Control Engineering Series

Robust Control System Design

Advanced State Space Techniques

Second Edition, Revised and Expanded

u(t)
G(s) = C(sI-A)^-1 B
y(t)
K_y
K_z
z(t)
(sI-F)^-1
L

Chia-Chi Tsui

Robust Control System Design

Advanced State Space Techniques

Second Edition, Revised and Expanded

Chia-Chi Tsui
DeVry Institute of Technology
Long Island City, New York, U.S.A.

Although great care has been taken to provide accurate and current information, neither the author(s) nor the publisher, nor anyone else associated with this publication, shall be liable for any loss, damage, or liability directly or indirectly caused or alleged to be caused by this book. The material contained herein is not intended to provide specific advice or recommendations for any specific situation.

Trademark notice: Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress.

ISBN: 0-8247-4869-7

This book is printed on acid-free paper.

Headquarters
Marcel Dekker, Inc., 270 Madison Avenue, New York, NY 10016, U.S.A.
tel: 212-696-9000; fax: 212-685-4540

Distribution and Customer Service
Marcel Dekker, Inc., Cimarron Road, Monticello, New York 12701, U.S.A.
tel: 800-228-1160; fax: 845-796-1772

Eastern Hemisphere Distribution
Marcel Dekker AG, Hutgasse 4, Postfach 812, CH-4001 Basel, Switzerland
tel: 41-61-260-6300; fax: 41-61-260-6333

World Wide Web
http://www.dekker.com

The publisher offers discounts on this book when ordered in bulk quantities. For more information, write to Special Sales/Professional Marketing at the headquarters address above.

Current printing (last digit):

10 9 8 7 6 5 4 3 2 1

PRINTED IN THE UNITED STATES OF AMERICA

CONTROL ENGINEERING

A Series of Reference Books and Textbooks

1. Nonlinear Control of Electric Machinery, *Darren M. Dawson, Jun Hu, and Timothy C. Burg*
2. Computational Intelligence in Control Engineering, *Robert E. King*
3. Quantitative Feedback Theory: Fundamentals and Applications, *Constantine H. Houpis and Steven J. Rasmussen*
4. Self-Learning Control of Finite Markov Chains, *A. S. Poznyak, K. Najim, and E. Gómez-Ramírez*
5. Robust Control and Filtering for Time-Delay Systems, *Magdi S. Mahmoud*
6. Classical Feedback Control: With MATLAB, *Boris J. Lurie and Paul J. Enright*
7. Optimal Control of Singularly Perturbed Linear Systems and Applications: High-Accuracy Techniques, *Zoran Gajić and Myo-Taeg Lim*
8. Engineering System Dynamics: A Unified Graph-Centered Approach, *Forbes T. Brown*
9. Advanced Process Identification and Control, *Enso Ikonen and Kaddour Najim*
10. Modern Control Engineering, *P. N. Paraskevopoulos*
11. Sliding Mode Control in Engineering, *edited by Wilfrid Perruquetti and Jean Pierre Barbot*
12. Actuator Saturation Control, *edited by Vikram Kapila and Karolos M. Grigoriadis*

13. Nonlinear Control Systems, *Zoran Vukić, Ljubomir Kuljača, Dali Đonlagić, Sejid Tešnjak*
14. Linear Control System Analysis and Design with MATLAB: Fifth Edition, Revised and Expanded, *John J. D'Azzo, Constantine H. Houpis, and Stuart N. Sheldon*
15. Robot Manipulator Control: Theory and Practice, Second Edition, Revised and Expanded, *Frank L. Lewis, Darren M. Dawson, and Chaouki T. Abdallah*
16. Robust Control System Design: Advanced State Space Techniques, Second Edition, Revised and Expanded, *Chia-Chi Tsui*

Additional Volumes in Preparation

To Susan
and
James and Shane

Series Introduction

Many textbooks have been written on control engineering, describing new techniques for controlling systems, or new and better ways of mathematically formulating existing methods to solve the ever-increasing complex problems faced by practicing engineers. However, few of these books fully address the applications aspects of control engineering. It is the intention of this new series to redress this situation.

The series will stress applications issues, and not just the mathematics of control engineering. It will provide texts that present not only both new and well-established techniques, but also detailed examples of the application of these methods to the solution of real-world problems. The authors will be drawn from both the academic world and the relevant applications sectors.

There are already many exciting examples of the application of control techniques in the established fields of electrical, mechanical (including aerospace), and chemical engineering. We have only to look around in today's highly automated society to see the use of advanced robotics techniques in the manufacturing industries; the use of automated control and navigation systems in air and surface transport systems; the increasing use of intelligent control systems in the many artifacts available to the domestic consumer market; and the reliable supply of water, gas, and electrical power to the domestic consumer and to industry. However, there are currently many challenging problems that could benefit from wider exposure to the applicability of control methodologies, and the systematic systems-oriented basis inherent in the application of control techniques.

This series presents books that draw on expertise from both the academic world and the applications domains, and will be useful not only as academically recommended course texts but also as handbooks for practitioners in many applications domains. *Robust Control Systems* is another outstanding entry in Dekker's Control Engineering series.

Preface

This second edition of *Robust Control System Design* introduces a new design approach to modern control systems. This design approach guarantees, for the first time, the full realization of robustness properties of generalized state feedback control for most open-loop system conditions. State and generalized state feedback control can achieve feedback system performance and robustness far more effectively than other basic forms of control. Performance and robustness (versus model uncertainty and control disturbance) are mutually contradictory, yet they are the key properties required by practical control systems. Hence, this design approach not only enriches the existing modern control system design theory, but also makes possible its wide application.

Modern (or state space) control theory was developed in the 1960s. The theory has evolved such that the state feedback control and its implementing observer are designed *separately* (following the so-called separation principle [Wil, 1995]). With this existing design approach, although the direct state feedback system can be designed to have good performance and robustness, almost all the actual corresponding observer feedback systems have entirely different robustness. In the new design approach presented here, the state feedback control and its implementing observer are designed *together*. More explicitly, the state feedback control is designed based on the results of its implementing observer. The resulting state feedback control is the *generalized state feedback control* [Tsui, 1999b].

This fundamentally new approach guarantees—for all open-loop systems with more outputs than inputs or with at least one stable transmission zero—the same loop transfer function and therefore the same robustness of the observer feedback system and the corresponding direct state feedback system. Most open-loop systems satisfy either of these two conditions. For all other open-loop systems, this approach guarantees that the difference between the loop transfer functions of the above two feedback systems be kept minimal in a simple least-square sense.

Modern and classical control theories are the two major components of control systems theory. Compared with classical control theory, modern control theory can describe a single system's performance and robustness more accurately, but it lacks a clear concept of feedback system robustness, such as the loop transfer function of classical control theory. By fully using the concept of loop transfer functions, the approach exploits the advantages of both classical and modern control theories. This approach guarantees the robustness and loop transfer function of classical control theory, while designing this loop transfer function much more effectively (though indirectly) using modern control design techniques. Thus it achieves *both* good robustness and performance for feedback control systems.

If the first edition of this book emphasized the first of the above two advantages (i.e., the true realization of robustness properties of feedback control), then this second edition highlights the second of the above two advantages—the far more effective design of high performance and robustness feedback control itself.

A useful control theory should provide general and effective guidance on complicated control system design. To achieve this, the design formulation must fully address both performance and robustness. It must also exploit fully the existing design freedom and apply a general, simple, and explicit design procedure. The approach presented here truly satisfies these requirements. Since this book concentrates on this new design approach and its relevant analysis, other analytical control theory results are presented with an emphasis on their physical meanings, instead of their detailed mathematical derivations and proofs.

The following list shows several of the book's most important results. With the exception of the third item, these results are not presented in any other books:

1. The first general dynamic output feedback compensator that can implement state or generalized state feedback control, and its design procedure. The feedback system of this compensator is the first general feedback system that has the same robustness properties of its corresponding direct state feedback system (Chapters 3 to 6).
2. A systematic, simple, and explicit eigenvalue assignment procedure using static output feedback control or generalized state feedback control (Section 8.1). This procedure enables the systematic eigenvector assignment procedures of this book, and is general to most open-loop system conditions if based on the generalized state feedback control of this book.
3. Eigenvector assignment procedures that can fully use the freedom of this assignment. Both numerical algorithms and analytical procedures are presented (Section 8.2).
4. A general failure detection, isolation, and accommodation compensator that is capable of considering system model uncertainty and measurement noise, and its systematic design procedure (Chapter 10).
5. The simplest possible formulation, and a truly systematic and general procedure, of minimal order observer design (Chapter 7).
6. Solution of the matrix equation $TA - FT = LC$ [matrix pair (A, C) is observable and eigenvalues of matrix F are arbitrarily assigned]. This solution is general and has all eigenvalues of F and all rows of T completely decoupled (F is in Jordan form). This solution uniquely enables the full use of the remaining freedom of this matrix equation, which is fundamentally important in most of the basic design problems of modern control theory (Chapters 5 to 8, 10).
7. The basic design concept of generating a state feedback control signal without estimating all state variables, and the generalization of this design concept from function observers only to all feedback compensators (Chapters 3 to 10).
8. The complete unification of two existing basic feedback structures of modern control theory—the zero input gain state

observer feedback structure and the static output feedback structure (Section 6.3).

9. A more generally accurate robust stability measure that is expressed in terms of the sensitivities of each system pole. This analytical measure can be used to guide systematic feedback system design (Sections 2.2.2 and 8.2).
10. Comparison of computational complexity and therefore trackability (ability to adjust the original design formulation based on the final and numerical design results) of all feedback control design techniques (Section 9.3).
11. Emphasis on the distinct advantages of high performance/robustness control design using eigenstructure assignment techniques over the techniques for the direct design of loop transfer functions (Chapters 2, 3, 8, 9).
12. The concept of adaptive control and its application in failure accommodation and control (Section 10.2).

The first five of the above results are actual design results. The last seven are new theoretical results and concepts that have enabled the establishment of the first five results. In other words, the main new result (result 1, the full realization of robustness properties of state/generalized state feedback control) is enabled by some significant and fundamental developments (such as results 6 to 8), and is validated by the distinct effectiveness of state/generalized state feedback control (results 2 to 3 and 9 to 11).

This book also addresses the computational reliability of its analysis and design algorithms. This is because practical control problems usually require a large amount of computation, and unreliable computation can yield totally unreliable results. Every effort has been made to use reliable computational methods in design algorithms, such as the computation of Hessenberg form (instead of the canonical form) and of orthogonal matrix operation (instead of elementary matrix operation).

As a result, the computation required in this book is slightly more complicated, but the more reliable results thus obtained make the effort worthwhile. It should be noted that the computation of polynomials required by the classical control theory is usually unreliable. The development of computational software has also eased considerably the complexity of computation. Each design procedure is presented in algorithm form, and each step of these algorithms can be implemented directly by the existing computational software.

