses and control signals of the two systems. (a) Compare the responses if the sample period is 0.07, which is about three samples per rise time. (b) Compare the responses with a sample period of $T_s = 0.035$ sec, which corresponds to about six samples per rise time.

---

[6]The process is entirely similar to that used in Chapter 3 to find the ordinary differential equation to which a rational Laplace transform corresponds.

Solution. (a) Using the substitution given by Eq. (W4.31), the discrete equivalent for $T_s = 0.07$ sec is given by replacing s by $s \leftarrow \frac{2}{0.07}\frac{z-1}{z+1}$ in $D_c(s)$ as

$$D_d(z) = 1.4\frac{\frac{2}{0.07}\frac{z-1}{z+1} + 6}{\frac{2}{0.07}\frac{z-1}{z+1}}, \tag{W4.41}$$

$$= 1.4\frac{2(z-1) + 6 * 0.07(z+1)}{2(z-1)}, \tag{W4.42}$$

$$= 1.4\frac{1.21z - 0.79}{(z-1)}. \tag{W4.43}$$

Based on this expression, the equation for the control is (the sample period is suppressed)

$$u(k+1) = u(k) + 1.4 * [1.21e(k+1) - 0.79e(k)]. \tag{W4.44}$$

(b) For $T_s = 0.035$ sec, the discrete transfer function is

$$D_d = 1.4\frac{1.105z - 0.895}{z-1}, \tag{W4.45}$$

for which the difference equation is

$$u(k+1) = u(k) + 1.4[1.105\,e(k+1) - 0.895\,e(k)].$$

A Simulink block diagram for simulating the two systems is given in Fig. W4.5, and plots of the step responses are given in Fig. W4.6a. The respective control signals are plotted in Fig. W4.6b. Notice that the discrete controller for $T_s = 0.07$ sec results in a substantial increase in the overshoot in the step response, while with $T_s = 0.035$ sec the digital controller matches the performance of the analog controller fairly well.

For controllers with many poles and zeros, making the continuous-to-discrete substitution called for in Eq. (W4.31) can be very tedious.

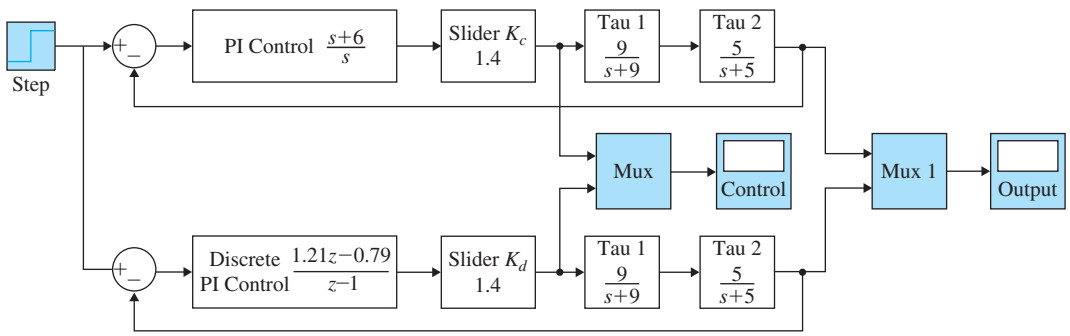


Figure W4.5
Simulink block diagram to compare continuous and discrete controllers

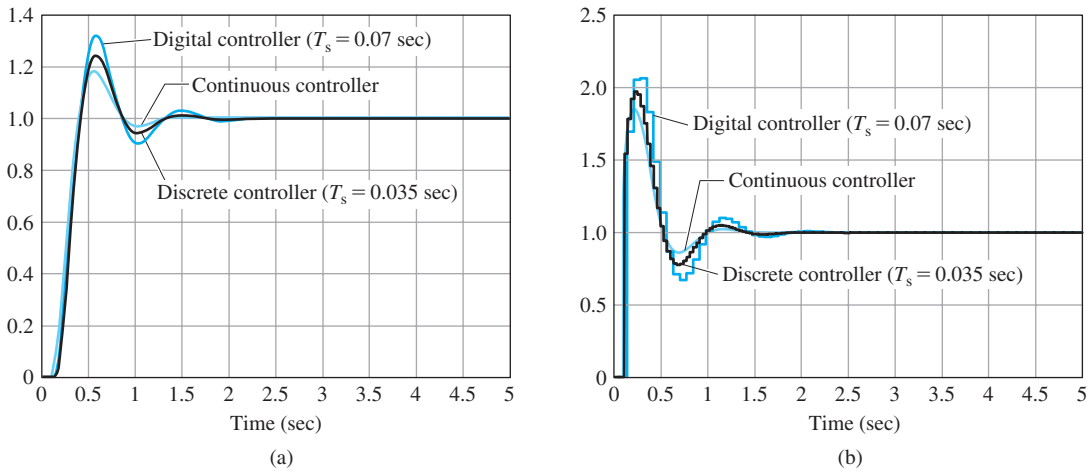


Figure W4.6

Comparison plots of a speed control system with continuous and discrete controllers: (a) output responses and (b) control signals

Fortunately, Matlab provides a command that does all the work. If one has a continuous transfer function given by $D_c(s) = \frac{\text{numD}}{\text{denD}}$ represented in Matlab as sysDc = tf(numD,denD), then the discrete equivalent with sampling period T_s is given by

$$\text{sysDd} = \text{c2d(sysDc,T}_s\text{,'t')}. \tag{W4.46}$$

In this expression, of course, the polynomials are represented in Matlab form. The last parameter in the c2d function given by 't' calls for the conversion to be done using the trapezoid (or Tustin) method. The alternatives can be found by asking Matlab for help c2d. For example, to compute the polynomials for $T_s = 0.07$ sec for Example W4.4, the commands would be

```
numD = [1 6];
denD = [1 - 0];
sysDc = tf(numD,denD)
sysDd = c2d( sysDc,0.07,'t')
```

Appendix W4.6
Sensitivity of Time Response to Parameter Change

We have considered the effects of errors on the steady-state gain of a dynamic system and showed how feedback control can reduce these errors. Since many control specifications are in terms of the step response, the sensitivity of the time response to parameter changes is sometimes very useful to explore. For example, by looking at the sensitivity plot, we can tell if increasing a particular parameter will increase or decrease the overshoot of the response.[1] The following analysis is also a good exercise in small-signal linearization.

To consider the sensitivity of the output $y(t, \theta)$ of a system having a parameter of interest, θ, we compute the effect of a perturbation in the parameter, $\delta\theta$, on the nominal response by using the Taylor's series expansion

$$y(t, \theta + \delta\theta) = y(t, \theta) + \frac{\partial y}{\partial \theta}\delta\theta + \cdots . \tag{W4.47}$$

The first-order approximation of the parameter perturbation effect is the term

$$\delta y(t) = \frac{\partial y}{\partial \theta}\delta\theta. \tag{W4.48}$$

This function can be generated from the system itself as shown by Perkins et al., 1991. We assume that the response depends linearly on the parameter and therefore that the overall transfer function $T(s, \theta)$ is composed of component transfer functions that can be defined to bring out the dependence on the parameter explicitly. A block diagram of the transfer function in terms of the components $T_{ij}(s)$ can be expressed as shown in Fig. W4.7, where we have labeled the parameter as θ and its input signal as Z. In terms

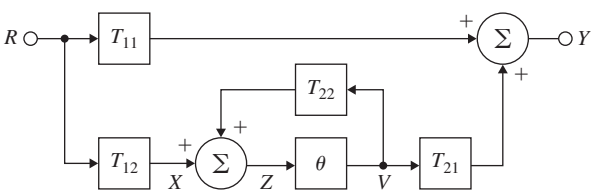


Figure W4.7
Block diagram showing the dependence of output Y on parameter θ

[1] As we will see in Chapter 5, the development of the MATLAB™ root locus interface rltool gives the designer a computer aid to this result.

of this block diagram, the equations relating Y and Z to the reference input can be written immediately.

$$Y = T_{11}R + T_{21}\theta Z, \tag{W4.49}$$

$$Z = T_{12}R + T_{22}\theta Z. \tag{W4.50}$$

The perturbed equations are

$$Y + \delta Y = T_{11}R + T_{21}(\theta + \delta\theta)(Z + \delta Z), \tag{W4.51}$$

$$Z + \delta Z = T_{12}R + T_{22}(\theta + \delta\theta)(Z + \delta Z). \tag{W4.52}$$

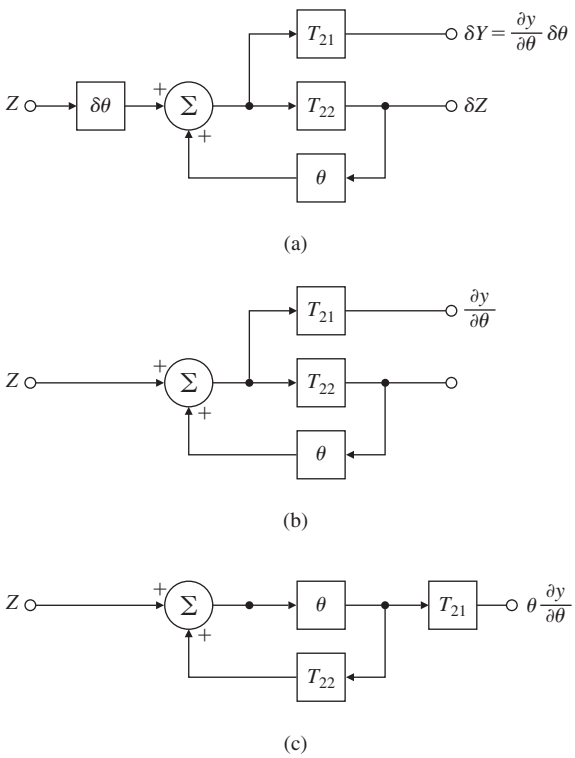
Multiplying these out and ignoring the small term $\delta\theta\delta Z$, the expressions for the perturbations in Y and Z are given by

$$\delta Y = T_{21}(Z\delta\theta + \theta\delta Z), \tag{W4.53}$$

$$\delta Z = T_{22}(Z\delta\theta + \theta\delta Z). \tag{W4.54}$$

The solutions to these equations can be best presented as a block diagram, shown in Fig. W4.8a. The output of this figure is $\delta Y = \dfrac{\partial y}{\partial \theta}\delta\theta$, and we notice that the input Z is multiplied by a gain of $\delta\theta$. Therefore, if we drop the block $\delta\theta$, the output will be simply $\dfrac{\partial y}{\partial \theta}$ as shown in Fig. W4.8b. Finally, to compute the sensitivity as the variation to a percent change in the parameter,

Figure W4.8
Block diagrams showing the generation of (a) δY and δZ; (b) $\dfrac{\partial y}{\partial \theta}$; and (c) $\theta\dfrac{\partial y}{\partial \theta}$



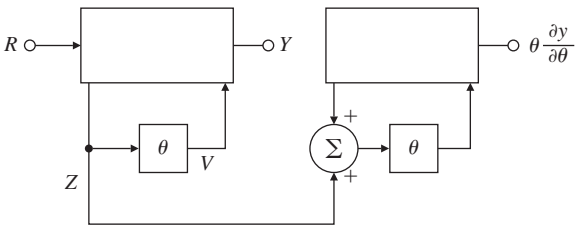


Figure W4.9
Block diagram showing the computation of $\theta \dfrac{\partial y}{\partial \theta}$ from the original transfer function

which is $\dfrac{\partial y}{\partial \ln \theta} = \dfrac{\dfrac{\partial y(t,\theta)}{\partial \theta}}{\dfrac{\partial \ln \theta}{\partial \theta}} = \theta \dfrac{\partial y}{\partial \theta}$, we need only shift the input Z from the output side of the θ block to its input as shown in Fig. W4.8c. We are now in a position to give the final block diagram of the system as it is to be implemented, shown in Fig. W4.9.

In this figure, it is clear that, to compute the sensitivity of the output to a parameter, one needs to simulate two copies of the system. The input to the first system is the reference input of interest, and the input to the second system is at the input to the parameter of interest of the variable Z taken from the input to the parameter in the original system. The transfer function from the reference input to the output sensitivity is readily computed to be

$$\frac{T_{12}\theta T_{21}}{(1-\theta T_{22})^2}. \tag{W4.55}$$

From this function it is clear that:

Response sensitivity

> To keep the sensitivity of the output signal to a parameter change low, it is important to have feedback with high gain around the parameter in question.

EXAMPLE W4.5 *Time-Domain Sensitivity*

Compute the sensitivity of the output of the speed control example shown in the upper portion of Fig. W4.10 with respect to the control gain, K_{cl}. Take the nominal values to be $K_{cl} = 9$, $\tau = 0.01$ sec, and $A = 1$ rad/volt-sec.