This book will be useful to control system designers and researchers. Although a solid background in basic linear algebra is required, it requires remarkably less mathematical sophistication than other books similar in scope. This book can also be used as a textbook for students who have had a first course (preferably including state space theory) in control systems. Multi-input and multi-output systems are discussed throughout. However, readers will find that the results have been substantially simplified to be quite easily understandable, and that the results have been well unified with the single-input and single-output system results. In addition, this book is comprehensive and self-contained, with every topic introduced at the most basic level. Thus it could also be used by honor program students with background in signals and systems only.

An overview of each chapter follows. Chapter 1 introduces basic system models and properties. Chapter 2 analyzes the performance and sensitivity of a single overall system. Chapter 3 describes the critical role of loop transfer functions on the sensitivity of feedback systems, including the observer feedback systems. Chapter 4 proposes the new design approach and analyzes its advantages. Chapter 5 presents the solution of a basic matrix equation. This solution is used throughout the remaining chapters (except Chapter 9). Chapter 6 presents the design of the dynamic part of the observer such that for any state feedback control signal generated by this observer, the loop transfer function of this control is also fully realized. Chapter 7 presents the design of the function observer, which generates an arbitrarily given state feedback control signal, with minimized observer order. Chapter 8 presents the eigenvalue/vector assignment control design methods. Chapter 9 introduces the linear quadratic optimal control design methods. Both designs of Chapters 8 and 9 will determine the output part of the observer of Chapter 6, as well as the "target" closed-loop system loop transfer function. Comparison of various designs reveals two distinct advantages of eigenstructure assignment design. Chapter 10 deals with the design of a general failure detection, isolation, and (adaptive) accommodation compensator that is capable of considering system model uncertainty and measurement noise. This compensator has the compatible structure of—and can be implemented in coordination with—the normal (free of major failure) robust control compensator of this book. There is a set of simple exercises at the end of each chapter.

To make the book self-contained, Appendix A provides a simple introduction to the relevant mathematical background material. Appendix B lists the mathematical models of eight real-world systems for synthesized design practice.

I would like to thank everyone who helped me, especially during my student years. I also thank my former student Reza Shahriar, who assisted with some of the computer graphics.

Chia-Chi Tsui

Contents

1

System Mathematical Models and Basic Properties

Unlike other engineering specialities whose subject of study is a specific engineering system such as an engine system or an airborne system, control systems theory studies only a general mathematical model of engineering systems. This chapter introduces two basic mathematical models and some basic system properties revealed by these models. There are four sections in this chapter.

Section 1.1 introduces the state space model and transfer function model of linear time-invariant multi-input and multi-output systems, and the basic relationship between these two models.

Section 1.2 describes the eigenstructure decomposition of the state space model, where the dynamic matrix of this model is in Jordan form.

Section 1.3 introduces two basic system properties—controllability and observability.

Section 1.4 introduces two basic system parameters—system poles and zeros. These properties and parameters can be simply and clearly described based on the eigenstructure decomposition of the state space model.

1.1 TWO KINDS OF MATHEMATICAL MODELS

This book studies only the linear time-invariant systems, which have also been the main subject of control systems theory. A linear time-invariant system can be represented by two kinds of mathematical models—the state space model and the transfer function model. The control theory based on the state space model is called the "state space control theory" or the "modern control theory," and the control theory based on the transfer function model is called the "classical control theory."

We will first introduce the state space model and its derivation.

A state space model is formed by a set of first-order linear differential equations with constant coefficients (1.1a) and a set of linear equations (1.1b)

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \tag{1.1a}$$

$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) \tag{1.1b}$$

where

$\mathbf{x}(t) = [x_1(t), \ldots, x_n(t)]'$ is the system state vector (the prime symbol stands for transpose)
$x_i(t), i = 1, \ldots, n$ are the system state variables
$\mathbf{u}(t) = [u_1(t), \ldots, u_p(t)]'$ is the system input
$\mathbf{y}(t) = [y_1(t), \ldots, y_m(t)]'$ is the system output

and the system matrices (A, B, C, D) are real, constant, and with dimensions $n \times n, n \times p, m \times n$, and $m \times p$, respectively.

In the above model, Eq. (1.1a) is called the "dynamic equation," which describes the "dynamic part" of the system and how the initial system state $\mathbf{x}(0)$ and system input $\mathbf{u}(t)$ will determine the system state $\mathbf{x}(t)$. Hence matrix A is called the "dynamic matrix" of the system. Equation (1.1b) describes how the system state $\mathbf{x}(t)$ and system input $\mathbf{u}(t)$ will instantly determine system output $\mathbf{y}(t)$. This is the "output part" of the system and is static (memoryless) as compared with the dynamic part of the system.

From the definition of (1.1), parameters p and m represent the number of system inputs and outputs, respectively. If $p > 1$, then we call the corresponding system "multi-input." If $m > 1$, then we call the corresponding system "multi-output." A multi-input or multi-output system is also called a "MIMO system." On the other hand, a system is called "SISO" if it is both single-input and single-output.

In (1.1), the physical meaning of system state $\mathbf{x}(t)$ is used to describe completely the energy distribution of the system at time t, especially at $t = 0$ (initial time of system operation).

For example, in electrical circuit systems with linear time-invariant circuit elements (inductors, resistors, and capacitors), the system state is formed by all independent capacitor voltages and inductor currents. Thus its initial condition $\mathbf{x}(0)$ can completely describe the initial electrical charge and initial magnetic flux stored in the circuit system.

Another example is in linear motion mechanical systems with linear time-invariant elements (springs, dampers, and masses), in which the system state is formed by all independent mass velocities and spring forces. Thus its initial state $\mathbf{x}(0)$ completely describes the initial dynamic energy and initial potential energy stored in the mechanical system.

Because of this reason, the number (n) of system states also indicates the number of the system's independent energy storage devices.

Example 1.1

The following electrical circuit system is a linear time-invariant system (Fig. 1.1).

Letting $v_1(t)$ and $v_2(t)$ be the node voltages of the circuit, and letting the capacitor voltage and inductor current be the two system states $x_1(t)$ and $x_2(t)$, respectively, we have

$$v_1(t) = x_1(t) \qquad \text{and} \qquad v_2(t) = x_1(t) - R_2 x_2(t) \tag{1.2}$$

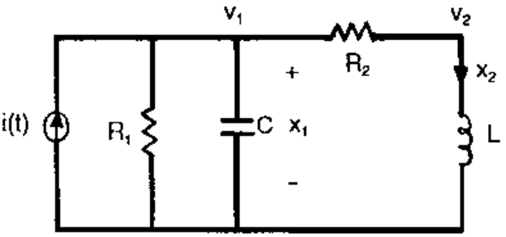


Figure 1.1 A linear time-invariant circuit system.

In other words, all node voltages and branch currents can be expressed in terms of system states and inputs. Thus the system's output part (1.1b) can be directly derived. For example, if the output $\mathbf{y}(t)$ is designated as $[v_1(t), v_2(t)]'$, then from (1.2),

$$\mathbf{y}(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -R_2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + 0 \underset{=}{\triangle} C\mathbf{x}(t) + 0\mathbf{u}(t)$$

The dynamic equation of this circuit system can also be derived by standard circuit analysis. Applying Kirchoff's current law at each node of the circuit, we have

$$i(t) = C\dot{v}_1(t) + \frac{v_1(t)}{R_1} + \frac{[v_1(t) - v_2(t)]}{R_2} \tag{1.3a}$$

$$0 = \frac{[v_2(t) - v_1(t)]}{R_2} + \frac{[\int v_2(t)\,dt]}{L} \tag{1.3b}$$

Substituting (1.2) into (1.3) and after simple manipulation [including taking derivatives on both sides of (1.3b)], we can have the form of (1.1a)

$$\dot{x}_1(t) = \frac{-1}{(CR_1)} x_1(t) + \left(\frac{-1}{C}\right) x_2(t) + \left(\frac{1}{C}\right) i(t)$$

$$\dot{x}_2(t) = \frac{1}{L} x_1(t) + (\frac{-R_2}{L}) x_2(t)$$

Thus comparing (1.1a), the system matrices are

$$A = \begin{bmatrix} -1/(CR_1) & -1/C \\ 1/L & -R_2/L \end{bmatrix} \qquad B = \begin{bmatrix} 1/C \\ 0 \end{bmatrix}$$

Example 1.2

The following linear motion mechanical system is a linear time-invariant system (Fig. 1.2).

Letting $v_1(t)$ and $v_2(t)$ be the node velocities in the system, and letting the mass velocity and spring force be the system states $x_1(t)$ and $x_2(t)$, respectively, then

$$v_1(t) = x_1(t) \qquad \text{and} \qquad v_2(t) = x_1(t) - D_2^{-1} x_2(t) \tag{1.4}$$

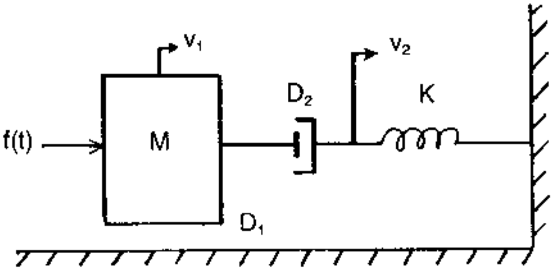


Figure 1.2 A linear time-invariant mechanical system.

In other words, all velocities and forces within this mechanical system can be expressed in terms of the system states and the applied input force. The system's output part (1.1b) can thus be directly derived. For example, if the system output $\mathbf{y}(t)$ is designated as $[v_1(t), v_2(t)]'$, then from (1.4),

$$\mathbf{y}(t) = \begin{bmatrix} v_2(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -D_2^{-1} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \overset{\triangle}{=} C\mathbf{x}(t) + 0\mathbf{u}(t)$$

The dynamic equation of this mechanical system can also be derived using standard dynamic analysis. Balancing the forces at each node of this system, we have

$$f(t) = M\dot{v}_1(t) + D_1 v_1(t) + D_2[v_1(t) - v_2(t)] \tag{1.5a}$$

$$0 = D_2[v_2(t) - v_1(t)] + K[\textstyle\int v_2(t)dt] \tag{1.5b}$$

Substituting (1.4) into (1.5) and after simple manipulation [including taking derivatives on both sides of (1.5b)], we can have the form of (1.1a)

$$\dot{x}_1(t) = \left(\frac{-D_1}{M}\right)x_1(t) + \left(\frac{-1}{M}\right)x_2(t) + \frac{1}{M}f(t)$$

$$\dot{x}_2(t) = Kx_1(t) + \left(\frac{-K}{D_2}\right)x_2(t)$$

Comparing (1.1a), the system matrices of this system are

$$A = \begin{bmatrix} -D_1/M & -1/M \\ K & -K/D_2 \end{bmatrix}, \qquad B = \begin{bmatrix} 1/M \\ 0 \end{bmatrix}$$

In the above two examples, the forms and derivations of state space models are very similar to each other. We call different physical systems that are similar in terms of mathematical models "analogs." This property enables the simulation of the behavior of one physical system (such as a mechanical system) by comparison with an analog but different physical system (such as a circuit system), or by the numerical solution of the mathematical model of that system. We call the former "analog simulation" and the latter "digital simulation."

The use of analogs can be extended to a wide range of linear time-invariant physical systems, such as rotational mechanical systems, thermodynamic systems, and fluid dynamic systems. Therefore, although the mathematical models and the control theory which is based on these models are abstract, they can have very general applications.

A linear time-invariant system can have another kind of mathematical model, called the transfer function model, which can be derived from its corresponding state space model.

Taking the Laplace transforms on both sides of (1.1),

$$X(s) = (sI - A)^{-1}\mathbf{x}(0) + (sI - A)^{-1}BU(s) \tag{1.6a}$$

$$Y(s) = CX(s) + DU(s) \tag{1.6b}$$

where $X(s)$, $U(s)$, and $Y(s)$ are the Laplace transforms of $\mathbf{x}(t)$, $\mathbf{u}(t)$, and $\mathbf{y}(t)$, respectively, and I stands for an n-dimensional identity matrix such that $sIX(s) = sX(s)$.

Substituting (1.6a) into (1.6b), we have

$$Y(s) = \underset{\substack{\text{Zero input response} \\ Y_{zi}(s)}}{C(sI - A)^{-1}\mathbf{x}(0)} + \underset{\substack{\text{Zero state response} \\ Y_{zs}(s)}}{[C(sI - A)^{-1}B + D]U(s)} \tag{1.6c}$$

From superposition principle of linear systems, Eqs. (1.6a) and (1.6c) each have two terms or two contributing factors. The first term is due to the system's initial state $\mathbf{x}(0)$ only and the second is due to system input $U(s)$ only. For example, in (1.6c), the system output (also called the system "response") $Y(s)$ equals the first term if the system input is zero. We therefore define the first term of (1.6c) as "zero input response $Y_{zi}(s)$." Similarly, $Y(s)$ equals the second term of (1.6c) if system initial state is zero, and it is therefore defined as the "zero state response $Y_{zs}(s)$." The form of (1.6) is guaranteed by the linearity property of the state space model (1.1) and of the Laplace transform operator.

The system's transfer function model $G(s)$ is defined from the system's zero state response as

$$Y_{zs}(s) = G(s)U(s) \tag{1.7}$$

Therefore from (1.6c),

$$G(s) = C(sI - A)^{-1}B + D \tag{1.8}$$

The definition of $G(s)$ shows that it reflects only the relationship between the system input $U(s)$ and output $Y(s)$. This relationship (1.7, 1.8) is derived by combining and simplifying a more detailed system structure (1.6a,b), which involves explicitly system state $X(s)$ and which is derived directly from the state space model (1.1). In addition, the transfer function model does not reflect directly and explicitly the system's zero input response, which is as important as zero state response.

Example 1.3

Consider the following *RC* circuit system (a) and mechanical system (b) with a mass M and a frictional force D (Fig. 1.3):

Balancing the currents of (a) and the forces of (b), we have

$$i(t) = C\dot{v}(t) + \frac{[v(t) - 0]}{R}$$

and

$$f(t) = M\dot{v}(t) + D[v(t) - 0]$$

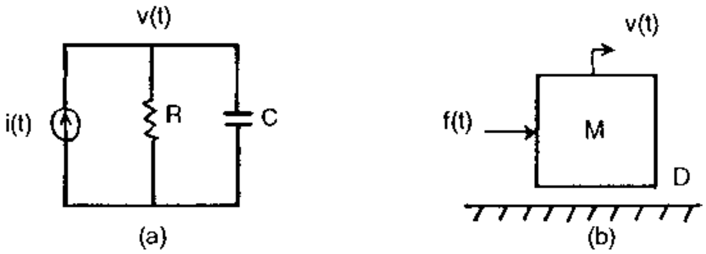


Figure 1.3 First-order circuit and mechanical systems.

Comparing (1.1a), the system matrices ($A \triangleq \lambda$, B) equal ($-1/RC$, $1/C$) and ($-D/M$, $1/M$) for the above two systems, respectively.

Taking Laplace transforms on these two equations and after manipulation, we have the form of (1.6a) or (1.6c) as

$$V(s) = \frac{1}{s-\lambda} v(0) + \frac{B}{s-\lambda} U(s)$$

where $V(s)$ and $U(s)$ are the Laplace transforms of $v(t)$ and system input signal [$i(t)$ or $f(t)$], respectively.

Letting $U(s) = F/s$ (or step function) and taking the inverse Laplace transforms on the above equation, we have, for $t \geqslant 0$,

$$\begin{aligned} v(t) = \mathscr{L}^{-1}\{V(s)\} &= \mathrm{e}^{\lambda t} v(0) + F\left(\frac{-B}{\lambda}\right)[1 - \mathrm{e}^{\lambda t}] \\ &\triangleq v_{zi}(t) + v_{zs}(t) \end{aligned}$$

In each of the above expressions of $V(s)$ and $v(t)$, the two terms are zero input response and zero state response, respectively. The two terms of $v(t)$ have the waveforms shown in Fig. 1.4.

The first waveform of Fig. 1.4 shows that the zero input response starts at its initial condition and then decays exponentially to zero with a time constant $|1/\lambda|$. In other words, the response decays to 36.8% of its initial value at $t = |1/\lambda|$.

This waveform has very clear physical meaning. In the circuit system (a), this waveform shows (when the input current is zero) how the capacitor charge [$= Cv(t)$] is discharged to zero through the resistor R with current $v(t)/R$, and with a time constant RC. In other words, the larger the capacitor or resistor, the slower the discharge process. In the mechanical system (b), this waveform shows with zero input force how the momentum ($= Mv(t)$) slows to zero by the frictional force $Dv(t)$, with a time constant M/D. In other words, the larger the mass and the smaller the friction D, the longer the time for the velocity to slow to 36.8% of its initial value.

The second waveform of Fig. 1.4 shows that the zero state response starts at zero and then reaches exponentially to its steady state level, which is specified by the input level F. This process also has a time constant $|1/\lambda|$, which means that the response reaches $1 - 36.8\% = 63.2\%$ of its final value at $t = |1/\lambda|$.

This waveform also has very clear physical meaning. In the circuit system (a), this waveform shows how the capacitor is charged from zero until $v(t) = -(B/\lambda)F = RF$, by a constant current source $F\mathbf{u}(t)$. The final

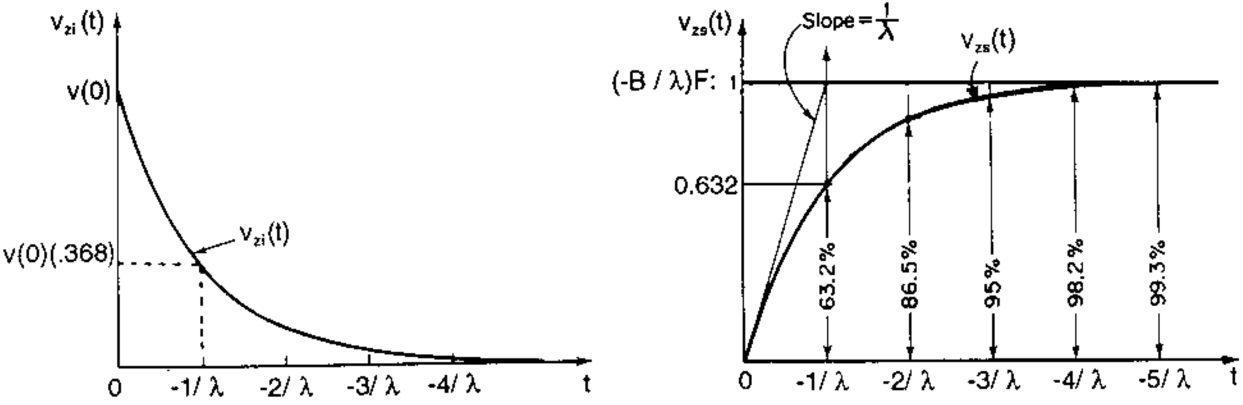


Figure 1.4 Waveforms of zero input response and zero state response of a first-order system.

value of $v(t)$ equals the supply side voltage, which means that the capacitor is fully charged. This charging process has a time constant RC, which means the larger the capacitor or the resistor, the slower the charging process. In the mechanical system (b), this waveform shows how the mass is accelerated from zero to $-(B/\lambda)F = F/D$ by a constant force $F\mathbf{u}(t)$. This acceleration process has a time constant M/D, which implies that the larger the mass or the higher the final velocity F/D, which is implied by a lower D, the longer the time for the mass to accelerate to 63.2% of its final velocity.

This example shows a very fitting analogy between the two systems, and the solution of their common mathematical model. This example also shows the importance of the initial state of the system (initial capacitor charge and initial mass velocity, respectively) and its effects on the system—the zero input response (discharging and de-acceleration, respectively).

The definition (1.7)–(1.8) of transfer function model $G(s)$ implies that $G(s)$ cannot in general describe explicitly and directly the system's zero input response, especially when the system has many state variables, inputs, and outputs. Because transient response is defined as the complete system response before reaching steady state and is therefore closely related to the system's zero input response, the inherent feature of the transfer function model will inevitably jeopardize the understanding of the system's transient response, whose quickness and smoothness is a major part of system performance, as will be defined in the next chapter.

In Example 1.3, the concept of time constant is used as a measure of transient response and is closely related to zero input response.

In both the state space model (1.1) and the transfer function model (1.8), the system matrix D reflects only an independent and static relation between system inputs and outputs. This relation can be easily measured and cancelled in the analysis and design. For this reason, we will assume $D = 0$ in the rest of this book. Using this assumption, the transfer function model of (1.8) now becomes

$$G(s) = C(sI - A)^{-1}B \tag{1.9}$$

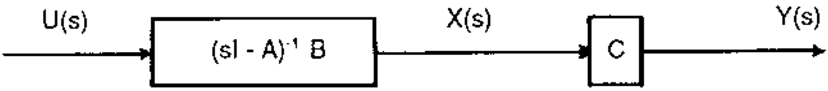


Figure 1.5 Partitioned block diagram representation of a system's transfer function model.

Finally, the transfer function model (1.9) can be represented by the block diagram in Fig. 1.5, which is a series connection of two blocks.