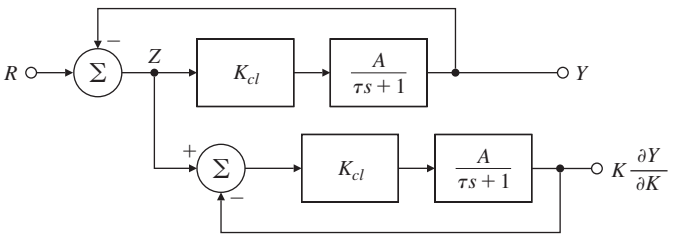


Figure W4.10
Block diagram showing the computation of the sensitivity of the output of the speed control example

Solution. The required block diagram for the computation is given in Fig. W4.10 based on Fig. W4.9. In MATLAB, we will construct the several transfer functions with $T_{ij} = \dfrac{n_{ij}}{d_{ij}}$ and implement Eq. (W4.55). For comparison, we compute the nominal response from Fig. W4.7 and add 10% of the sensitivity to the nominal response. The instructions to do the computation in MATLAB are

```
% script to compute sensitivity for Fig. W4.10
% First input the data for the component transfer functions Tij
% and the nominal parameter, Kcl for this problem
Kcl=9; tau=0.01;
n11=0; d11=1;
n12=1; d12=1;
n22=[0 -1]; d22= [tau 1];
n21=1; d21=[tau 1];
% Now compute the numerator and denominator polynomials of the transfer functions
% using the convolution function conv to multiply the polynomials
% and put them into system transfer function forms with the MATLAB function tf.
% The over-all transfer function is
% Y/R =n11/d11 + (n12*n21*d22)/(d12*d21*[d22-Kcl*n22]) = sysy
% The transfer function from the reference input to the sensitivity is
%  Kcl*(dy/dKcl)/R = sysdy
% Now define the numerators and denominators of several intermediate transfer functions
n1=Kcl*conv(n21,n12);
d1=conv(d21,d12);
n2=d22;
d2=[d22-Kcl*n22];
ny=conv(n1,n2);
dy=conv(d1,d2);
% Now put these together to form two intermediate transfer functions
sysy1 = tf(ny,dy);
sysy2 = tf(n11,d11);
% Now construct the final transfer functions
% The overall transfer function Y/R
sysy=sysy1+sysy2;
% The sensitivity transfer function
ndy=conv(ny,n2);
ddy=conv(dy,d2);
sysdy=tf(ndy,ddy);
% Now use these to compute the step responses and
% plot the output, the sensitivity and a perturbed response
[y,t]=step(sysy);
[yd,t]=step(sysdy);
plot(t,[y yd y+.1*yd]);
```

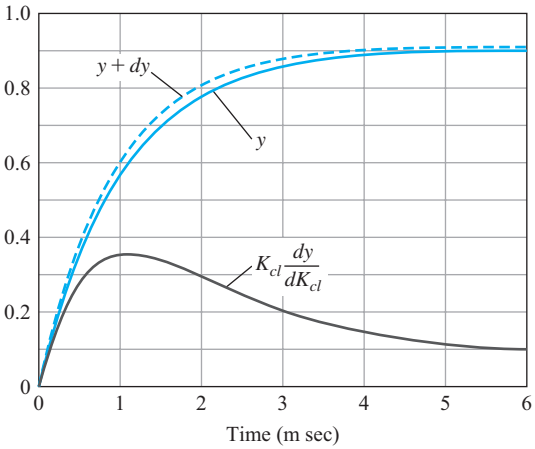


Figure W4.11
Plots of the output, the sensitivity, and the result of a 10% change in the parameter value for the speed control example

These instructions are constructed to compute the sensitivity for any system, given the several transfer functions. The script input is for the specific example. Plots of the output, its sensitivity, and the result of a 10% change in the parameter value are given in Fig. W4.11.

Appendix W5.4.4
Analog and Digital Implementations

Lead compensation can be physically realized in many ways. In analog electronics, a common method is to use an operational amplifier, an example of which is shown in Fig. W5.1. The transfer function of the circuit in Fig. W5.1 is readily found by the methods of Chapter 2 to be

$$D_{lead}(s) = -a\frac{s+z}{s+p}, \tag{W5.1}$$

where

$$a = \frac{p}{z}, \text{ if } R_f = R_1 + R_2,$$

$$z = \frac{1}{R_1 C},$$

$$p = \frac{R_1 + R_2}{R_2} \cdot \frac{1}{R_1 C}.$$

If a design for $D_c(s)$ is complete and a digital implementation is desired, then the technique of Appendix W4.5 can be used by first selecting a sampling period T_s and then making substitution of $\frac{2}{T_s}\frac{z-1}{z+1}$ for s. For example, consider the lead compensation $D_c(s) = \frac{s+2}{s+13}$. Then, since the rise time is about 0.3, a sampling period of six samples per rise time results in the selection of $T_s = 0.05$ sec. With the substitution of $\frac{2}{0.05}\frac{z-1}{z+1}$ for s into this transfer function, the discrete transfer function is

$$\frac{U(z)}{E(z)} = \frac{40\frac{z-1}{z+1} + 2}{40\frac{z-1}{z+1} + 13}$$

$$= \frac{1.55z - 1.4}{1.96z - 1}. \tag{W5.2}$$

Clearing fractions and using the fact that operating on the time functions $zu(kT_s) = u(kT_s + T_s)$, we see that Eq. (W5.2) is equivalent to the formula for the controller given by

$$u(kT_s + T_s) = \frac{1}{1.96}u(kT_s) + \frac{1.55}{1.96}e(kT_s + T_s) - \frac{1.4}{1.96}e(kT_s). \tag{W5.3}$$

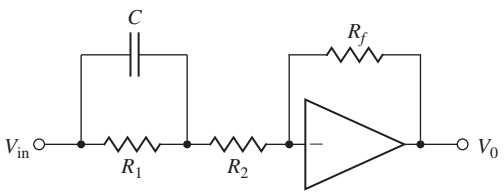


Figure W5.1
Possible circuit of a lead compensation

The Matlab commands to generate the discrete equivalent controller are

```
sysC=tf([1 2],[1 13]);
sysD=c2D(sysC,0.05,tustin)
```

Fig. W5.2 shows the Simulink diagram for implementing the digital controller. The result of the simulation is contained in Fig. W5.3, which shows the comparison of analog and digital control outputs, and Fig. W5.4, which shows the analog and digital control outputs.

As with lead compensation, lag or notch compensation can be implemented using a digital computer and following the same procedure. However,

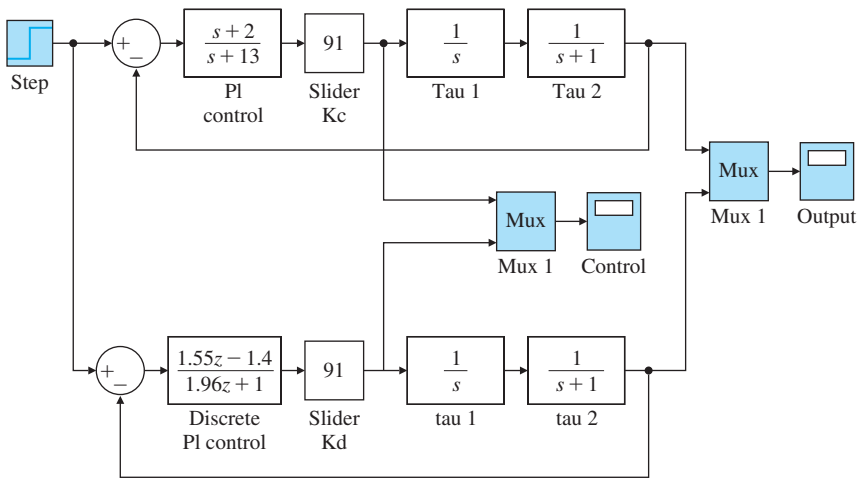


Figure W5.2
Simulink diagram for comparison of analog and digital control

Figure W5.3
Comparison of analog and digital control output responses

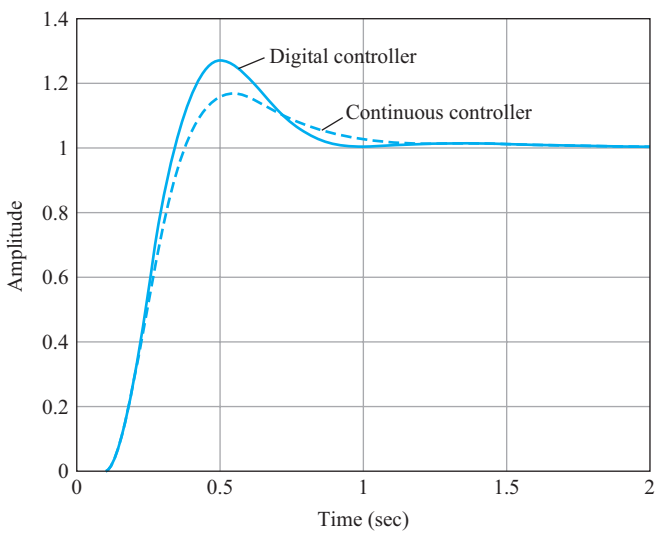


Figure W5.4
Comparison of analog and digital control time histories

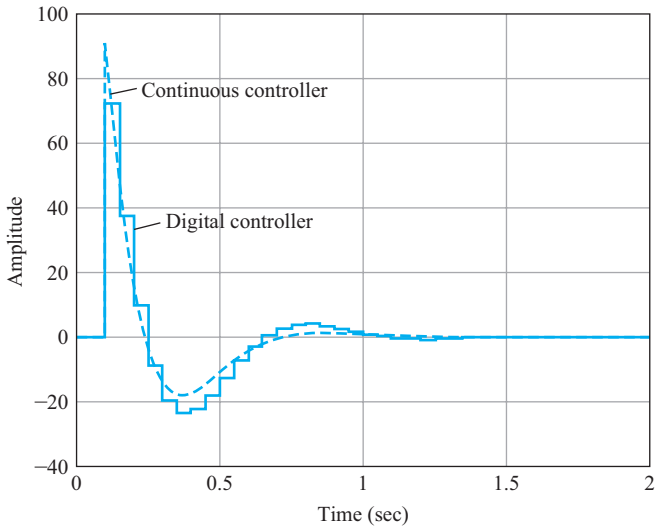
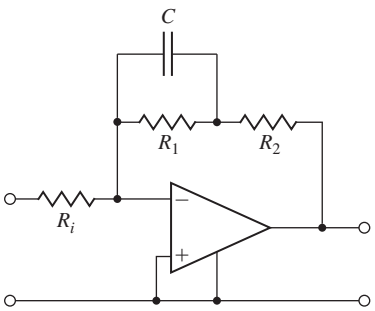


Figure W5.5
Possible circuit of lag compensation



they too can be implemented using analog electronics, and a circuit diagram of a lag network is given in Fig. W5.5. The transfer function of this circuit can be shown to be

$$D_c(s) = -a\frac{s+z}{s+p},$$

where

$$a = \frac{R_2}{R_i},$$

$$z = \frac{R_1 + R_2}{R_1 R_2 C},$$

$$p = \frac{1}{R_1 C}.$$

Usually $R_i = R_2$, so the high-frequency gain is unity, or $a = 1$, and the low-frequency increase in gain to enhance K_v or other error constant is set by $k = a\frac{z}{p} = \frac{R_1+R_2}{R_2}$.

Appendix W5.6.3
Root Locus with Time Delay

Time delays always reduce the stability of a system

Time delays often arise in control systems, both from delays in the process itself and from delays in the processing of sensed signals. Chemical plants often have processes with a time delay representing the time material takes to be transported via pipes or other conveyer. In measuring the attitude of a spacecraft en route to Mars, there is a significant time delay for the sensed quantity to arrive back on Earth due to the speed of light. There is also a small time delay in any digital control system due to the cycle time of the computer and the fact that data is processed at discrete intervals. Time delay *always* reduces the stability of a system; therefore, it is important to be able to analyze its effect. In this section, we discuss how to use the root locus for such analysis. Although an exact method of analyzing time delay is available in the frequency-response methods described in Chapter 6, knowing several different ways to analyze a design provides the control designer with more flexibility and an ability to check the candidate solutions.

An example of a root locus with time delay

Consider the problem of designing a control system for the temperature of the heat exchanger described in Chapter 2. The transfer function between the control A_s and the measured output temperature T_m is described by two first-order terms plus a time delay T_d of 5 sec. The time delay results because the temperature sensor is physically located downstream from the exchanger, so that there is a delay in its reading. The transfer function is

$$G(s) = \frac{e^{-5s}}{(10s+1)(60s+1)}, \tag{W5.4}$$

where the e^{-5s} term arises from the time delay.[1]

The corresponding root-locus equations with respect to proportional gain K are

$$\begin{aligned} 1 + KG(s) &= 0, \\ 1 + K\frac{e^{-5s}}{(10s+1)(60s+1)} &= 0, \\ 600s^2 + 70s + 1 + Ke^{-5s} &= 0. \end{aligned} \tag{W5.5}$$

How would we plot the root locus corresponding to Eq. (W5.5)? Since it is not a polynomial, we cannot proceed with the methods used in previous examples. So we reduce the given problem to one we have previously

[1] Time delay is often referred to as "transportation lag" in the process industries.

solved by approximating the nonrational function e^{-5s} with a rational function. Since we are concerned with control systems and hence typically with low frequencies, we want an approximation that will be good for small s.[2] The most common means for finding such an approximation is attributed to H. Padé. It consists of matching the series expansion of the transcendental function e^{-5s} with the series expansion of a rational function whose numerator is a polynomial of degree p and whose denominator is a polynomial of degree q. The result is called a (p, q) **Padé approximant**[3] to e^{-5s}. We will initially compute the approximants to e^{-s}, and in the final result, we will substitute $T_d s$ for s to allow for any desired delay.