1.2 EIGENSTRUCTURE DECOMPOSITION OF A STATE SPACE MODEL

To gain a simpler yet deeper understanding of system structure and properties, we partition the system dynamic matrix

$$A = V\Lambda V^{-1} \underset{=}{\triangle} \begin{bmatrix} | & & | \\ V_1 : & \ldots : & V_q \\ | & & | \end{bmatrix} \begin{bmatrix} \Lambda_1 & & \\ & \ddots & \\ & & \Lambda_q \end{bmatrix} \begin{bmatrix} \text{-}T_1\text{-} \\ \vdots \\ \text{-}T_q\text{-} \end{bmatrix} \tag{1.10a}$$
$$\underset{=}{\triangle} T^{-1}\Lambda T$$

where $\Lambda = \mathrm{diag}\{\Lambda_1, \ldots, \Lambda_q\}$ is called a "Jordan form matrix," whose diagonal matrix blocks Λ_i ($i = 1, \ldots, q$, called "Jordan blocks") are formed by the eigenvalues ($\lambda_i, i = 1, \ldots, n$) of matrix A according to the following rules:

$\Lambda_i = \lambda_i$, if λ_i is real and distinct

$$\Lambda_i = \begin{bmatrix} \sigma_i & \omega_i \\ -\omega_i & \sigma_i \end{bmatrix}, \quad \begin{matrix}\text{if the corresponding } \lambda_i \text{ and } \lambda_{i+1} \text{ are} \\ \text{a complex conjugate pair } \sigma_i \pm j\omega_i\end{matrix}$$

$\Lambda_i = \mathrm{diag}\{\Lambda_{i,j}, j = 1, \ldots, q_i\}$, if the corresponding λ_i repeats n_i times, and the $n_{i,j}$ dimensional matrix

$$\Lambda_{i,j} = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix} \text{(blank entries are all 0's)} \tag{1.10b}$$

and is called "bidiagonal form matrix," where

$$n_{i,1} + \cdots + n_{i,qi} = n_i$$

Finally, the sum of dimensions of all Jordan blocks Λ_i ($i = 1, \ldots, q$) equals n.

When matrix A is in (1.10), the corresponding state space model is said to be in "Jordan canonical form." Any real square matrix (and any dynamic matrix) A can have the eigenstructure decomposition such as (1.10).

Because (1.10) implies $AV - V\Lambda = 0$, we call matrix V the "right eigenvector matrix" of matrix A, and call each column of matrix $V, \mathbf{v}_i$ $(i = 1, \ldots, n)$, the "right eigenvector" of matrix A corresponding to λ_i. Similarly, because $TA - \Lambda T = 0$, we call matrix T the "left eigenvector matrix" of matrix A and call each row of matrix $T, \mathbf{t}_i$ $(i = 1, \ldots, n)$, the "left eigenvector" of matrix A corresponding to λ_i. All but the first eigenvectors corresponding to the Jordan block (1.10b) are derived based on each other and are called the "generalized eigenvectors."

From (1.10),

$$(sI - A)^{-1} = [V(sI - \Lambda)V^{-1}]^{-1} = V(sI - \Lambda)^{-1}V^{-1}$$

Therefore, from (1.9) and the inverse matrix rules,

$$G(s) = CV(sI - \Lambda)^{-1}V^{-1}B \tag{1.11a}$$

$$= \frac{CV\text{adj}(sI - \Lambda)V^{-1}B}{\det(sI - \Lambda)} \tag{1.11b}$$

$$= \frac{CV\text{adj}(sI - \Lambda)V^{-1}B}{(s - \lambda_1)\ldots(s - \lambda_n)} \tag{1.11c}$$

where adj$(\cdot)$ and det$(\cdot)$ stand for the adjoint and the determinant of the corresponding matrix, respectively.

From (1.11c), transfer function $G(s)$ is a rational polynomial matrix. It has an n-th order denominator polynomial whose n roots equal the n eigenvalues of the system dynamic matrix, and which is called the "characteristic polynomial" of the system.

Comparing (1.11a) with (1.9), a new system matrix triple $(\Lambda,\ V^{-1}B,\ CV)$ has the same transfer function as that of system matrix triple $(A,\ B,\ C)$, provided that $A = V\Lambda V^{-1}$. We call these two state space models and their corresponding systems "similar" to each other and call the transformation between the two similar state space models "similarity transformation." This property can be extended to any system matrix triple $(Q^{-1}AQ,\ Q^{-1}B,\ CQ)$ for a nonsingular Q.

The physical meaning of similar systems can be interpreted as follows. Let $\mathbf{x}(t)$ and $\bar{\mathbf{x}}(t)$ be the state vectors of state space models $(A,\ B,\ C)$ and $(Q^{-1}AQ,\ Q^{-1}B,\ CQ)$, respectively. Then from (1.1),

$$\dot{\bar{\mathbf{x}}}(t) = Q^{-1}AQ\bar{\mathbf{x}}(t) + Q^{-1}B\mathbf{u}(t) \tag{1.12a}$$

$$\mathbf{y}(t) = CQ\bar{\mathbf{x}}(t) + D\mathbf{u}(t) \tag{1.12b}$$

It is clear from (1.1a) and (1.12a) that

$$\mathbf{x}(t) = Q\bar{\mathbf{x}}(t) \qquad \text{or} \qquad \bar{\mathbf{x}}(t) = Q^{-1}\mathbf{x}(t) \tag{1.13}$$

From Definitions A.3–A.4 of Appendix A, (1.13) implies that the only difference between the state space models (1.1) and (1.12) is that the state vectors are based on different basis vector matrices (I and Q, respectively).

Similarity transformation, especially when the state space model is transformed to "Jordan canonical form" where the dynamic matrix is in Jordan form, is a very effective and very frequently used scheme which can substantially simplify the understanding of the system, as will be shown in the rest of this chapter.

1.3 SYSTEM ORDER, CONTROLLABILITY, AND OBSERVABILITY

Definition 1.1

The order n of a system equals the order of the system's characteristic polynomial. It is clear from (1.11c) that system order also equals the number of states of the system.

Let us discuss the situation of the existence of common factors between the transfer function's numerator and denominator polynomials. Because this denominator polynomial is defined as the system's characteristic polynomial, and because common factors can cancel out each other, the above situation implies that the corresponding system order is reducible. We call this kind of system "reducible." Otherwise the system is said to be "irreducible."

The situation of reducible systems can be more explicitly described by their corresponding state space models. Definition 1.1 implies that in reducible systems, some of the system states are not involved with the system's input and output relation $G(s)$. In other words, in reducible systems, some of the system states either cannot be influenced by any of the system inputs, or cannot influence any of the system outputs. We will define these two situations separately in the following.

Definition 1.2

If there is at least one system state which cannot be influenced by any of the system inputs, then the system is uncontrollable; otherwise the system is controllable. Among many existing criteria of controllability, perhaps the simplest is that a system is controllable if and only if there exists no constant λ such that the rank of matrix $[\lambda I - A : B]$ is less than n.

Definition 1.3

If there is at least one system state which cannot influence any of the system outputs, then the system is unobservable; otherwise the system is observable. Among many existing criteria of observability, perhaps the simplest is that a system is observable if and only if there exists no constant λ such that the rank of matrix $[\lambda I' - A' : C']$ is less than n.

Because the rank of matrix $\lambda I - A$ always equals n if λ is not an eigenvalue of A, the above criteria can be checked only for the n values of λ which equal the eigenvalues of matrix A.

It is clear that an irreducible system must be both controllable and observable. Any uncontrollable or unobservable system is also reducible.

Up to this point, we can see a common and distinct phenomenon of linear systems—duality. For example, in linear systems, current and voltage, force and velocity, charge and flux, dynamic energy and potential energy, capacitance and inductance, mass and spring are dual pairs. In linear algebra and linear control theory which describe linear systems, matrix columns and rows, right and left eigenvectors, inputs and outputs, and controllability and observability are also dual to each other.

The phenomenon of duality can not only help us understand linear systems comprehensively, but also help us solve some specific analysis and design problems. For example, the determination of whether a system (A, B) is controllable can be replaced by the determination of whether a system $(A = A', C = B')$ is observable instead.

Because matrix $[\lambda I - Q^{-1}AQ : Q^{-1}B] = Q^{-1}[(\lambda I - A)Q : B]$ has the same rank as that of matrix $[\lambda I - A : B]$, similarity transformation will not change the controllability property of the original system. Similarity transformation changes only the basis vector matrix of state vectors of the system's state space model and therefore cannot change the system's basic properties such as controllability. From duality, similarity transformation cannot change the observability of the system either. It is therefore valid to determine a system's controllability and observability conditions after similarity transformation.

The following three examples show the relative simplicity of determining controllability and observability when the system matrices are in special forms (especially the Jordan canonical form), which can be derived from any system matrices by similarity transformation.

Example 1.4

Determine whether the system

$$(A, B, C) = \left(\begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix}, \begin{bmatrix} \text{-}\mathbf{b}_1\text{-} \\ \text{-}\mathbf{b}_2\text{-} \\ \text{-}\mathbf{b}_3\text{-} \end{bmatrix}, [\mathbf{c}_1 : \mathbf{c}_2 : \mathbf{c}_3] \right)$$

is controllable and observable.

From Definition 1.2, it is clear that if any row of matrix B equals zero, say $\mathbf{b}_i = 0$ $(i = 1,\ 2,\ 3)$, then there exist a constant $\lambda = -i$ such that the i-th row of matrix $[\lambda I - A : B]$ equals zero. Only when every row of matrix B is nonzero, then the rank of matrix $[\lambda I - A : B]$ equals n, for $\lambda = -i$ $(i = 1, 2, 3) =$ all eigenvalues of matrix A. Thus the necessary and sufficient condition for this system to be controllable is that every row of matrix B is nonzero.

Similarly (from duality), the necessary and sufficient condition for this system to be observable is that every column of matrix C is nonzero.

From (1.9), the transfer function of this system is

$$\begin{aligned} G(s) &= C(sI - A)^{-1}B \\ &= \frac{\mathbf{c}_1(s+2)(s+3)\mathbf{b}_1 + \mathbf{c}_2(s+1)(s+3)\mathbf{b}_2 + \mathbf{c}_3(s+1)(s+2)\mathbf{b}_3}{(s+1)(s+2)(s+3)} \end{aligned}$$

It is clear that if any $\mathbf{b}_i$ or $\mathbf{c}_i$ equals zero $(i = 1, 2, 3)$, then there will be common factors between the numerator and denominator polynomials of $G(s)$. However, the reducible transfer function $G(s)$ cannot indicate the converse: whether a row of matrix B or a column of matrix C is zero, or whether the system is uncontrollable or unobservable or both. In this sense, the information provided by the transfer function model is less complete and explicit than the state space model.

Controllability and observability conditions can also be clearly revealed from the system's block diagram.

Figure 1.6 shows clearly that any system state $x_i(t)$ is influenced by the input $\mathbf{u}(t)$ if and only if the corresponding $\mathbf{b}_i \neq 0$ $(i = 1,\ 2,\ 3)$, and that any $x_i(t)$ influences output $\mathbf{y}(t)$ if and only if $\mathbf{c}_i \neq 0$.

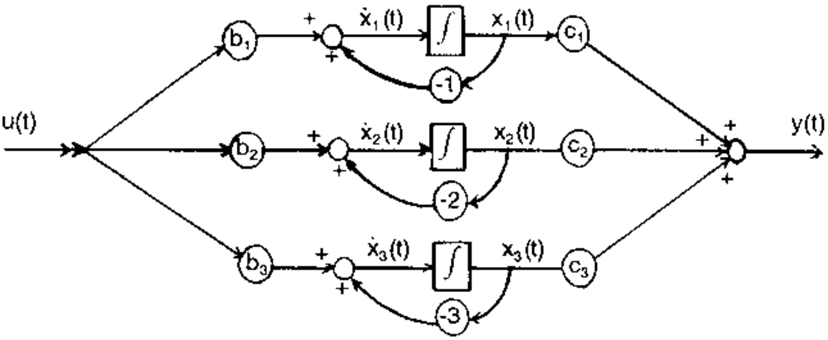


Figure 1.6 Block diagram of the system from Example 1.4.

Example 1.5

Example 1.4 is a Jordan canonical formed system with distinct and real eigenvalues. The present example studies the same system with multiple eigenvalues [see (1.10b)]. Let

$$(A, B, C) = \left(\begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix}, \begin{bmatrix} \text{-}\mathbf{0}\text{-} \\ \text{-}\mathbf{0}\text{-} \\ \text{-}\mathbf{b}_3\text{-} \end{bmatrix}, [\mathbf{c}_1 : 0 : 0] \right)$$

It is clear that the rank of matrix $[\lambda I - A : B]$ equals n if and only if $\mathbf{b}_3 \neq 0$, and the rank of matrix $[\lambda I' - A' : C']$ equals n if and only if $\mathbf{c}_1 \neq 0$.

In examining the block diagram of this system (Fig. 1.7), it is clear that $\mathbf{b}_3$ and $\mathbf{c}_1$ are the only links between the system states and the system's inputs and outputs, respectively. Because all system states are on a single path in Fig. 1.7, it is of interest to observe that any system state is observable if and only if all gains on that path and on the right side of this state are nonzero. In the dual sense, any state is controllable if and only if all gains on that path and on the left side of this state are nonzero. This property can help

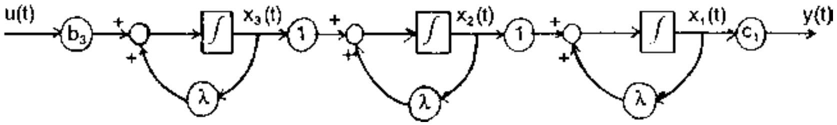


Figure 1.7 Block diagram of the system from Example 1.5.

one to extract the controllable or observable part of the system from the rest of the system (see Sec. 5.1).

Examples 1.4 and 1.5 show that a system's controllability and observability properties can be easily checked based on Jordan canonical forms of the system's state space model. Unfortunately, the computational problem of the similarity transformation to Jordan canonical form is difficult and is usually very sensitive to the initial data variation.

On the other hand, the form of state space model of Example 1.5 is a special case of a so-called Hessenberg form, which can be easily and reliably computed and which can also be used to determine system controllability and observability (see Sec. 5.1). In the next two examples, we will study a second special case of the Hessenberg form state space model.

Example 1.6

The observable canonical form state space model:

$$(A, B, C) = \left(\begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 \\ -a_2 & 0 & 1 & & \\ \vdots & \vdots & & & 0 \\ -a_{n-1} & 0 & & & 1 \\ -a_n & 0 & & \dots & 0 \end{bmatrix}, \begin{bmatrix} \text{-}\mathbf{b}_1\text{-} \\ \text{-}\mathbf{b}_2\text{-} \\ \vdots \\ \text{-}\mathbf{b}_{n-1}\text{-} \\ \text{-}\mathbf{b}_n\text{-} \end{bmatrix}, [c_1, 0, \dots, 0] \right) \tag{1.14}$$

This is a single-output (although it can be a multiple input) system. The above system matrices are said to be in the "observable canonical form." In addition, the system matrix A of (1.14) is called a "companion form" or "canonical form" matrix. Let us examine the block diagram of this system.

Figure 1.8 shows that all system states can influence system output (observable) if and only if $c_1 \neq 0$, but if any of the 1's of matrix A becomes 0, then all system states left of this 1 on the main path (with all system states) of Fig. 1.8 will become unobservable. It has been proven that any single-output (n-th order) observable system is similar to (1.14) [Luenberger, 1967; Chen, 1984].

From duality, if a system model is (A', C', B'), where the system matrix triple (A, B, C) is from (1.14), then this system model is said to be in "controllable canonical form" and is controllable if and only if $c_1 \neq 0$. Any single-input controllable system is similar to this (A', C', B').

Controllable and observable canonical form state space models share an important property in their corresponding transfer function $G(s)$.

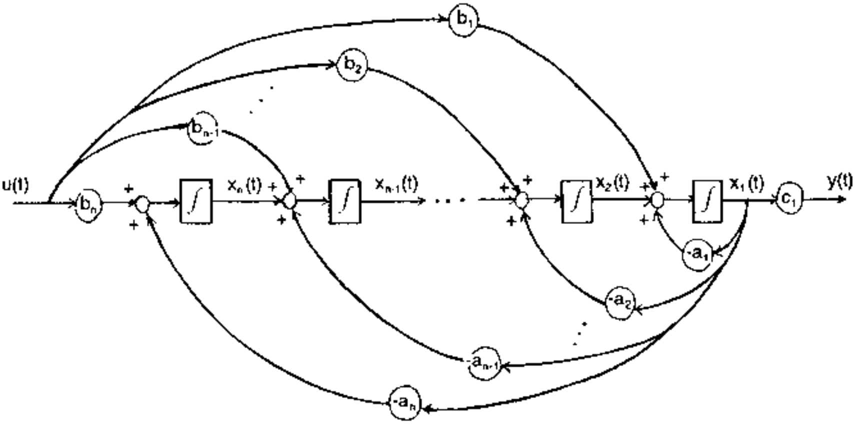


Figure 1.8 Block diagram of a single-output system in observable canonical form.

Substituting (1.14) into (1.9), we have

$$
\begin{aligned}
G(s) &= C(sI - A)^{-1}B \\
&= \frac{c_1(\mathbf{b}_1 s^{n-1} + \mathbf{b}_2 s^{n-2} + \cdots + \mathbf{b}_{n-1}s + \mathbf{b}_n)}{s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots + a_{n-1}s + a_n} \\
&\triangleq \frac{N(s)}{D(s)}
\end{aligned}
\tag{1.15}
$$

In other words, the unknown parameters of the canonical state space model fully match the unknown parameters of the corresponding transfer function. In addition, the n unknown parameters of the companion form matrix A fully match the n unknown coefficients of its characteristic polynomial $D(s)$, which further fully determines all n eigenvalues of the matrix. For this reason, we also call all (either Jordan, controllable, or observable) canonical form state space model the "minimal parameter" model.

The computation of similarity transformation from a general state space model to canonical forms (1.14) and (1.10) implies the compression of system dynamic matrix parameters from general $n \times n$ to only n. In this sense, the computation of (1.14) and (1.10) can be equally difficult [Laub, 1985].

In this single-output system, the corresponding transfer function has the denominator $D(s)$ as a scalar polynomial, and the numerator $N(s)$ as a polynomial row vector. In the next example, we will extend this result into multi-output systems whose transfer function has both its denominator $D(s)$ and numerator $N(s)$ as a polynomial matrix.

Example 1.7

A multi-output observable system in canonical form:

$$A = \begin{bmatrix} A_1 & I_2 & 0 & \ldots & 0 \\ A_2 & 0 & I_3 & & \vdots \\ \vdots & \vdots & & & 0 \\ A_{v-1} & 0 & & & I_v \\ A_v & 0 & & \ldots & 0 \end{bmatrix} \qquad B = \begin{bmatrix} B_1 \\ B_2 \\ : \\ B_{v-1} \\ B_v \end{bmatrix} \tag{1.16}$$

$$C = [I_1 \quad 0 \quad \ldots \quad 0]$$

where the matrix blocks I_i and $i = 1, \ldots, v$ have dimensions $m_{i-1}xm_i$ $(m_0 = m)$ and equal an m_{i-1} dimensional identity matrix with $m_{i-1} - m_i$ columns eliminated. Here $m_1 + \cdots + m_v = n$.

For example, for $m_{i-1} = 3$, the corresponding I_i matrix blocks can be:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Without loss of generality (by assuming that all system outputs are linearly independent [Chen, 1984]), we let $m_1 = m$ and let I_1 be an m-dimensional identity matrix. These m columns will disappear gradually at matrices I_i subsequent to I_1 $(i = 2, \ldots)$. Once the j-th column disappears at I_i, this column and its corresponding row will disappear at subsequent matrices $I_{i+1}, \ldots$. We can therefore distinguish and assign a constant parameter $v_j = i, j = 1, \ldots, m$. From Example 1.6, the disappearance of the j-th column also implies that the j-th output is no longer influenced by any more system states.

It is apparent that the largest value of v_j equals v because all m columns disappear at matrix I_{v+1} in (1.16). It is also proven that any observable system is similar to (1.16) [Luenberger, 1967; Chen, 1984], which is called the "block-observable canonical form" (see Sec. 5.1 also).

To match all unknown parameters of (1.16) directly to all unknown parameters of the corresponding transfer function

$$G(s) = D^{-1}(s)N(s) \tag{1.17a}$$

as was done in Example 1.6, where $D(s)$ and $N(s)$ are $m \times m$ and $m \times p$ dimensional polynomial matrices, respectively, we need to perform the following two preliminary and simple operations.

First, fill $m - m_i$ zero rows into each matrix block A_i and $B_i \, (i = 1, \ldots, v)$. The rows will be filled at the positions corresponding to all missing columns of matrix block I_i and its preceding I_j's $(j = i-1, \ldots, 1)$. For example, if $m = 3$ and I_i takes the above seven different forms, then the zero rows shall be filled at the third, the second, the first, the second and third, the first and third, and the first and second positions of the second to the seventh matrix, respectively. At the end of this operation, all matrix blocks A_i and B_i will become $m \times m$ and $m \times p$ dimensional matrix blocks $\overline{A}_i$ and $\overline{B}_i \, (i = 1, \ldots, v)$, respectively.