Padé approximant

The resulting $(1, 1)$ Padé approximant ($p = q = 1$) is

$$e^{-T_d s} \cong \frac{1 - (T_d s/2)}{1 + (T_d s/2)}. \tag{W5.6}$$

If we assume $p = q = 2$, we have five parameters and a better match is possible. In this case, we have the $(2, 2)$ approximant, which has the transfer function

$$e^{-T_d s} \cong \frac{1 - T_d s/2 + (T_d s)^2/12}{1 + T_d s/2 + (T_d s)^2/12}. \tag{W5.7}$$

The comparison of these approximants can be seen from their pole–zero configurations as plotted in Fig. W5.6. The locations of the poles are in the LHP and the zeros are in the RHP at the reflections of the poles.

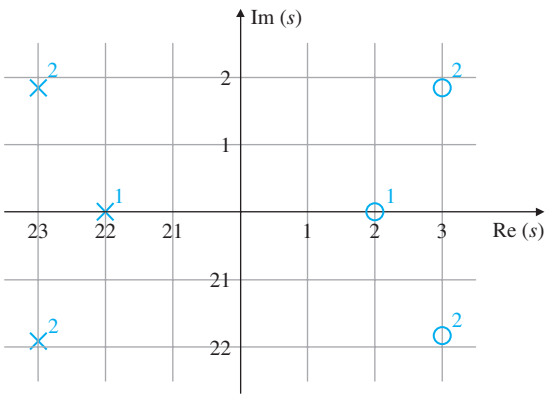


Figure W5.6
Poles and zeros of the Padé approximants to e^{-s}, with superscripts identifying the corresponding approximants; for example, x^1 represents the (1,1) approximant

[2]The nonrational function e^{-5s} is analytic for all finite values of s and so may be approximated by a rational function. If nonanalytic functions such as $\sqrt{s}$ were involved, great caution would be needed in selecting an approximation that is valid near $s = 0$.

[3]The (p, p) Padé approximant for a delay of T sec is most commonly used and is computed by the Matlab command `[num,den] = pade(T, P)`.

In some cases, a very crude approximation is acceptable. For small delays, the (0, 1) approximant can be used, which is simply a first-order lag given by

$$e^{-T_d s} \cong \frac{1}{1 + T_d s}. \tag{W5.8}$$

Contrasting methods of approximating delay

To illustrate the effect of a delay and the accuracy of the different approximations, root loci for the heat exchanger are drawn in Fig. W5.7 for four cases. Notice that, for low gains and up to the point where the loci cross the imaginary axis, the approximate curves are very close to exact. However, the (2, 2) Padé curve follows the exact curve much further than does the first-order lag, and its increased accuracy would be useful if the delay were larger. All analyses of the delay show its destabilizing effect and how it limits the achievable response time of the system.

While the Padé approximation leads to a rational transfer function, in theory, it is not necessary for plotting a root locus. A direct application of the phase condition can be used to plot portions of an exact locus of a system with time delay. The phase-angle condition does not change if the transfer function of the process is nonrational, so we still must search for values of s for which the phase is $180° + 360°l$. If we write the transfer function as

$$G(s) = e^{-T_d s}\bar{G}(s),$$

the phase of $G(s)$ is the phase of $\bar{G}(s)$ minus $\lambda\omega$ for $s = \sigma + j\omega$. Thus we can formulate a root-locus problem as searching for locations where the phase of $\bar{G}(s)$ is $180° + T_d\omega + 360°(l - 1)$. To plot such a locus, we would fix ω and search along a horizontal line in the s-plane until we found a point on the locus, then raise the value of ω, change the target angle, and repeat. Similarly, the departure angles are modified by $T_d\omega$, where ω is the imaginary part of the pole from which the departure is being computed. Matlab does not provide a program to plot the root locus of systems with

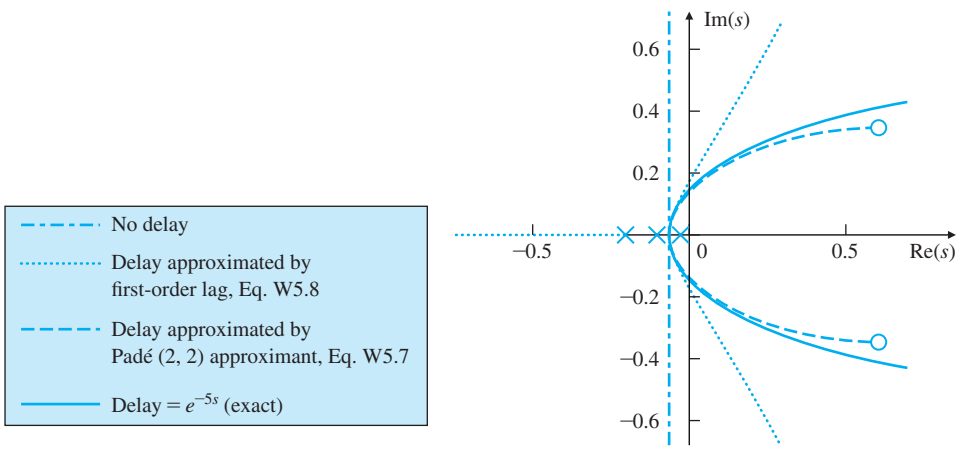


Figure W5.7
Root loci for the heat exchanger with and without time delay

delay, so we must be satisfied here with Padé approximants. Since it is possible to plot the frequency response (or Bode plot) of delay exactly and easily, if the designer feels that the Padé approximant is not satisfactory, the expedient approach is to use the frequency-response design methods described in Chapter 6.

Appendix W6.7.2
Digital Implementation of Example 6.15

EXAMPLE W6.1 *Lead Compensation for a DC Motor*

As an example of designing a lead compensator, let us repeat the design of compensation for the DC motor with the transfer function

$$G(s) = \frac{1}{s(s+1)}$$

that was carried out in Section 5.4.1. This also represents the model of a satellite-tracking antenna (see Fig. 3.60). This time we wish to obtain a steady-state error of less than 0.1 for a unit-ramp input. Furthermore, we desire an overshoot $M_p < 25\%$.

1. Determine the lead compensation satisfying the specifications.
2. Determine the digital version of the compensation with $T_s = 0.05$ sec.
3. Compare the step and ramp responses of both implementations.

Solution.

1. The steady-state error is given by

$$e_{ss} = \lim_{s\to 0} s\left[\frac{D_c}{1 + KD_c(s)G(s)}\right]R(s), \qquad \text{(W6.1)}$$

where $R(s) = 1/s^2$ for a unit ramp, so Eq. (W6.1) reduces to

$$e_{ss} = \lim_{s\to 0}\left\{\frac{1}{s + KD_c(s)[1/(s+1)]}\right\} = \frac{1}{KD_c(0)}.$$

Therefore, we find that $KD(0)$, which is the steady-state gain of the compensation, cannot be less than 10 ($K_v \geq 10$) if it is to meet the error criterion, so we pick $K = 10$. To relate the overshoot requirement to PM, Fig. 6.37 shows that a PM of 45° should suffice. The frequency response of $KG(s)$ in Fig. W6.1 shows that the PM = 20° if no phase lead is added by compensation. If it were possible to simply add phase without affecting the magnitude, we would need an additional phase of only 25° at the $KG(s)$ crossover frequency of $\omega = 3$ rad/sec. However, maintaining the same low-frequency gain and adding a compensator zero would increase the crossover frequency; hence, more than a 25° phase contribution will be required from the lead compensation. To be

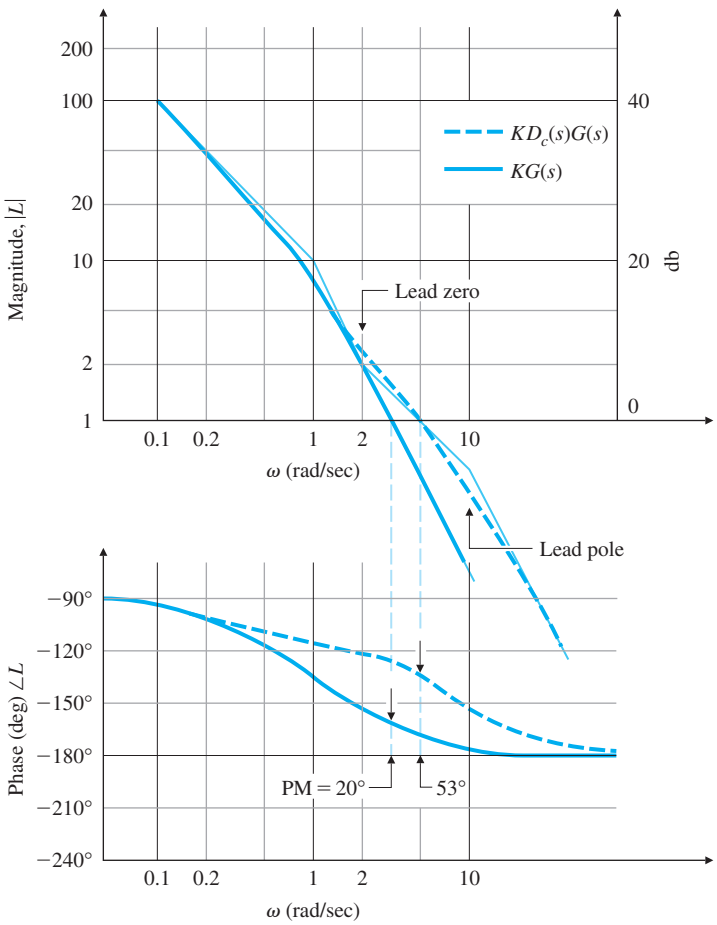


Figure W6.1
Frequency response for lead-compensation design

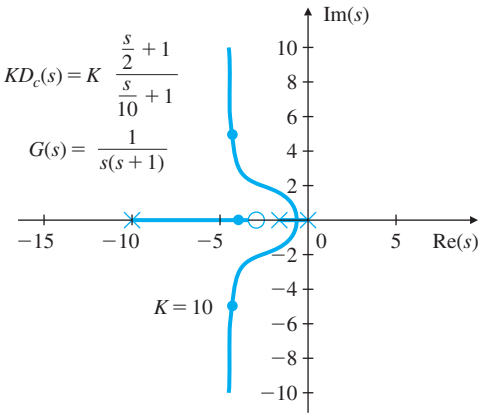
safe, we will design the lead compensator so that it supplies a maximum phase lead of 40°. Fig. 6.53 shows that $1/\alpha = 5$ will accomplish that goal. We will derive the greatest benefit from the compensation if the maximum phase lead from the compensator occurs at the crossover frequency. With some trial and error, we determine that placing the zero at $\omega = 2$ rad/sec and the pole at $\omega = 10$ rad/sec causes the maximum phase lead to be at the crossover frequency. The compensation, therefore, is

$$KD_c(s) = 10\frac{s/2+1}{s/10+1}.$$

The frequency-response characteristics of $L(s) = KD_c(s)G(s)$ in Fig. W6.1 can be seen to yield a PM of 53°, which satisfies the design goals.

The root locus for this design, originally given as Fig. 5.24, is repeated here as Fig. W6.2, with the root locations marked for $K = 10$. The locus is not needed for the frequency-response design procedure; it

Figure W6.2
Root locus for lead-compensation design



is presented here only for comparison with the root locus design method presented in Chapter 5. The entire process can be expedited by the use of Matlab's sisotool routine, which simultaneously provides the root locus and the Bode plot through an interactive GUI interface. For this example, the Matlab statements

```
G=tf(1,[1 1 0]);
Dc=tf(10*[1/2 1],[1/10 1]);
sisotool(G,Dc)
```

will provide the plots as shown in Fig. W6.3. It also can be used to generate the Nyquist and time-response plots if desired.