Second, form matrices $[I : -\overline{A}_1 : -\overline{A}_2 : \ldots : -\overline{A}_v]$ and $[\overline{B}_1 : \overline{B}_2 : \ldots : \overline{B}_v]$ and then circular shift (shift in zeros) each row (say, the j-th row) of these two matrices to the right by $m(v - v_j)$ or $p(v - v_j)$ positions, respectively, $j = 1, \ldots, m$. We denote the two resulting matrices of this step as $[\tilde{I}_0 : \tilde{A}_1 : \tilde{A}_2 : \ldots : \tilde{A}_v]$ and $[\tilde{B}_1 : \tilde{B}_2 : \ldots : \tilde{B}_v]$, respectively.

Finally, in (1.17a),

$$D(s) = \tilde{I}_0 s^v + \tilde{A}_1 s^{v-1} + \cdots + \tilde{A}_{v-1} s + \tilde{A}_v \tag{1.17b}$$

and

$$N(s) = \tilde{B}_1 s^{v-1} + \tilde{B}_2 s^{v-2} + \cdots + \tilde{B}_{v-1} s + \tilde{B}_v \tag{1.17c}$$

It can be verified that the above (1.17) equals the $G(s)$ of (1.9), which is computed from (A, B, C) of (1.16) [Tsui and Chen, 1983a]. (See Exercise 1.3 to 1.6 for the numerical examples of this result.)

The above two steps do not change, add or eliminate any parameter of (A, B, C) of (1.16). Therefore, these two steps, which have not appeared explicitly before, enable the direct match between the parameters of state space model (1.16) and the parameters of the transfer function model (1.17a). A significant aspect of this direct parametric match is that it enables the finding of the corresponding state space model (1.16) from a given transfer function model (1.17). This problem is called "realization."

Comparing the forms of (1.14) and (1.16), the former is truly a special case of the latter when $m = 1$. Therefore, the novel operation of (1.17) is *the* direct generalization of realization problem, from the SISO case ($m = 1$) to the MIMO case ($m > 1$).

Because the realization from (1.17) to (1.16) is easy, a transfer function model-based design method can easily find its corresponding method in state space theory. On the other hand, the computation of (1.16) from a general state space model is very difficult (see the previous example). Therefore it is difficult to find the corresponding design method in classical control theory. This is another important reflection of the advantage of state space control theory over classical control theory.

This book discusses only controllable and observable systems.

1.4 SYSTEM POLES AND ZEROS

Definition 1.4

A system pole is a constant λ such that $G(s = \lambda) = \infty$. From (1.11), a system pole is a root of the characteristic polynomial of the system $G(s)$ and is also an eigenvalue of the dynamic matrix of the system. Thus the number of poles of an irreducible system is n.

Definition 1.5

In SISO systems, a system zero is a finite constant z such that $G(s = z) = 0$. From (1.11), a system zero is a root of the numerator polynomial $CV\mathrm{adj}(sI - \Lambda)V^{-1}B$ of $G(s)$, of an irreducible system.

In MIMO systems, $CV\mathrm{adj}(sI - \Lambda)V^{-1}B$ is not a scalar. Therefore, the definition of system zeros is more complicated. From Rosenbrock [1973], we define any finite constant z such that $G(s = z) = 0$ as "blocking zero." A system with blocking zero z has zero response to input $\mathbf{u}_0 e^{zt}$ for *any* $\mathbf{u}_0$.

We also define any finite constant z such that the rank of $G(s = z)$ is less than $\min\{m, p\}$ (the minimum of m and p) as "transmission zero." Thus a system with transmission zero z and with more outputs than inputs ($m > p$) has at least one constant vector $\mathbf{u}_0$ such that $G(s = z)\mathbf{u}_0 = 0$. In other words, such a system has zero response to input $\mathbf{u}_0 e^{zt}$, where $\mathbf{u}_0$ must satisfy $G(s = z)\mathbf{u}_0 = 0$. Therefore, blocking zero is a special case of transmission zero. There is no difference between blocking zeros and transmission zeros in SISO systems.

There is a clear and simple relationship between system transmission zeros and the system's state space model $(A,\ B,\ C,\ D)$ [Chen, 1984]. Because

$$\begin{bmatrix} I & 0 \\ C(zI-A)^{-1} & -I \end{bmatrix} \begin{bmatrix} zI-A & B \\ C & -D \end{bmatrix} = \begin{bmatrix} zI-A & B \\ 0 & G(s=z) \end{bmatrix}$$

hence

$$\begin{aligned} \text{rank}[S] \overset{\triangle}{=} \text{rank}\begin{bmatrix} zI-A & B \\ C & -D \end{bmatrix} &= \text{rank}\begin{bmatrix} zI-A & B \\ 0 & G(s=z) \end{bmatrix} \\ &= \text{rank}[zI-A] + \text{rank}[G(s=z)] \\ &= n + \min\{m,p\} \end{aligned} \tag{1.18}$$

In other words, transmission zero z must make the rank of matrix S (which is formed by state space model parameters) less than $n+\min\{m,p\}$. This relation is based on the assumption of irreducible systems so that z cannot be a system pole and so that rank $[zI-A]$ is guaranteed to be n.

Example 1.8

Let the transfer function of a system with three outputs and two inputs be

$$G(s) = \begin{bmatrix} 0 & (s+1)/(s^2+1) \\ s(s+1)/(s^2+1) & (s+1)(s+2)/(s^2+2s+3) \\ s(s+1)(s+2)/(s^4+2) & (s+1)(s+2)/(s^2+2s+2) \end{bmatrix}$$

From Definition 1.5, this system has a blocking zero -1 and two transmission zeros -1 and 0, but -2 is not a transmission zero.

This example shows that when a system has a different number of inputs and outputs $(p \neq m)$, its number of transmission zeros is usually much less than its number of system poles. However, when a system has the same number of inputs and outputs $(m=p)$, its number of transmission zeros is usually $n-m$. In addition, if such a system (with $m=p$) has matrix product CB nonsingular, then its number of transmission zeros is always $n-m$. These properties have been proved based on the determinant of matrix S of (1.18) [Davison and Wang, 1974].

An interesting property of transmission zeros is as follows. Suppose there are r transmission zeros of system $(A,\ B,\ C)$, then for any nonsingular matrix K which approaches infinity, among the n eigenvalues of matrix

$A - BKC$, r of them will approach each of the transmission zeros and $n - r$ of them will approach infinity [Davison, 1978].

Another interesting property of transmission zeros is that when a system $G(s)$ is connected with a dynamic feedback compensator system $H(s)$, the set of transmission zeros of the overall feedback system equals the union of the transmission zeros of $G(s)$ and the poles of $H(s)$ [Patel, 1978].

In addition, we will assign all stable transmission zeros of $G(s)$ as the poles of its corresponding dynamic feedback compensator $H(s)$ (in Chap. 5). Hence the accurate computation of transmission zeros of a given system is important.

There are several methods of computing transmission zeros of a given system [Davison and Wang, 1974; Davison, 1976, 1978; Kouvaritakis and MacFarlane, 1976; MacFarlane and Karcaniar, 1976; Sinswat et al., 1976]. The following is a brief description of the so-called QZ method [Laub and Moore, 1978]. This method computes all finite generalized eigenvalues z such that there exists an $n + p$ dimensional vector $\mathbf{w}$ satisfying

$$S\mathbf{w} = 0 \tag{1.19}$$

where matrix S is already defined in (1.18).

Equation (1.19) is valid for the case $m \geqslant p$. The transpose (or the dual) of (1.19) can be used for the case $m \leqslant p$. The advantage of this method arises from the existence of a numerically stable algorithm [Moler and Stewart, 1973] for computing the generalized eigenvalues [Laub and Moore, 1978].

We have briefly discussed the properties of system zeros. The properties of system poles will be discussed in the next chapter, which shows that the system poles are the most important parameters in determining a system's performance.

EXERCISES

1.1 For a linear time-invariant circuit system shown in Fig. 1.9:

(a) Let the currents of the two resistors be the two outputs of this system, respectively. Find the state space model (1.1) of this system.

(b) Derive the transfer function model (1.9) of this system.

(c) Plot the linear motion mechanical system which is analogous to this circuit system. Indicate all signals and elements of this mechanical system in terms of the corresponding circuit system signals and elements.

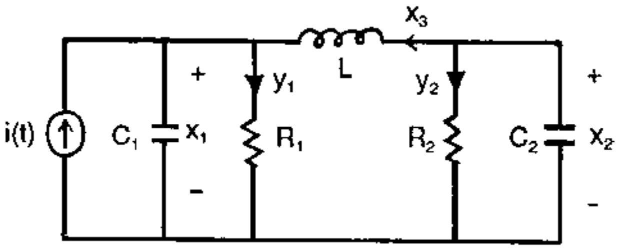


Figure 1.9 A linear time-invariant circuit system.

1.2 Let a controllable canonical form state space model be the dual from Example 1.6.

$$\overline{A} = A' \qquad \overline{B} = C' = [c_1, 0 \ldots 0]'$$
$$\overline{C} = B' = [\mathbf{b}_1' : \ldots : \mathbf{b}_n']'$$

(a) Plot the block diagram similar to Fig. 1.8.
(b) Prove that $c_1 \neq 0$ is the necessary and sufficient condition for the system $(\overline{A}, \overline{B}, \overline{C})$ to be controllable.
(c) Prove that the transfer functions of $(\overline{A}, \overline{B}, \overline{C})$ is the transpose of that from Example 1.6.

1.3 Let a two-output observable canonical form system state space model be

$$A = \begin{bmatrix} 2 & 3 & 1 \\ 4 & 5 & 0 \\ 6 & 7 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 8 \\ 9 \\ 10 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

(a) From the description from Example 1.7 (or Definition 5.1), find the observability indices $v_i \ (i = 1, 2)$.
(b) Following the two-step procedure from Example 1.7, derive the polynomial matrix fraction description of the transfer function of this system $G(s) = D^{-1}(s)N(s)$.

(c) Find the poles and zeros (if any) of this system.