2. To find the discrete equivalent of $D_c(s)$, we use the trapezoidal rule given by Eq. (W4.31). That is,

$$D_d(z) = \frac{\frac{2}{T_s}\frac{z-1}{z+1}/2 + 1}{\frac{2}{T_s}\frac{z-1}{z+1}/10 + 1}, \tag{W6.2}$$

which, with $T_s = 0.05$ sec, reduces to

$$D_d(z) = \frac{4.2z - 3.8}{z - 0.6}. \tag{W6.3}$$

This same result can be obtained by the Matlab statement

```
sysDc = tf([0.5 1],[0.1 1]);
sysDd = c2d(sysDc, 0.05, 'tustin').
```

Because

$$\frac{U(z)}{E(z)} = KD_d(z), \tag{W6.4}$$

the discrete control equation that results is

$$u(k+1) = 0.6u(k) + 10(4.2e(k+1) - 3.8e(k)). \tag{W6.5}$$

3. The Simulink block diagram of the continuous and discrete versions of $D_c(s)$ controlling the DC motor is shown in Fig. W6.4. The step

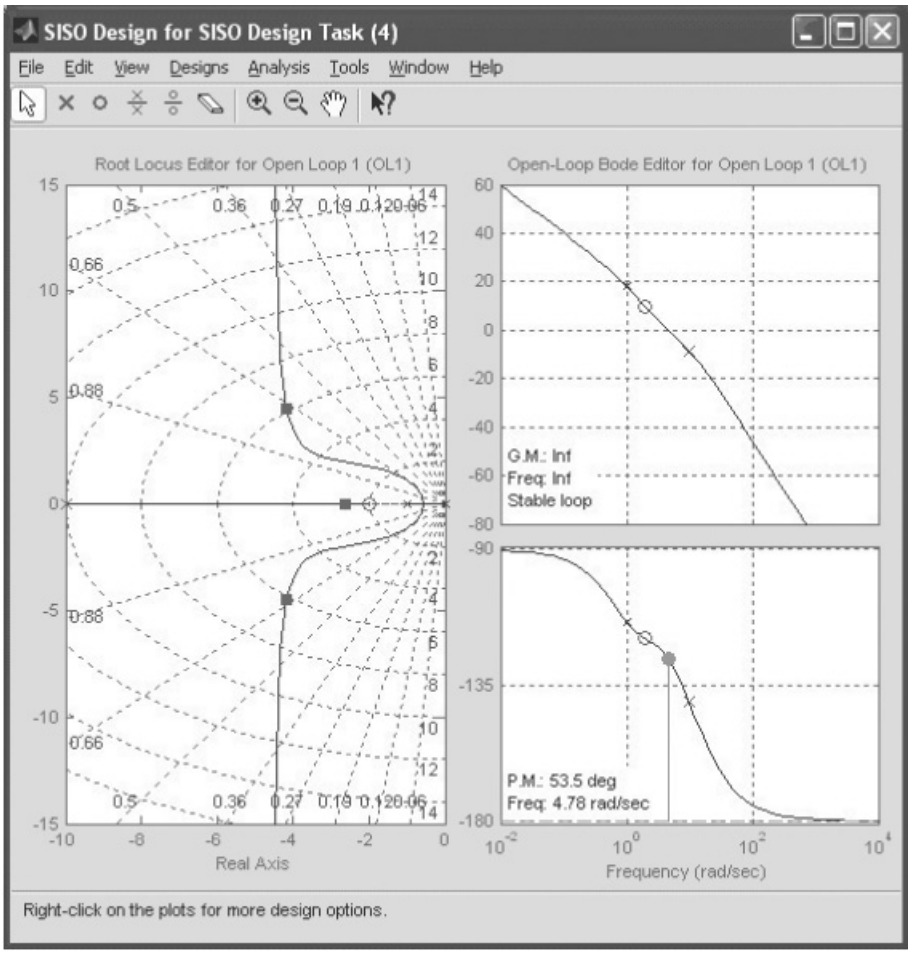


Figure W6.3
SISOTOOL graphical interface for Example W6.1

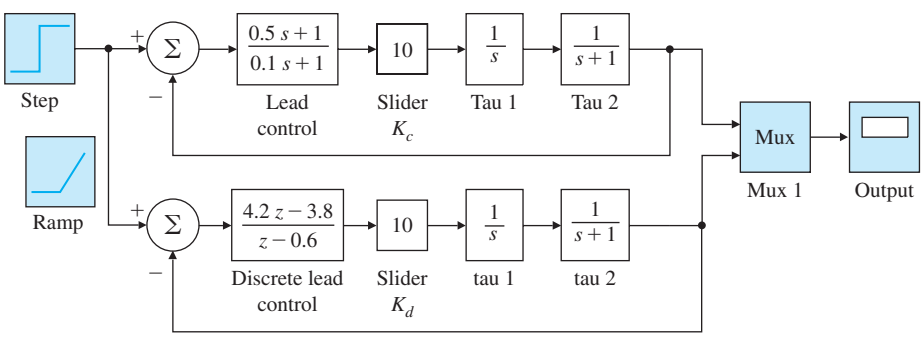
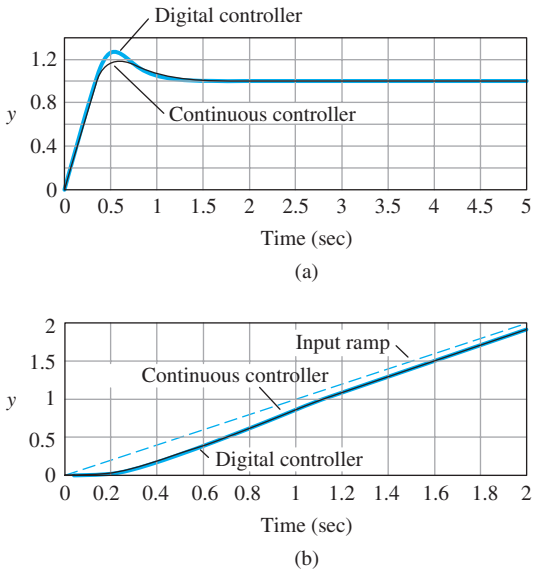


Figure W6.4
Simulink block diagram for transient response of lead-compensation design

Figure W6.5
Lead-compensation design: (a) step response; (b) ramp response



responses of the two controllers are plotted together in Fig. W6.5a and are reasonably close to one another; however, the discrete controller does exhibit slightly increased overshoot, as is often the case. Both overshoots are less than 25%, and thus meet the specifications. The ramp responses of the two controllers, shown in Fig. W6.5b, are essentially identical, and both meet the 0.1 specified error.

Appendix W6.8.1
Time Delay via the Nyquist Diagram

EXAMPLE W6.2 *Nyquist Plot for a System with Time Delay*

Consider the system with

$$KG(s) = \frac{Ke^{-T_d s}}{s},$$

where $T_d = 1$ sec. Determine the range of K for which the system is stable.

Solution. Because the Bode plotting rules do not apply for the phase of a time-delay term, we will use an analytical approach to determine the key features of the frequency response plot. As just discussed, the magnitude of the frequency response of the delay term is unity, and its phase is $-\omega$ radians. The magnitude of the frequency response of the pure integrator is $1/\omega$ with a constant phase of $-\pi/2$. Therefore,

$$\begin{aligned} G(j\omega) &= \frac{1}{\omega} e^{-j(\omega+\pi/2)} \\ &= \frac{1}{\omega}(-\sin\omega - j\cos\omega). \end{aligned} \tag{W6.6}$$

Using Eq. (W6.6) and substituting in different values of ω, we can make the Nyquist plot, which is the spiral shown in Fig. W6.6.

Let us examine the shape of the spiral in more detail. We pick a Nyquist path with a small detour to the right of the origin. The effect of the pole at the origin is the large arc at infinity with a 180° sweep, as shown in Fig. W6.6. From Eq. (W6.6), for small values of $\omega > 0$, the real part of the frequency response is close to -1 because $\sin\omega \cong \omega$ and $\mathrm{Re}[G(j\omega)] \cong -1$. Similarly, for small values of $\omega > 0$, $\cos\omega \cong 1$ and $\mathrm{Im}[G(j\omega)] \cong -1/\omega$—that is, very large negative values, as shown in Fig. W6.6. To obtain the crossover points on the real axis, we set the imaginary part equal to zero:

$$\frac{\cos\omega}{\omega} = 0. \tag{W6.7}$$

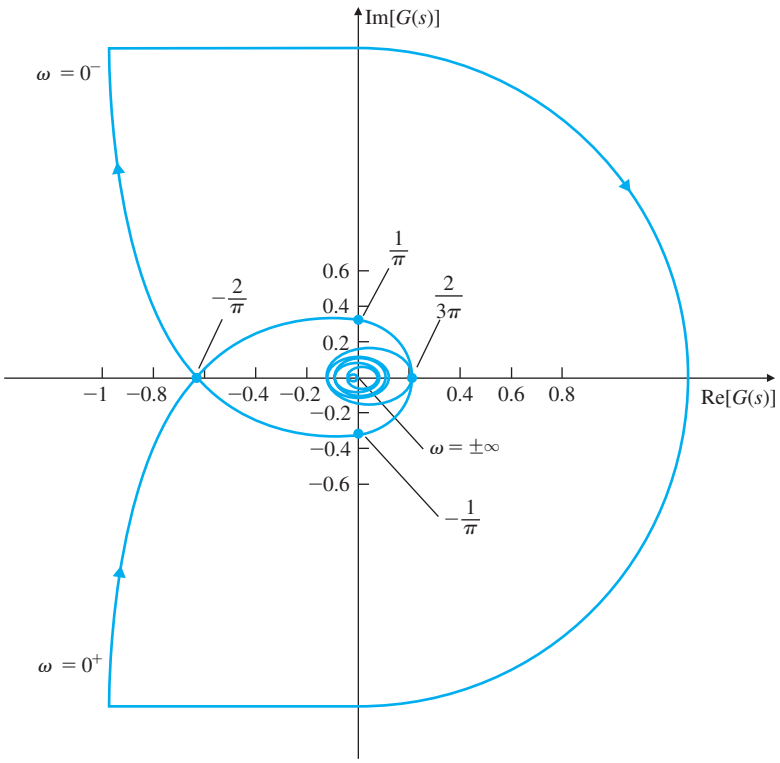
The solution is then

$$\omega_0 = \frac{(2n+1)\pi}{2}, \qquad n = 0, 1, 2, \ldots. \tag{W6.8}$$

After substituting Eq. (W6.8) back into Eq. (W6.6), we find that

$$G(j\omega_0) = \frac{(-1)^n}{(2n+1)}\left(\frac{2}{\pi}\right), \quad n = 0, 1, 2, \ldots.$$

Figure W6.6
Nyquist plot for Example W6.2



So the first crossover of the negative real axis is at $-2/\pi$, corresponding to $n = 0$. The first crossover of the positive real axis occurs for $n = 1$ and is located at $2/3\pi$. As we can infer from Fig. W6.6, there are an infinite number of other crossings of the real axis. Finally, for $\omega = \infty$, the Nyquist plot converges to the origin. Note that the Nyquist plot for $\omega < 0$ is the mirror image of the one for $\omega > 0$.

The number of poles in the RHP is zero ($P = 0$), so for closed-loop stability, we need $Z = N = 0$. Therefore, the Nyquist plot cannot be allowed to encircle the $-1/K$ point. It will not do so as long as

$$-\frac{1}{K} < -\frac{2}{\pi}, \tag{W6.9}$$

which means that, for stability, we must have $0 < K < \pi/2$.

Appendix W6.9.2
The Inverse Nyquist Diagram

The **inverse Nyquist plot** is simply the reciprocal of the Nyquist plot described in Section 6.3 and used in Section 6.4 for the definition and discussion of stability margins. It is obtained most easily by computing the inverse of the magnitude from the Bode plot and plotting that quantity at an angle in the complex plane, as indicated by the phase from the Bode plot. It can be used to find the PM and GM in the same way that the Nyquist plot was used. When $|G(j\omega)| = 1$, $|G^{-1}(j\omega)| = 1$ also, so the definition of PM is identical on the two plots. However, when the phase is $-180°$ or $+180°$, the value of $|G^{-1}(j\omega)|$ is the GM directly; no calculation of an inverse is required, as was the case for the Nyquist plot.

The inverse Nyquist plot for the system in Fig. 6.24 (Example 6.9) is shown in Fig. W6.7 for the case where $K = 1$ and the system is stable. Note that GM $= 2$ and PM $\cong 20°$. As an example of a more complex case, Fig. W6.8 shows an inverse Nyquist plot for the sixth-order case whose Nyquist plot was shown in Fig. 6.41 and whose Nichols chart was shown in Fig. 6.83. Note here that GM $= 1.2$ and PM $\cong 35°$. Had the two crossings of the unit circle not occurred at the same point, the crossing with the smallest PM would have been the appropriate one to use.

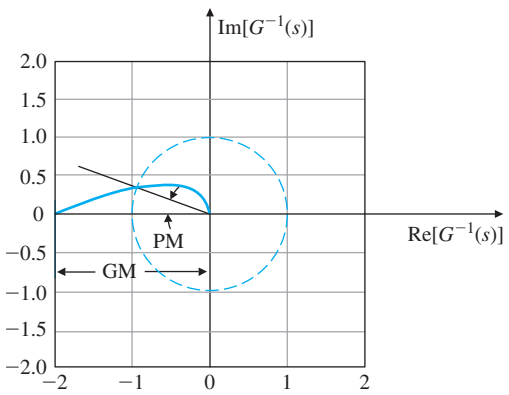
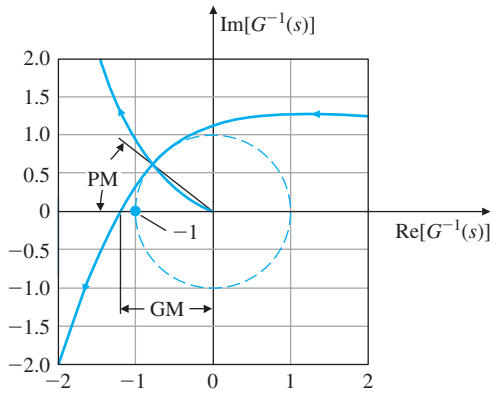


Figure W6.7
Inverse Nyquist plot for Example 6.9

Figure W6.8

Inverse Nyquist plot of the system whose Nyquist plot is in Fig. 6.41



Appendix W7.8
Digital Implementation of Example 7.31

EXAMPLE W7.1 *Redesign of the DC Servo Compensator*

For Example 7.31, derive an equivalent discrete controller with a sampling period of $T_s = 0.1$ sec (10 times the fastest pole), and compare the continuous and digital control outputs and control efforts. Verify the design by plotting the step response and commenting on the comparison of the continuous and discrete responses.

Solution. The discrete equivalent for the controller is obtained from Matlab with the c2d command, as in

```
nc=94.5*conv([1 7.98],[1 2.52]); % form controller numerator
dc=conv([1 8.56 59.5348],[1 10.6]); % form controller denominator
sysDc=tf(nc,dc); % form controller system description
ts=0.1;% sampling time of 0.1 sec
sysDd=c2d(sysDc,ts,'zoh'); % convert controller to discrete time
```

Discrete controller

The resulting controller has the discrete transfer function

$$D_d(z) = \frac{5.9157(z + 0.766)(z + 0.4586)}{(z - 0.522 \pm 0.3903j)(z + 0.3465)}.$$

The equation for the control law (with the sample period suppressed for clarity) is

$$u(k+1) = 1.3905u(k) - 0.7866u(k-1) + 0.1472u(k-2) + e(k) - 7.2445e(k-2) + 2.0782e(k-2).$$

Simulink simulation

A Simulink diagram for simulating both the continuous and discrete systems is shown in Fig. W7.1. A comparison of the continuous and discrete step responses and control signals is shown in Fig. W7.2. Better agreement between the two responses can be obtained if the sampling period is reduced.

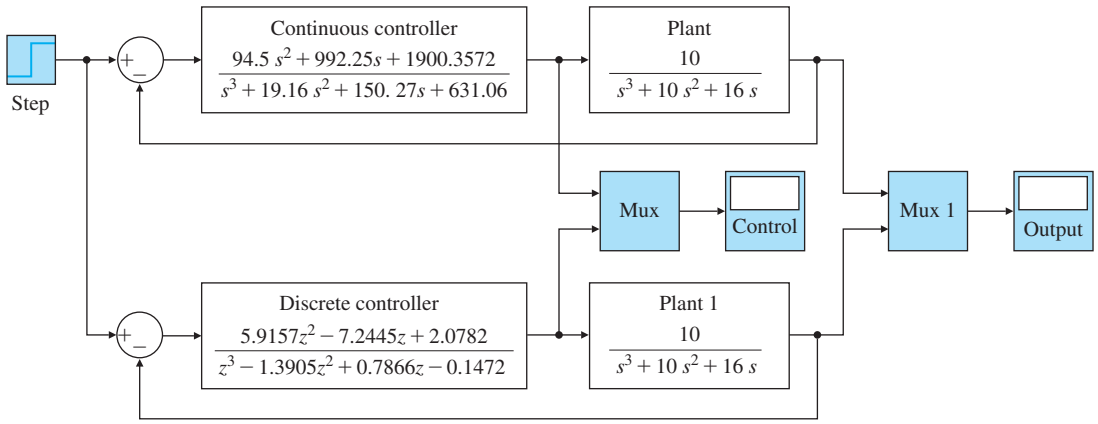
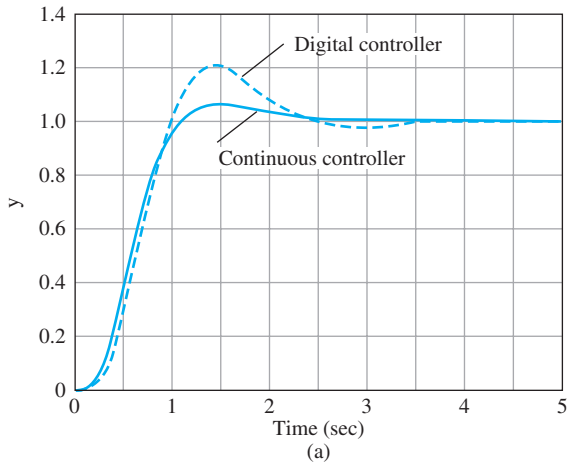


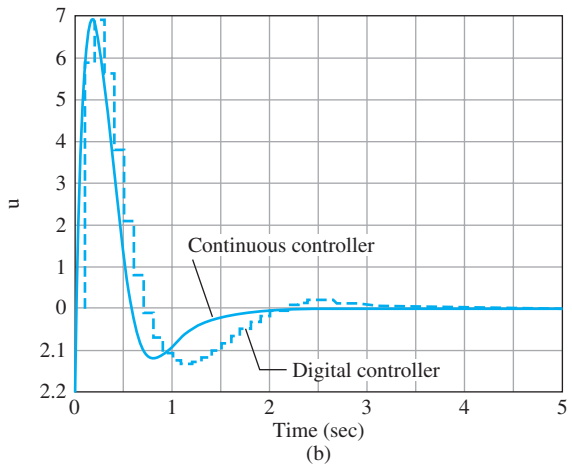
Figure W7.1

Simulink block diagram to compare continuous and discrete controllers

Figure W7.2

Comparison of step responses and control signals for continuous and discrete controllers: (a) step responses; (b)control signals





Appendix W7.9
Digital Implementation of Example 7.33

EXAMPLE W7.2 *Servomechanism*

For Example 7.33, derive an equivalent discrete controller with a sampling period of $T_s = 0.1$ sec ($20 \times \omega_n = 20 \times 0.05 = 0.1$ sec), and compare the continuous and digital control outputs, as well as the control efforts. Verify the design by plotting the step response and commenting on the comparison of the continuous and discrete responses.

Solution. The discrete equivalent for the controller is obtained from Matlab by using the c2d command, as in

Matlab c2d

```
nc=conv([1 1],[8.32 0.8]); % controller numerator
dc=conv([1 4.08],[1 0.0196]); % controller denominator
sysDc=tf(nc,dc); % form controller system description
ts=0.1; % sampling time of 0.1 sec
sysDd=c2d(sysDc,ts,'zoh'); % convert to discrete time controller
```

The discrete controller has the discrete transfer function

$$D_d(z) = \frac{8.32z^2 - 15.8855z + 7.5721}{z^2 - 1.6630z + 0.6637} = \frac{8.32(z - 0.9903)(z - 0.9191)}{(z - 0.998)(z - 0.6665)}.$$

The equation for the control law (with sample period suppressed for clarity) is

$$u(k+1) = 1.6630u(k) + 0.6637u(k-1) + 8.32e(k+1) - 15.8855e(k) + 7.5721e(k-1).$$

Simulink simulation

A Simulink diagram for simulating both the continuous and discrete systems is shown in Fig. W7.3. A comparison of the continuous and discrete step responses and control signals is shown in Fig. W7.4. Better agreement between the two responses can be achieved if the sampling period is reduced.

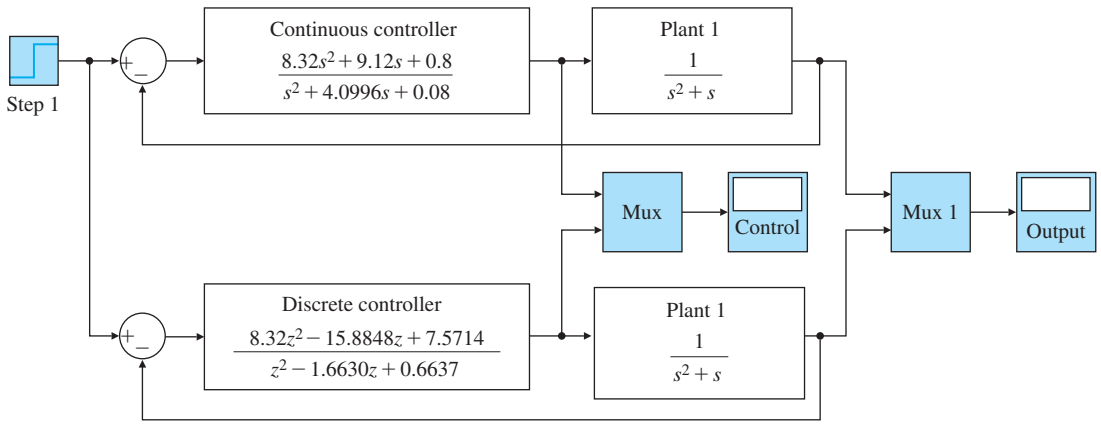
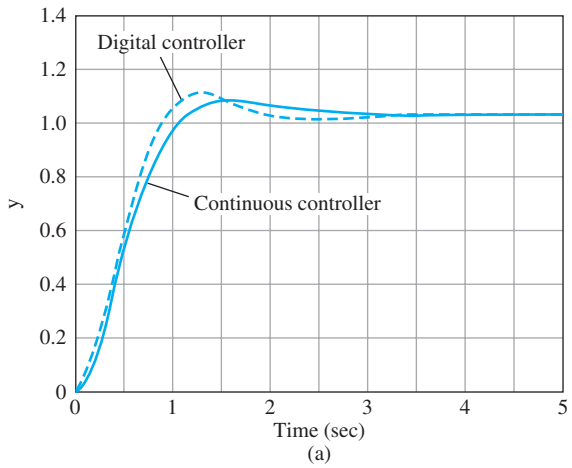


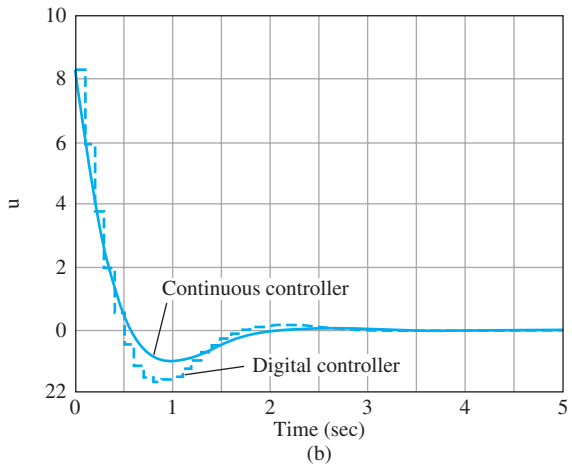
Figure W7.3

Simulink block diagram to compare continuous and discrete controllers

Figure W7.4

Comparison of step responses and control signals for continuous and discrete controllers: (a) step responses; (b) control signals





Appendix W7.14
Solution of State Equations

In this section, we consider the solution of state variable equations. This material is not necessary to understand the design of pole placement but will give a deeper insight into the method of state variables. It is instructive to consider first the unforced, or *homogenous*, system, which has the form

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}, \quad \mathbf{x}(\mathbf{0}) = \mathbf{x_0}. \tag{W7.1}$$

If the elements of $\mathbf{A}(t)$ are continuous functions of time, then the above equation has a *unique* solution for any initial state vector $\mathbf{x_0}$. There is a useful representation for the solution of this equation in terms of a matrix, called the *transition matrix*. Let $\phi_i(t, t_0)$ be the solution to the special initial condition

$$\mathbf{x}(\mathbf{0}) = \mathrm{e}_i = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i\text{th row} \tag{W7.2}$$

If $\mathbf{x}(t_0)$ is the actual initial condition at t_0, then we can express it in the decomposed form

$$\mathbf{x}(t_0) = \mathbf{x}_{01}\mathbf{e}_1 + \mathbf{x}_{02}\mathbf{e}_2 + \cdots + \mathbf{x}_{0n}\mathbf{e}_n. \tag{W7.3}$$

Because Eq. (W7.1) is linear, the state $\mathbf{x}(t)$ also can be expressed as a sum of the solutions to the special initial condition ϕ_i, as

$$\mathbf{x}(t) = \mathbf{x}_{01}\phi_1(t, t_0) + \mathbf{x}_{02}\phi_2(t, t_0) + \cdots + \mathbf{x}_{0n}\phi_n(t, t_0), \tag{W7.4}$$

or in matrix notation, as

$$\mathbf{x}(t) = \begin{bmatrix} \phi_1(t, t_0), & \phi_2(t, t_0), & \cdots, & \phi_n(t, t_0) \end{bmatrix} \begin{bmatrix} \mathbf{x}_{01} \\ \mathbf{x}_{02} \\ \vdots \\ \mathbf{x}_{0n} \end{bmatrix}. \tag{W7.5}$$

So we can define the *transition matrix*[1] to be

$$\mathbf{\Phi}(t, t_0) = \begin{bmatrix} \phi_1(t, t_0), & \phi_2(t, t_0), & \cdots, & \phi_n(t, t_0) \end{bmatrix}, \tag{W7.6}$$

and write the solution as

$$\mathbf{x}(t) = \mathbf{\Phi}(t, t_0)\mathbf{x}(t_0), \tag{W7.7}$$

[1] This is also referred to as the *fundamental matrix* of the differential equation.