Answer: $v_1 = 2, v_2 = 1, D(s) = \begin{bmatrix} s^2 - 2s - 6 & -3s - 7 \\ -4 & s - 5 \end{bmatrix}$

$$N(s) = \begin{bmatrix} 8s + 10 \\ 9 \end{bmatrix}$$

1.4 Repeat 1.3 for the system

$$A = \begin{bmatrix} a & b & 0 \\ c & d & 1 \\ e & f & 0 \end{bmatrix} \qquad B = \begin{bmatrix} g & h \\ i & j \\ k & l \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Answer: $v_1 = 1, v_2 = 2, D(s) = \begin{bmatrix} s - a & -b \\ -cs - e & s^2 - ds - f \end{bmatrix}$

$$N(s) = \begin{bmatrix} g & h \\ is - k & js + l \end{bmatrix}$$

1.5 Repeat 1.3 for the system

$$A = \begin{bmatrix} a & b & 1 & 0 \\ c & d & 0 & 1 \\ e & f & 0 & 0 \\ g & h & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} i \\ j \\ k \\ l \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Answer: $v_1 = v_2 = 2, D(s) = \begin{bmatrix} s^2 - as - e & -bs - f \\ -cs - g & s^2 - ds - h \end{bmatrix}$

$$N(s) = \begin{bmatrix} is + k \\ js + l \end{bmatrix}$$

1.6 Repeat 1.3 for the system

$$A = \begin{bmatrix} a & b & 1 & 0 \\ c & d & 0 & 0 \\ e & f & 0 & 1 \\ g & h & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} i \\ j \\ k \\ l \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Answer: $v_1 = 3, v_2 = 1$

$$D(s) = \begin{bmatrix} s^3 - as^2 - es - g & -bs^2 - fs - h \\ -c & s - d \end{bmatrix}$$

$$N(s) = \begin{bmatrix} is^2 + ks + l \\ j \end{bmatrix}$$

1.7 Let two system dynamic matrices be

$$A_1 = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & -2 \end{bmatrix} \qquad A_2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -2 \end{bmatrix}$$

Compute the Jordan form decomposition (1.10) of the two matrices.

1.8 Verify $N(s)$ [in $G(s) = D^{-1}(s)N(s)$] from Examples 6.1 and 6.3, according to the two-step procedure from Example 1.7.

2

Single-System Performance and Sensitivity

High system performance and low sensitivity are the two required properties of control systems. Low sensitivity is defined with respect to the system's mathematical model uncertainty and terminal disturbance, and is called "robustness."

Unfortunately, high performance and robustness are usually contradictory to each other—higher performance systems usually have higher sensitivity and worse robustness properties. Yet both high performance and high robustness are essential to most practical engineering systems. Usually, only high-performance systems have serious robustness problems and only such systems are worthy of controlling. Robustness, which can be considered as reliability, is also essential in most practical cases. Therefore, both performance and sensitivity properties must be studied. This chapter consists of two sections which study system performance and sensitivity properties, respectively.

Section 2.1 studies some system properties such as system stability, quickness, and smoothness of system transient response, which are most important (and most difficult to achieve) in system performance. This section explains how these properties are most directly and explicitly determined by the system poles.

Section 2.2 studies the property of system sensitivity via a novel perspective of the sensitivities of system poles. A basic result of numerical linear algebra is that the sensitivity of an eigenvalue is determined by its corresponding left and right eigenvectors.

2.1 SYSTEM PERFORMANCE

The reason that systems control theory has concentrated mainly on linear time-invariant systems is that only the mathematical models of this kind of systems can have general and explicit solutions. Furthermore, only the general and explicit understanding of the system can be used to guide generally, systematically, and effectively the complicated control system design.

The analytical solution of the state space model (1.1a) is, for $t > 0$,

$$\mathbf{x}(t) = \mathrm{e}^{At}\mathbf{x}(0) + \int_0^t \mathrm{e}^{A(t-\tau)} B\mathbf{u}\,(\tau)\; d\tau \tag{2.1}$$

where $\mathbf{x}(0)$ and $\mathbf{u}(\tau)\,(0 \leqslant \tau \leqslant t)$ are given system initial state and system input, respectively. One way of deriving this result is by taking the inverse Laplace transform on (1.6a). We call (2.1) the "complete system response" of system state $\mathbf{x}(t)$.

Substituting (1.10) into (2.1) and using the Cayley–Hamilton theorem

$$\mathbf{x}(t) = V\mathrm{e}^{\Lambda t}V^{-1}\mathbf{x}(0) + \int_0^t V\mathrm{e}^{\Lambda(t-\tau)}V^{-1}B\mathbf{u}(\tau)\; d\tau \tag{2.2}$$

$$= \left(\sum_{i=1}^{q} V_i \mathrm{e}^{\Lambda i t} T_i\right)\mathbf{x}(0) + \int_0^t \sum_{i=1}^{q} V_i \mathrm{e}^{\Lambda i(t-\tau)} T_i B\mathbf{u}(\tau)\; d\tau \tag{2.3}$$

Therefore, $\mathrm{e}^{\Lambda i t}$ $(i = 1, \ldots, q)$ are the only time function terms related to the system in the system response of (2.1)–(2.3). In other words, the eigenvalues (Λ) of system dynamic matrix A (or the system poles) are parameters which most directly and explicitly determine the system response.

Let us analyze all possible waveforms of the function $\mathrm{e}^{\Lambda_i t}$ based on the definitions (1.10) of Jordan blocks Λ_i.

$$\mathrm{e}^{\Lambda_i t} = \mathrm{e}^{\lambda_i t}, \quad \text{if } \Lambda_i = \lambda_i \tag{2.4a}$$

$$\mathrm{e}^{\Lambda_i t} = \begin{bmatrix} \mathrm{e}^{\sigma t}\cos(\omega t) & \mathrm{e}^{\sigma t}\sin(\omega t) \\ -\mathrm{e}^{\sigma t}\sin(\omega t) & \mathrm{e}^{\sigma t}\cos(\omega t) \end{bmatrix} \quad \text{if } \Lambda_i = \begin{bmatrix} \sigma & \omega \\ -\omega & \sigma \end{bmatrix} \tag{2.4b}$$

The linear combinations of the elements of this matrix can be simplified as:

$$a\mathrm{e}^{\sigma t}\cos(\omega t) + b\mathrm{e}^{\sigma t}\sin(\omega t) = (a^2 + b^2)^{1/2}\mathrm{e}^{\sigma t}\cos\left[\omega t - \tan^{-1}\left(\frac{b}{a}\right)\right]$$

where a and b are real numbers.

$$\mathrm{e}^{\Lambda_i t} = \begin{bmatrix} 1 & t & t^2/2 & \dots & t^{n-1}/(n-1)! \\ 0 & 1 & t & \dots & t^{n-2}/(n-2)! \\ 0 & 0 & 1 & \dots & t^{n-3}/(n-3)! \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & \dots & 0\ 1 \end{bmatrix} \mathrm{e}^{\lambda_i t} \tag{2.4c}$$

if Λ_i is an n-dimensional bidiagonal matrix of (1.10b).

Figure 2.1 plots all different waveforms of (2.4). In the figure, an eigenvalue (or a pole) is indicated by a symbol “x” and its coordinative position, and the corresponding waveform of this eigenvalue is plotted near that position. We can derive the following important conclusions directly from Fig. 2.1.

Definition 2.1

A system is asymptotically stable if and only if for any initial state $\mathbf{x}(0)$ the system’s zero-input response $\mathrm{e}^{At}\mathbf{x}(0)$ converges to zero.

Conclusion 2.1

From Fig. 2.1, a system is asymptotically stable if and only if every system pole (or dynamic matrix eigenvalue) has a negative real part. We will refer to “asymptotic stable” as “stable” in the rest of this book.

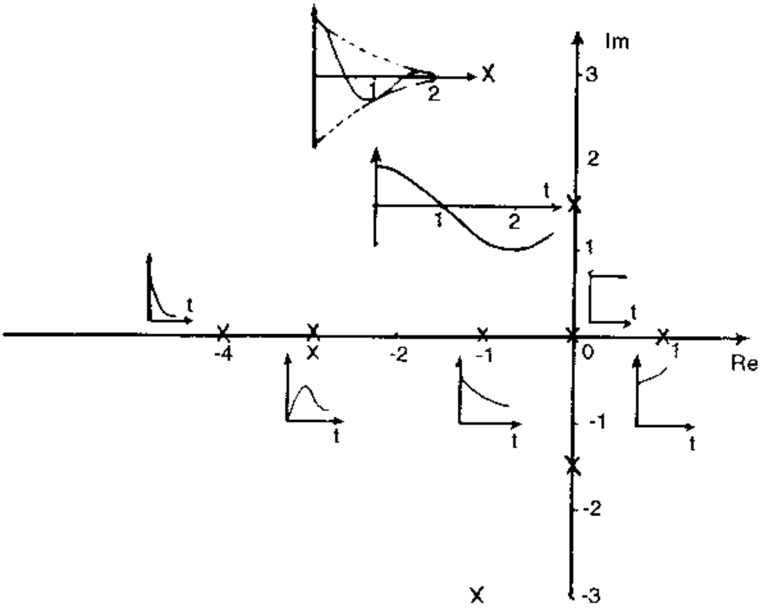


Figure 2.1 Possible system poles and waveforms of their corresponding system response.

Definition 2.2

The system response (2.1) of an asymptotically stable system always reaches a steady state, which is called "steady state response" and which is often the desired state of response. The system response (2.1) before reaching its steady state is called "transient response." Therefore, the faster and the smoother the transient response, the better (higher) the performance of the system.

Conclusion 2.2

From (2.1), the transient response is mainly determined by the term $e^{\Lambda it}$. Some conclusions about system performance can be drawn from Fig. 2.1.

(a) The more negative the real part σ of the system poles, especially the poles with least negative σ, the faster the corresponding term

$e^{\sigma t}$ converges to zero, and therefore the higher the system performance.

(b) For complex conjugate system poles, the larger the imaginary part ω of the system poles, the higher the oscillation frequency ω of the corresponding transient response, and the faster that response reaches its first zero. However, the oscillatory feature of response is generally undesirable regarding the smoothness requirement (see Definition 2.2).

(c) Multiple poles generally cause slower and rougher transient response.

We define stability, and the fastness and smoothness of the system transient response, as the main measures of system performance. Conclusions 2.1 and 2.2 indicate that the system poles determine system performance most directly, accurately, and comprehensively.

For the first-order system examples from Example 1.3, the systems are stable because their only pole λ is negative. Furthermore, the more negative the λ, the smaller the time constant $|1/\lambda|$, and the faster the zero-input response and zero-state response reach zero and steady state $(= -FB/\lambda)$, respectively. Furthermore, the first-order systems do not have multiple eigenvalues. Hence their responses are smooth.