where as the name implies, the transition matrix provides the transition between the state at time t_0 to the state at time t. Furthermore, from Eq. (W7.7), we have

$$\frac{d}{dt}\left[\mathbf{x}(t)\right] = \frac{d}{dt}\left[\mathbf{\Phi}(t,t_0)\right]\mathbf{x}(t_0), \tag{W7.8}$$

and from Eqs. (W7.1) and (W7.8), we have

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) = \mathbf{A}\mathbf{\Phi}(t,t_0)\mathbf{x}(t_0). \tag{W7.9}$$

Therefore

$$\frac{d}{dt}\left[\mathbf{\Phi}(t,t_0)\right] = \mathbf{A}\mathbf{\Phi}(t,t_0), \tag{W7.10}$$

and also

$$\mathbf{\Phi}(t,t) = \mathbf{I}. \tag{W7.11}$$

The transition matrix can be shown to have many interesting properties. Among them are the following:

1. $\mathbf{\Phi}(t_2,t_0) = \mathbf{\Phi}(t_2,t_1)\mathbf{\Phi}(t_1,t_0)$; (W7.12)
2. $\mathbf{\Phi}^{-1}(t,\tau) = \mathbf{\Phi}(\tau,t)$; (W7.13)
3. $\dfrac{d}{dt}\mathbf{\Phi}(t,\tau) = -\mathbf{\Phi}(t,\tau)\mathbf{A}(\tau)$; (W7.14)
4. $\det \mathbf{\Phi}(t,t_0) = e^{\int_{t_0}^{t} \mathrm{trace}\mathbf{A}(\tau)d\tau}$. (W7.15)

The second property implies that $\mathbf{\Phi}(t,\tau)$ is always invertible. What this means is that the solution is always unique so that, given a particular value of state at time τ, we can not only compute the future states from $\mathbf{\Phi}(t,\tau)$ but also past values $\mathbf{\Phi}^{-1}(t,\tau)$.

For the inhomogenous case, with a forcing function input $u(t)$, the equation is

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)u(t), \tag{W7.16}$$

and the solution is

$$\mathbf{x(t)} = \mathbf{\Phi}(t,t_0)\mathbf{x_0} + \int_{t_0}^{t} \mathbf{\Phi}(t,\tau)\mathbf{B}(\tau)u(\tau)d\tau. \tag{W7.17}$$

We can verify this by substituting the supposed solution, Eq. (W7.17), into the differential equation, Eq. (W7.16), as

$$\frac{d}{dt}\mathbf{x}(t) = \frac{d}{dt}\mathbf{\Phi}(t,t_0)\mathbf{x_0} + \frac{d}{dt}\int_{t_0}^{t} \mathbf{\Phi}(t,\tau)\mathbf{B}(\tau)u(\tau)d\tau. \tag{W7.18}$$

The second term from calculus (using the Leibnitz formula) is

$$\frac{d}{dt}\int_{t_0}^{t}\mathbf{\Phi}(t,\tau)\mathbf{B}(\tau)u(\tau)d\tau = \int_{t_0}^{t}\mathbf{A}(t)\mathbf{\Phi}(t,\tau)\mathbf{B}(\tau)u(\tau)d\tau + \mathbf{\Phi}(t,t)\mathbf{B}(t)u(t). \tag{W7.19}$$

Using the basic relations, we have

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}(t)\mathbf{\Phi}(t,t_0)\mathbf{x_0} + \mathbf{A}(t)\int_{t_0}^{t}\mathbf{\Phi}(t,\tau)\mathbf{B}(\tau)u(\tau)d\tau$$

$$+ \mathbf{B}(t)u(t), \tag{W7.20}$$

$$= \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)u(t), \tag{W7.21}$$

which shows that the proposed solution satisfies the system equation.
For the time-invariant case

$$\mathbf{\Phi}(t,t_0) = e^{\mathbf{A}(t-t_0)} = \mathbf{\Phi}(t-t_0), \tag{W7.22}$$

where

$$e^{\mathbf{A}t} = \left(\mathbf{I} + \mathbf{A}t + \frac{\mathbf{A}^2t^2}{2!} + \cdots\right) = \sum_{k=0}^{\infty}\frac{\mathbf{A}^kt^k}{k!}, \tag{W7.23}$$

is an invertible $n \times n$ exponential matrix, and by letting $t = 0$, we see that

$$e^{\mathbf{0}} = \mathbf{I}. \tag{W7.24}$$

The state solution is now

$$\mathbf{x}(t) = \mathbf{e}^{\mathbf{A}(t-t_0)}\mathbf{x_0} + \int_{t_0}^{t}\mathbf{e}^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau, \tag{W7.25}$$

and

$$y(t) = \mathbf{C}\mathbf{x}(t) = \mathbf{C}\mathbf{e}^{\mathbf{A}(t-t_0)}\mathbf{x_0} + \mathbf{C}\int_{t_0}^{t}\mathbf{e}^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau + Du(t). \tag{W7.26}$$

Suppose $\mathbf{x}(t_0) = \mathbf{x}_0 \equiv \mathbf{0}$, then the output is given by the convolution integral

$$y(t) = \int_{t_0}^{t}\mathbf{h}(t-\tau)\mathbf{B}u(\tau)d\tau, \tag{W7.27}$$

where $h(t)$ is the *impulse response*. In terms of the state variables matrices,

$$h(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{B} + D\delta(t). \tag{W7.28}$$

While there is no uniformly best way to compute the transition matrix, there are several methods that can be used to compute accurate approximations to it (See Moler, 2003; Franklin, Powell, and Workman, 1998). Three of these methods are matrix exponential series, inverse Laplace transform,

and diagonalization of the system matrix. In the first technique, we use Eq. (W7.23):

$$e^{\mathbf{A}t} \triangleq \mathbf{I} + \mathbf{A}t + \frac{\mathbf{A}^2t^2}{2!} + \frac{\mathbf{A}^3t^3}{3!} + \cdots, \tag{W7.29}$$

and the series should be computed in a reliable fashion. For the second method, we notice that if we define

$$\mathbf{\Phi}(s) = (s\mathbf{I} - \mathbf{A})^{-1}, \tag{W7.30}$$

then we can compute $\mathbf{\Phi}(s)$ from the $\mathbf{A}$ matrix and matrix algebra. Given this matrix, we can use the inverse Laplace Transform to compute

$$\mathbf{\Phi}(t) = \mathcal{L}^{-1}\left\{\mathbf{\Phi}(s)\right\}, \tag{W7.31}$$

$$= \mathcal{L}^{-1}\left\{(s\mathbf{I} - \mathbf{A})^{-1}\right\}. \tag{W7.32}$$

The last method we mentioned operates on the system matrix. If the system matrix can be diagonalized, that is, if we can find a transformation matrix $\mathbf{T}$ so that

$$\Lambda = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}, \tag{W7.33}$$

where $\mathbf{A}$ is reduced to the similar but diagonal matrix

$$\Lambda = \text{diag}\left\{\lambda_1, \lambda_2, \cdots, \lambda_n\right\}, \tag{W7.34}$$

then from the series, Eq. (W7.23), we see that we need only compute scalar exponentials, since

$$e^{\mathbf{A}t} = \mathbf{T}^{-1}\text{diag}\left\{e^{\lambda_1 t}, e^{\lambda_2 t}, \cdots, e^{\lambda_n t}\right\}\mathbf{T}. \tag{W7.35}$$

Appendix W8.7
Discrete State-Space Design Methods

We have seen in previous chapters that a linear, constant-coefficient continuous system can be represented by a set of first-order matrix differential equations of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u, \tag{W8.1}$$

where u is the control input to the system. The output equation can be expressed as

$$y = \mathbf{C}\mathbf{x} + Du. \tag{W8.2}$$

The solution to these equations (see Franklin et al., 1998) is

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)\,d\tau. \tag{W8.3}$$

It is possible to use Eq. (W8.3) to obtain a discrete state-space representation of the system. Because the solution over one sample period results in a difference equation, we can alter the notation a bit (letting $t = kT + T$ and $t_0 = kT$) to arrive at a particularly useful version of Eq. (W8.3):

$$\mathbf{x}(kT+T) = e^{\mathbf{A}T}\mathbf{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{A}(kT+T-\tau)}\mathbf{B}u(\tau)\,d\tau. \tag{W8.4}$$

This result is not dependent on the type of hold, because u is specified in terms of its continuous time history $u(\tau)$ over the sample interval. To find the discrete model of a continuous system where the input $u(t)$ is the output of a ZOH, we let $u(\tau)$ be a constant throughout the sample interval—that is,

$$u(\tau) = u(kT), kT \le \tau < kT + T.$$

To facilitate the solution of Eq. (W8.4) for a ZOH, we let

$$\eta = kT + T - \tau,$$

which converts Eq. (W8.4) to

$$\mathbf{x}(kT+T) = e^{\mathbf{A}T}\mathbf{x}(kT) + \left(\int_0^T e^{\mathbf{A}\eta}\,d\eta\right)\mathbf{B}u(kT).$$

If we let

$$\Phi = e^{\mathbf{A}T}$$

and

$$\mathbf{\Gamma} = \left(\int_0^T e^{\mathbf{A}\eta}\,d\eta\right)\mathbf{B}, \tag{W8.5}$$

Difference equations in standard form

Eqs. (W8.4) and (W8.2) reduce to difference equations in standard form:

$$\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}u(k), \tag{W8.6}$$

$$y(k) = \mathbf{C}\mathbf{x}(k) + Du(k). \tag{W8.7}$$

Here $\mathbf{x}(k+1)$ is a shorthand notation for $\mathbf{x}(kT+T)$, $\mathbf{x}(k)$ for $\mathbf{x}(kT)$, and $u(k)$ for $u(kT)$. The series expansion

$$\mathbf{\Phi} = e^{\mathbf{A}T} = \mathbf{I} + \mathbf{A}T + \frac{\mathbf{A}^2T^2}{2!} + \frac{\mathbf{A}^3T^3}{3!} + \cdots$$

also can be written as

$$\mathbf{\Phi} = \mathbf{I} + \mathbf{A}T\mathbf{\Psi}, \tag{W8.8}$$

where

$$\mathbf{\Psi} = \mathbf{I} + \frac{\mathbf{A}T}{2!} + \frac{\mathbf{A}^2T^2}{3!} + \cdots.$$

The $\mathbf{\Gamma}$ integral in Eq. (W8.5) can be evaluated term by term to give

$$\begin{aligned}\mathbf{\Gamma} &= \sum_{k=0}^{\infty} \frac{\mathbf{A}^k T^{k+1}}{(k+1)!}\mathbf{B} \\ &= \sum_{k=0}^{\infty} \frac{\mathbf{A}^k T^k}{(k+1)!}T\mathbf{B} \\ &= \mathbf{\Psi}T\mathbf{B}.\end{aligned} \tag{W8.9}$$

We evaluate $\mathbf{\Psi}$ by a series in the form

$$\mathbf{\Psi} \cong \mathbf{I} + \frac{\mathbf{A}T}{2}\left\{\mathbf{I} + \frac{\mathbf{A}T}{3}\left[\mathbf{I} + \cdots \frac{\mathbf{A}T}{N-1}\left(\mathbf{I} + \frac{\mathbf{A}T}{N}\right)\right]\right\},$$

which has better numerical properties than the direct series. We then find $\mathbf{\Gamma}$ from Eq. (W8.9) and $\mathbf{\Phi}$ from Eq. (W8.8). For a discussion of various methods of numerical determination of $\mathbf{\Phi}$ and $\mathbf{\Gamma}$, see Franklin et al. (1998) and Moler and van Loan (1978, 2003). The evaluation of the $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ matrices in practice is carried out by the `c2d` function in Matlab.

Matlab c2d

To compare this method of representing the plant with the discrete transfer function, we can take the z-transform of Eqs. (W8.6) and (W8.7) with $J = 0$ to obtain

$$(z\mathbf{I} - \mathbf{\Phi})\mathbf{X}(z) = \mathbf{\Gamma}U(z), \tag{W8.10}$$

$$Y(z) = \mathbf{C}\mathbf{X}(z). \tag{W8.11}$$

Therefore,

$$\frac{Y(z)}{U(z)} = G(z) = \mathbf{C}(z\mathbf{I} - \mathbf{\Phi})^{-1}\mathbf{\Gamma}. \tag{W8.12}$$

EXAMPLE W8.1 *Discrete State-Space Representation of $1/s^2$ Plant*

Use the relation in this section to verify that the discrete model of the $1/s^2$ plant preceded by a ZOH is that given in the solution to Example 8.4.

Solution. The $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ matrices can be calculated using Eqs. (W8.8) and (W8.9). Example 7.1 (with $I = 1$) showed that the values for $\mathbf{A}$ and $\mathbf{B}$ are

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Because $\mathbf{A}^2 = \mathbf{0}$ in this case, we have

$$\begin{aligned} \mathbf{\Phi} &= \mathbf{I} + \mathbf{A}T + \frac{\mathbf{A}^2T^2}{2!} + \cdots \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} T = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \\ \mathbf{\Gamma} &= \left(\mathbf{I} + \mathbf{A}\frac{T}{2!}\right) T\mathbf{B} \\ &= \left(\begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \frac{T^2}{2}\right) \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}. \end{aligned}$$

Hence, using Eq. (W8.12), we obtain

$$\begin{aligned} G(z) = \frac{Y(z)}{U(z)} &= [1 \quad 0] \left(z \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}\right)^{-1} \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \\ &= \frac{T^2}{2} \left[\frac{z+1}{(z-1)^2}\right]. \end{aligned} \tag{W8.13}$$

This is the same result we obtained using Eq. (8.33) and the z-transform tables in Example 8.4.

Note that to compute Y/U, we find that the denominator of Eq. (W8.13) is $\det(z\mathbf{I} - \mathbf{\Phi})$, which was created by the matrix inverse in Eq. (W8.12). This determinant is the characteristic polynomial of the transfer function, and the zeros of the determinant are the poles of the plant. We have two poles at $z = 1$ in this case, corresponding to two integrations in this plant's equations of motion.

We can further explore the question of poles and zeros and the state-space description by considering again the transform formulas [Eqs. (W8.10) and (W8.11)]. One way to interpret transfer-function poles from the perspective of the corresponding difference equation is that a pole is a value of z such that the equation has a nontrivial solution when the forcing input is zero. From Eq. (W8.10), this interpretation implies that the linear equations

$$(z\mathbf{I} - \mathbf{\Phi})\mathbf{X}(z) = \mathbf{0}$$

have a nontrivial solution. From matrix algebra the well-known requirement for a nontrivial solution is that $\det(z\mathbf{I} - \mathbf{\Phi}) = 0$. Using the system in Example W8.1, we get

$$\begin{aligned}\det(z\mathbf{I} - \mathbf{\Phi}) &= \det\left(\begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}\right) \\ &= \det\begin{bmatrix} z-1 & -T \\ 0 & z-1 \end{bmatrix} \\ &= (z-1)^2 = 0,\end{aligned}$$

which is the characteristic equation, as we have seen. In Matlab, the poles of the system are found by `P = eig(Phi)`.

Along the same line of reasoning, a system zero is a value of z such that the system output is zero even with a nonzero state-and-input combination. Thus, if we are able to find a nontrivial solution for $\mathbf{X}(z_0)$ and $U(z_0)$ such that $Y(z_0)$ is identically zero, then z_0 is a zero of the system. In combining Eqs. (W8.10) and (W8.11), we must satisfy the requirement that

$$\begin{bmatrix} z\mathbf{I} - \mathbf{\Phi} & -\mathbf{\Gamma} \\ \mathbf{C} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{X}(z) \\ U(z) \end{bmatrix} = \mathbf{0}.$$

Once more the condition for the existence of nontrivial solutions is that the determinant of the square coefficient system matrix be zero. For Example W8.1, the calculation is

$$\begin{aligned}\det\begin{bmatrix} z-1 & -T & -T^2/2 \\ 0 & z-1 & -T \\ 1 & 0 & 0 \end{bmatrix} &= \det\begin{bmatrix} -T & -T^2/2 \\ z-1 & -T \end{bmatrix} \\ &= T^2 + \frac{T^2}{2}(z-1) \\ &= \frac{T^2}{2}z + \frac{T^2}{2} \\ &= \frac{T^2}{2}(z+1).\end{aligned}$$

Thus we have a single zero at $z = -1$, as we have seen from the transfer function. In Matlab, the zeros are found by `Z=tzero(Phi,Gam,C,D)`.

Much of the algebra for discrete state-space control design is the same as for the continuous-time case discussed in Chapter 7. The poles of a discrete system can be moved to desirable locations by linear state-variable feedback

$$u = -\mathbf{K}\mathbf{x}$$

such that

$$\det(z\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K}) = \alpha_c(z), \tag{W8.14}$$

provided that the system is controllable. The system is controllable if the controllability matrix

$$C_{\mathbf{X}} = \begin{bmatrix} \mathbf{\Gamma} & \mathbf{\Phi}\mathbf{\Gamma} & \mathbf{\Phi}^2\mathbf{\Gamma} & \dots & \mathbf{\Phi}^{n-1}\mathbf{\Gamma} \end{bmatrix}$$

is full rank.

A discrete full-order estimator has the form

$$\bar{\mathbf{x}}(k+1) = \mathbf{\Phi}\bar{\mathbf{x}}(k) + \mathbf{\Gamma}u(k) + \mathbf{L}[y(k) - \mathbf{C}\bar{\mathbf{x}}(k)],$$

where $\bar{\mathbf{x}}$ is the state estimate. The error equation,

$$\tilde{\mathbf{x}}(k+1) = (\mathbf{\Phi} - \mathbf{LC})\tilde{\mathbf{x}}(k),$$

can be given arbitrary dynamics $\alpha_e(z)$, provided that the system is observable, which requires that the observability matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C\Phi} \\ \mathbf{C\Phi}^2 \\ \vdots \\ \mathbf{C\Phi}^{n-1} \end{bmatrix}$$

be full rank.

As was true for the continuous-time case, if the open-loop transfer function is

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b(z)}{a(z)},$$

then a state-space compensator can be designed such that

$$\frac{Y(z)}{R(z)} = \frac{K_s\gamma(z)b(z)}{\alpha_c(z)\alpha_e(z)},$$

where r is the reference input. The polynomials $\alpha_c(z)$ and $\alpha_e(z)$ are selected by the designer using exactly the same methods discussed in Chapter 7 for continuous systems. $\alpha_c(z)$ results in a control gain $\mathbf{K}$ such that $\det(z\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma K}) = \alpha_c(z)$, and $\alpha_e(z)$ results in an estimator gain $\mathbf{L}$ such that $\det(z\mathbf{I} - \mathbf{\Phi} + \mathbf{LC}) = \alpha_e(z)$. If the estimator is structured according to Fig. 7.48a, the system zeros $\gamma(z)$ will be identical to the estimator poles $\alpha_e(z)$, thus removing the estimator response from the closed-loop system response. However, if desired, we can arbitrarily select the polynomial $\gamma(z)$ by providing suitable feed-forward from the reference input. Refer to Franklin et al. (1998) for details.

EXAMPLE W8.2 *State-Space Design of a Digital Controller*

Design a digital controller for a $1/s^2$ plant to meet the specifications given in Example 8.2. Use state-space design methods, including the use of an estimator, and structure the reference input in two ways: (a) Use the error command shown in Fig. 7.48b, and (b) use the state command shown in Fig. 7.15 and Fig. 7.48a.

Solution. We find the state-space model of the $1/s^2$ plant preceded by a ZOH using the Matlab statements

```
sysSSc = ss([0 1;0 0], [0; 1], [1 0], 0];
T = 1;
sysSSd = c2d(sysSSc, T);
[Phi,Gam,H] = ssdata(sysSSd);
```

Using discrete analysis for Example 8.4, we find that the desired z-plane roots are at $z = 0.78 \pm 0.18j$. Solving the discrete pole-placement problem involves placing the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma K}$, as indicated by Eq. (W8.14). Likewise, the solution of the continuous pole-placement problem involves placing the eigenvalues of $\mathbf{A} - \mathbf{BK}$, as indicated by Eq. (7.72). Because these two tasks are identical, we use the same function in Matlab for the continuous and discrete cases. Therefore, the control feedback matrix $\mathbf{K}$ is found by

```
pc = [0.78 + 0.18*j; 0.78 - 0.18*j];
K = acker(Phi,Gam,pc);
```

which yields

$$\mathbf{K} = [0.0808 \; 0.3996].$$

To ensure that the estimator roots are substantially faster than the control roots (so that the estimator roots will have little effect on the output), we choose them to be at $z = 0.2 \pm 0.2j$. Therefore, the estimator feedback matrix $\mathbf{L}$ is found by

```
pe = [0.2 + 0.2*j; 0.2 - 0.2*j];
L = acker(Phi, C, pe);
```

which yields

$$\mathbf{L} = \begin{bmatrix} 1.6 \\ 0.68 \end{bmatrix}.$$

The equations of the compensation for $r = 0$ (regulation to $\mathbf{x}^T = [0 \quad 0]$) are then

$$\bar{\mathbf{x}}(k+1) = \mathbf{\Phi}\bar{\mathbf{x}}(k) + \mathbf{\Gamma}u(k) + \mathbf{L}[y(k) - \mathbf{C}\bar{\mathbf{x}}(k)], \tag{W8.15}$$

$$u(k) = -\mathbf{K}\bar{\mathbf{x}}(k). \tag{W8.16}$$

1. For the error command structure where the compensator is placed in the feed-forward path, as shown in Fig. 7.48b in the book, $y(k)$ from Eq. (W8.15) is replaced with $y(k) - r$, so the state description of the plant plus the estimator (a fourth-order system whose state vector is $[\mathbf{x}\,\bar{\mathbf{x}}]^T$) is

```
A = [Phi    - Gam*K; L*C   Phi - Gam*K - L*C];
B = [0; 0; -L];
C = [1 0 0 0];
D = 0;
step(A,B,C,D).
```

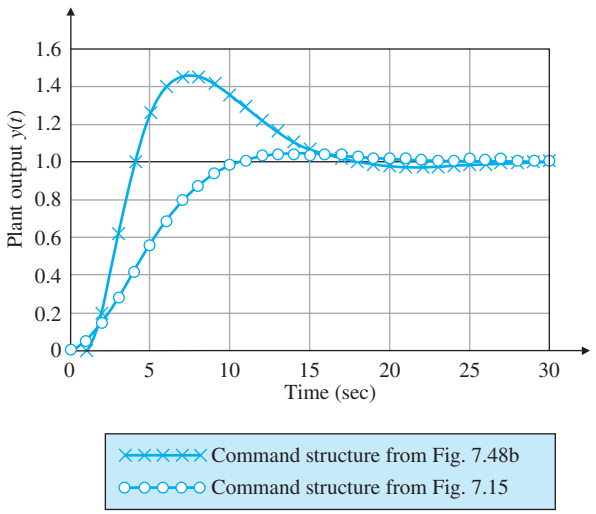


Figure W8.1
Step response of Example W8.2

The resulting step response in Fig. W8.1 shows a response similar to that of the step responses in Fig. 8.19 in the book.

2. For the state command structure described in Section 7.9 in the book, we wish to command the position element of the state vector so that

$$\mathbf{N_x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

and the $1/s^2$ plant requires no steady control input for a constant output y. Therefore $N_u = 0$. To analyze a system with this command structure, we need to modify matrix **B** from the preceding Matlab statement to properly introduce the reference input r according to Fig.7.15. The Matlab statement

```
B = [Gam*K*Nx; Gam*K*Nx];
```

channels r into both the plant and estimator equally, thus not exciting the estimator dynamics. The resulting step response in Fig. W8.1 shows a substantial reduction in the overshoot with this structure. In fact, the overshoot is now about 5%, which is expected for a second-order system with $\zeta \cong 0.7$. The previous designs all had considerably greater overshoot because of the effect of the extra zero and pole.

SUMMARY OF STATE-SPACE DESIGN

- The continuous state-space form of a differential equation,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u,$$
$$y = \mathbf{C}\mathbf{x} + Du,$$

has a discrete counterpart in the difference equations

$$\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}u(k),$$
$$y(k) = \mathbf{C}x(k) + Du(k),$$

where

$$\mathbf{\Phi} = e^{\mathbf{A}T}$$
$$\mathbf{\Gamma} = \left(\int_0^T e^{\mathbf{A}\eta}\, d\eta\right)\mathbf{B}.$$

These matrices can be computed in Matlab by `[Phi, Gam] = c2d(F,G,H,J)` and used in state-space **discrete design** methods.

- The pole placement and estimation ideas are identical in the continuous and discrete domains.

PROBLEMS

W8.1 In Problem 8.11 in the book, we dealt with an experiment in magnetic levitation described by Eq. (8.62) that reduces to

$$\ddot{x} = 1000x + 20i.$$

Let the sampling time be 0.01 sec.

(a) Use pole placement to design a controller for the magnetic levitator so that the closed-loop system meets the following specifications: settling time, $t_s \leq 0.25$ sec, and overshoot to an initial offset in x that is less than 20%.

(b) Plot the step response of $x, \tilde{x}$, and i to an initial displacement in x.

(c) Plot the root locus for changes in the plant gain, and mark the pole locations of your design.

(d) Introduce a command reference input r (as discussed in Section 7.9) that does not excite the estimate of x. Measure or compute the frequency response from r to the system error $r - x$ and give the highest frequency for which the error amplitude is less than 20% of the command amplitude.

W8.2 *Servomechanism for Antenna Elevation Control*: Suppose it is desired to control the elevation of an antenna designed to track a satellite. A photo of such a system is shown in Fig. W8.2, and a schematic diagram is depicted in Fig. W8.3. The antenna and drive parts have a moment of inertia J and damping B, arising to some extent from bearing and aerodynamic friction, but mostly from the back emf of the DC drive motor. The equation of motion is

$$J\ddot{\theta} + B\dot{\theta} = T_c + T_d,$$

where

$$T_c = \text{net torque from the drive motor,} \tag{W8.17}$$
$$T_d = \text{disturbance torque due to wind.} \tag{W8.18}$$

Figure W8.2
Satellite-tracking antenna *(Courtesy Space Systems/Loral)*

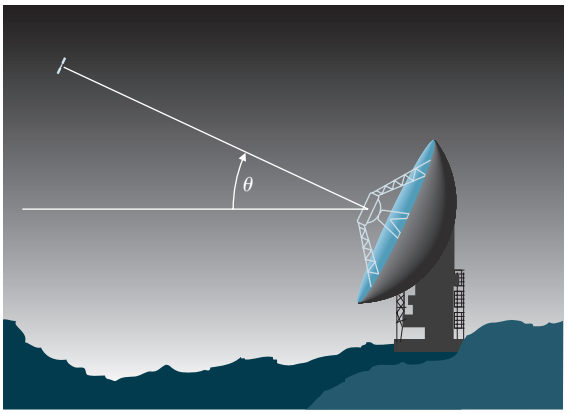


Figure W8.3
Schematic diagram of satellite-tracking antenna

If we define

$$\frac{B}{J} = a, \quad u = \frac{T_c}{B}, \quad \text{and} \quad w_d = \frac{T_d}{B},$$

the equation simplifies to

$$\frac{1}{a}\ddot{\theta} + \dot{\theta} = u + w_d.$$

After using the Laplace transformation, we obtain

$$\theta(s) = \frac{1}{s(s/a+1)}[u(s) + w_d(s)],$$

or with no disturbance,

$$\frac{\theta(s)}{u(s)} = \frac{1}{s(s/a+1)} = G_2(s).$$

With $u(k)$ applied through a ZOH, the transfer function for an equivalent discrete-time system is

$$G_2(z) = \frac{\theta(z)}{u(z)} = K\frac{z+b}{(z-1)(z-e^{-aT})},$$

where

$$K = \frac{aT - 1 + e^{-aT}}{a}, \quad b = \frac{1 - e^{-aT} - aTe^{-aT}}{aT - 1 + e^{-aT}}.$$

(a) Let $a = 0.1$ and $x_1 = \dot{\theta}$, and write the continuous-time state equations for the system.

(b) Let $T = 1$ sec, and find a state feedback gain $\mathbf{K}$ for the equivalent discrete-time system that yields closed-loop poles corresponding to the following points in the s-plane: $s = -1/2 \pm j\left(\sqrt{\frac{3}{2}}\right)$. Plot the step response of the resulting design.

(c) Design an estimator: Select $\mathbf{L}$ so that $\alpha_e(z) = z^2$.

(d) Using the values for $\mathbf{K}$ and $\mathbf{L}$ computed in parts (b) and (c) as the gains for a combined estimator/controller, introduce a reference input that will leave the state estimate undisturbed. Plot the response of the closed-loop system due to a step change in the reference input. Also plot the system response to a step wind-gust disturbance.

(e) Plot the root locus of the closed-loop system with respect to the plant gain, and mark the locations of the closed-loop poles.

W8.3 *Tank Fluid Temperature Control*: The temperature of a tank of fluid with a constant inflow and outflow rate is to be controlled by adjusting the temperature of the incoming fluid. The temperature of the incoming fluid is controlled by a mixing valve that adjusts the relative amounts of hot and cold supplies of the fluid (see Fig. W8.4). The distance between the valve and the point of discharge into the tank creates a time delay between the application of a temperature change at the mixing valve and the discharge of the flow with the changed temperature into the tank. The differential equation governing the tank temperature is

$$\dot{T}_e = \frac{1}{cM}(q_{\text{in}} - q_{\text{out}}),$$

where

T_e = tank temperature,

c = specific heat of the fluid,

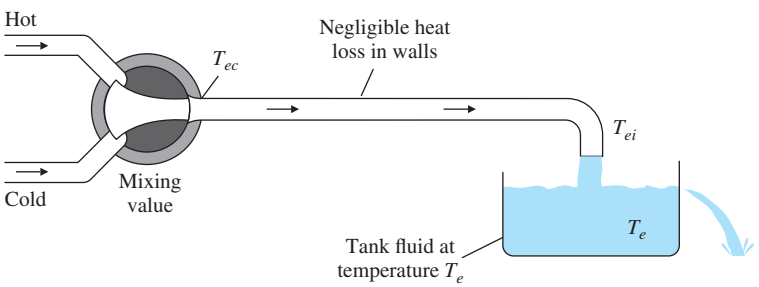


Figure W8.4
Tank temperature control

M = fluid mass contained in the tank,

$q_{\text{in}} = c\dot{m}_{\text{in}}T_{ei}$,

$q_{\text{out}} = c\dot{m}_{\text{out}}T_e$,

$\dot{m}$ = mass flow rate ($\dot{m}_{\text{in}} = \dot{m}_{\text{out}}$),

T_{ei} = temperature of fluid entering tank.

However, the temperature at the input to the tank at time t is equal to the control temperature τ_d seconds in the past. This relationship may be expressed as

$$T_{ei}(t) = T_{ec}(t - \tau_d),$$

where

$\tau_d =$ delay time,

$T_{ec} =$ temperature of fluid immediately after the control valve and directly controllable by the valve.

Combining constants, we obtain

$$\dot{T}_e(t) + aT_e(t) = aT_{ec}(t - \tau_d),$$

where

$$a = \frac{\dot{m}}{M}.$$

The transfer function of the system is thus

$$\frac{T_e(s)}{T_{ec}(s)} = \frac{e^{-\tau ds}}{s/a + 1} = G_3(s).$$

To form a discrete transfer function equivalent to G_3 preceded by a ZOH, we must compute

$$G_3(z) = \mathcal{Z}\left\{\left(\frac{1 - e^{-sT}}{s}\right)\left(\frac{e^{-\tau_d s}}{s/a + 1}\right)\right\}.$$

We assume that for some integer l, $\tau_d = lT - mT$, where $0 \leq m < 1$. Then

$$\begin{aligned}
G_3(z) &= \mathcal{Z}\left\{\left(\frac{1 - e^{-sT}}{s}\right)\left(\frac{e^{-lsT}e^{msT}}{s/a + 1}\right)\right\} \\
&= (1 - z^{-1})z^{-l}\mathcal{Z}\left\{\frac{e^{msT}}{s(s/a + 1)}\right\} \\
&= (1 - z^{-1})z^{-l}\mathcal{Z}\left\{\frac{e^{msT}}{s} - \frac{e^{msT}}{s + a}\right\} \\
&= \frac{z - 1}{z}\left(\frac{1}{z^l}\right)\mathcal{Z}\{1(t + mT) - e^{-a(t+mT)}1(t + mT)\} \\
&= \frac{z - 1}{z}\left(\frac{1}{z^l}\right)\left(\frac{z}{z - 1} - \frac{e^{-amT}z}{z - e^{-aT}}\right)
\end{aligned}$$

$$= \frac{1}{z^l}\left[\frac{(1-e^{-amT})z + e^{-amT} - e^{-aT}}{z - e^{-aT}}\right]$$

$$= \left(\frac{1-e^{-amT}}{z^l}\right)\left(\frac{z+\alpha}{z-e^{-aT}}\right),$$

and

$$\alpha = \frac{e^{-amT} - e^{-aT}}{1 - e^{-amT}}.$$

The zero location $-\alpha$ varies from $\alpha = \infty$ at $m = 0$ to $\alpha = 0$ as $m \to 1$. Note also that $G_3(1) = 1.0$ for all a, m, and l. For the specific values $\tau_d = 1.5$, $T = 1$, $a = 1$, $l = 2$, and $m = \frac{1}{2}$, the transfer function reduces to

$$G_3(z) = 0.3935\frac{z + 0.6065}{z^2(z - 0.3679)}.$$

(a) Write the discrete-time system equations in state-space form.

(b) Design a state feedback gain that yields $\alpha_c(z) = z^3$.

(c) Design a state estimator with $\alpha_e(z) = z^3$.

(d) Plot the root locus of the system with respect to the plant gain.

(e) Plot the step response of the system.

W8.4 Consider the linear equation $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is an $n \times n$ matrix. When $\mathbf{b}$ is given, one way of solving for $\mathbf{x}$ is to use the discrete-time recursion

$$\mathbf{x}(k+1) = (\mathbf{I} + c\mathbf{A})\mathbf{x}(k) - c\mathbf{b},$$

where c is a scalar to be chosen.

(a) Show that the solution of $\mathbf{Ax} = \mathbf{b}$ is the equilibrium point $\mathbf{x}^*$ of the discrete-time system. An equilibrium point $\mathbf{x}^*$ of a discrete-time system $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k))$ satisfies the relation $\mathbf{x}^* = \mathbf{f}(\mathbf{x}^*)$.

(b) Consider the error $\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{x}^*$. Write the linear equation that relates the error $\mathbf{e}(k+1)$ to $\mathbf{e}(k)$.

(c) Suppose $|1 + c\lambda_i(\mathbf{A})| < 1, i = 1, \ldots, n$, where $\lambda_i(\mathbf{A})$ denotes the ith eigenvalue of $\mathbf{A}$. Show that, starting from any initial guess $\mathbf{x}_0$, the algorithm converges to $\mathbf{x}^*$. [*Hint:* For any matrix $\mathbf{B}$, $\lambda_i(\mathbf{I}+\mathbf{B}) = 1+\lambda_i(\mathbf{B})$.]

W8.5 The open-loop plant of a unity feedback system has the transfer function

$$G(s) = \frac{1}{s(s+2)}.$$

Determine the transfer function of the equivalent digital plant using a sampling period of $T = 1$ sec, and design a proportional controller for the discrete-time system that yields dominant closed-loop poles with a damping ratio ζ of 0.7.

W8.6 Write a computer program to compute $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ from $\mathbf{A}$, $\mathbf{B}$, and the sample period T. It is okay to use Matlab, but don't use C2D. Write code in Matlab to compute the discrete matrices using the relations developed in this chapter. Use your program to compute $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ when

(a)

$$\mathbf{A} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad T = 0.2 \text{ sec},$$

(b)

$$\mathbf{A} = \begin{bmatrix} -3 & -2 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad T = 0.2\text{sec}.$$

W8.7 Consider the following discrete-time system in state-space form:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 10 \end{bmatrix} u(k).$$

Use state feedback to relocate all of the system's poles to 0.5.

W8.8 Let

$$\mathbf{\Phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{\Gamma} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}.$$

(a) Find a transformation matrix $\mathbf{T}$ so that, if $\mathbf{x} = \mathbf{T}\mathbf{w}$, the state equations for $\mathbf{w}$ will be in control canonical form.

(b) Compute the gain $\mathbf{K}_w$ so that, if $u = -\mathbf{K}_w\mathbf{w}$, the characteristic equation will be $\alpha_c(z) = z^2 - 1.6z + 0.7$.

(c) Use $\mathbf{T}$ from part (a) to compute $\mathbf{K}_x$, which is the feedback gain required by the state equations in $\mathbf{x}$ to achieve the desired characteristic polynomial.

W8.9 Consider a system whose plant transfer function is $1/s^2$ and has a piecewise constant input of the form

$$u(t) = u(kT), \quad kT \le t < (k+1)T.$$

(a) Show that, if we restrict attention to the time instants kT, $k = 0, 1, 2, \ldots$, the resulting sampled-data system can be described by the equations

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ T & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} T \\ T^2/2 \end{bmatrix} u(k).$$

$$y(k) = [0 \quad 1][x_1(k)\ x_2(k)]^T.$$

(b) Design a second-order estimator that will always drive the error in the estimate of the initial state vector to zero in time $2T$ or less.

(c) Is it possible to estimate the initial state exactly with a first-order estimator? Justify your answer.

W8.10 In this problem, you will show how to compute $\mathbf{\Phi}$ by changing states so that the system matrix is diagonal.

(a) Using an infinite series expansion, compute $e^{\mathbf{A}T}$ for

$$\mathbf{A} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}.$$

(b) Show that if $\overline{\mathbf{A}} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}$ for some nonsingular transformation matrix $\mathbf{T}$, then

$$e^{\overline{\mathbf{A}}T} = \mathbf{T}e^{\mathbf{A}T}\mathbf{T}^{-1}.$$

(c) Show that if

$$\overline{\mathbf{A}} = \begin{bmatrix} -3 & 1 \\ -2 & 0 \end{bmatrix},$$

there exists a $\mathbf{T}$ such that $\mathbf{T}\mathbf{A}\mathbf{T}^{-1} = \overline{\mathbf{A}}$. (*Hint*: Write $\mathbf{T}\mathbf{A} = \overline{\mathbf{A}}\mathbf{T}$, assume four unknowns for the elements of $\mathbf{T}$, and solve. Next show that the columns of $\mathbf{T}$ are the eigenvectors of $\overline{\mathbf{A}}$.)

(d) Compute $e^{\overline{\mathbf{A}}T}$.