In classical control theory, the system performance is measured by bandwidth (BW). Assume a second-order SISO system has complex conjugate poles $\sigma \pm j\omega_0$:

$$\begin{aligned} G(s) &= \frac{\omega_n^2}{[s-(\sigma+j\omega_0)][s-(\sigma-j\omega_0)]} \\ &= \frac{\omega_n^2}{s^2+(-2\sigma)s+(\sigma^2+\omega_0^2)} \\ &\overset{\triangle}{=} \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2} \end{aligned} \tag{2.5a}$$

where

$$\omega_n = (\sigma^2+\omega_0^2)^{1/2} \qquad \text{and} \qquad \zeta = -\frac{\sigma}{\omega_n}\,(0<\zeta<1) \tag{2.5b}$$

The magnitude of frequency response $|G(j\omega)|$ of this system (also called an “underdamped system”) is shown in Fig. 2.2.

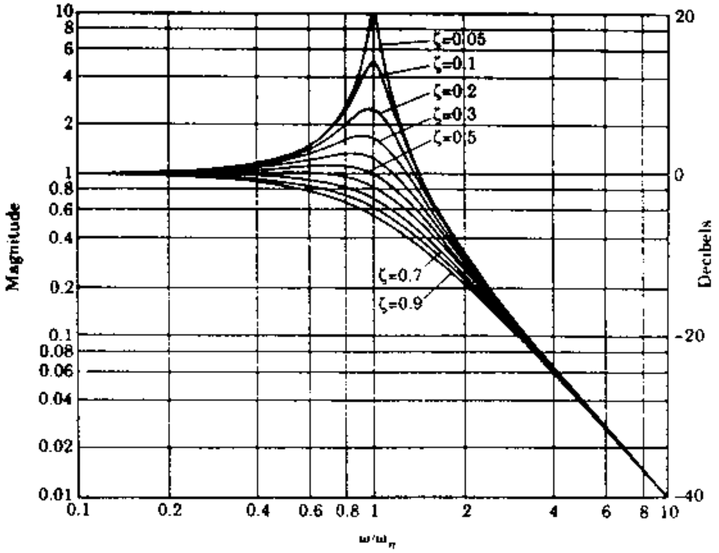


Figure 2.2 Frequency response of an underdamped system.

Figure 2.2 shows that as frequency ω increases from 0 to infinity, the function $|G(j\omega)|$ starts at 1 and eventually decays to 0. The bandwidth is defined as the frequency ω at which $|G(j\omega)| = 1/\sqrt{2} \approx 0.707$. Figure 2.2 shows that [Chen, 1993]

$$\mathrm{BW} \approx 1.6\omega_n \rightarrow 0.6\omega_n \qquad \text{when } \zeta = 0.1 \rightarrow 1 \tag{2.6}$$

In other words, BW is proportional with respect to ω_n, or $|\sigma|$ and $|\omega_0|$. Therefore from Conclusion 2.2, the wider the bandwidth, the higher the performance (generally) of the system.

However, relation (2.6) is based on a rather strict assumption (2.5) of the system, and the indication of BW is indirectly derived from Conclusion 2.2. The bandwidth, although it is simpler to measure, is generally far less accurate than the system poles in indicating the system performance. If this tradeoff in accuracy was formerly necessary because of the lack of effective computational means, the development of computer-aided design (CAD) capability has obviated this necessity.

Example 2.1 *Zero-Input Response of two Third-Order Systems*

Let the dynamic matrices of two systems be

$$A_1 = \begin{bmatrix} 1 & 2 & 0 \\ -2 & -3 & 0 \\ 0 & 1 & -2 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 2 & -1 & -2 \end{bmatrix}$$

The two matrices have the same eigenvalues $-1, -1$, and -2, but different Jordan forms.

$$\begin{aligned} A_1 = V_1 \Lambda_1 T_1 &= \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1/2 & 0 \\ -1 & -1/2 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 2 & 0 \\ 2 & 1 & 1 \end{bmatrix} \\ &= V_1 \text{ diag}\{\Lambda_{11}, \Lambda_{12}\} T_1 \\ A_2 = V_2 \Lambda_2 T_2 &= \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1/2 & 0 \\ -1 & -1/2 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 2 & 0 \\ 2 & 1 & 1 \end{bmatrix} \\ &= V_2 \text{ diag}\{\Lambda_{21}, \Lambda_{22}, \Lambda_{23}\} T_2 \end{aligned}$$

From (2.4),

$$e^{\Lambda_1 t} = \begin{bmatrix} e^{-t} & te^{-t} & 0 \\ 0 & e^{-t} & 0 \\ 0 & 0 & e^{-2t} \end{bmatrix} \quad \text{and} \quad e^{\Lambda_2 t} = \begin{bmatrix} e^{-1} & 0 & 0 \\ 0 & e^{-t} & 0 \\ 0 & 0 & e^{-2t} \end{bmatrix}$$

From (2.1)–(2.2), for a common initial state $\mathbf{x}(0) = [1 \quad 2 \quad 3]'$, the zero-input response for the state $\mathbf{x}(t)$ is

$$e^{A_1 t}\mathbf{x}(0) = \begin{bmatrix} e^{-t} + 6te^{-t} \\ 2e^{-t} - 6te^{-t} \\ -4e^{-t} - 6te^{-t} + 7e^{-2t} \end{bmatrix} \quad \text{and}$$

$$e^{A_2 t}\mathbf{x}(0) = \begin{bmatrix} e^{-t} \\ 2e^{-t} \\ -4e^{-t} + 7e^{-2t} \end{bmatrix}$$

The waveforms of these two functions are shown in Fig. 2.3.

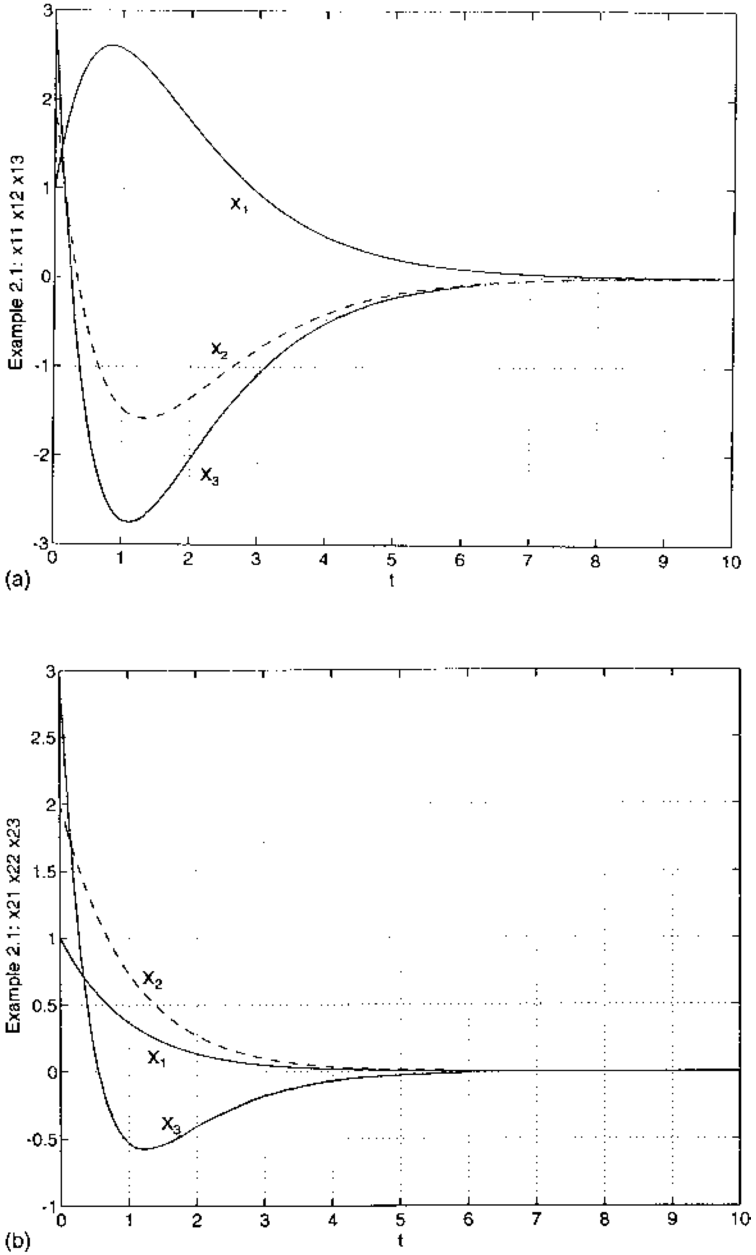


Figure 2.3 Waveforms of state zero-input responses of two systems with same poles.

The second waveform is remarkably better than the first in terms of both fastness and smoothness. This is caused by the only difference between the two systems: there is a generalized eigenvector (for eigenvalue -1) for the first dynamic matrix and none for the second dynamic matrix. This difference cannot be reflected by the transfer function model $G(s)$.

From (2.2), the difference in the function $\mathrm{e}^{\Lambda t}$ inevitably makes a difference in the system's zero-state response. Hence the state space model can also describe the zero-state response (the transient part) more explicitly than the transfer function model, even though the transfer function model is defined from the system's zero-state response only.

Example 2.2 *Zero-State Response of a Third-Order System*

Let

$$(A, B, C) = \left(\begin{bmatrix} -1 & 3 & 0 \\ -3 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 5/3 & 0 & 1 \\ 0 & 5 & 1 \end{bmatrix} \right)$$

Because matrix A is already in Jordan form, we apply (2.4a, b) directly and get

$$\mathrm{e}^{At} = \begin{bmatrix} \mathrm{e}^{-t}\cos(3t) & \mathrm{e}^{-t}\sin(3t) & 0 \\ -\mathrm{e}^{-t}\sin(3t) & \mathrm{e}^{-t}\cos(3t) & 0 \\ 0 & 0 & \mathrm{e}^{-2t} \end{bmatrix}$$

Then from (2.1), for a unit step input $(\mathbf{u}(t) = 1, t \geqslant 0)$, the zero-state response of $\mathbf{x}(t)$ is

$$\mathbf{x}(t) = \int_0^t \mathrm{e}^{A(t-\tau)} B \, d\tau = \begin{bmatrix} 3/10 + (1/\sqrt{10})\mathrm{e}^{-t}\cos(3t - 198^\circ) \\ 1/10 + (1/\sqrt{10})\mathrm{e}^{-t}\cos(3t - 108^\circ) \\ 1/2 - (1/2)\mathrm{e}^{-2t} \end{bmatrix}$$

The waveform of $\mathbf{x}(t)$ and the corresponding system output $\mathbf{y}(t) \triangleq [y_1(t)\; y_2(t)]' = C\mathbf{x}(t)$ are shown in Fig. 2.4.

The waveforms all start at zero, which conforms to the assumptions of zero initial state and of finite power input signal. The waveforms of states $x_1(t)$ and $x_2(t)$ oscillate with period $2\pi/\omega = 2\pi/3 \approx 2$ before reaching their respective steady states 0.3 and 0.1. This feature conforms with Conclusion 2.2 (Part B).

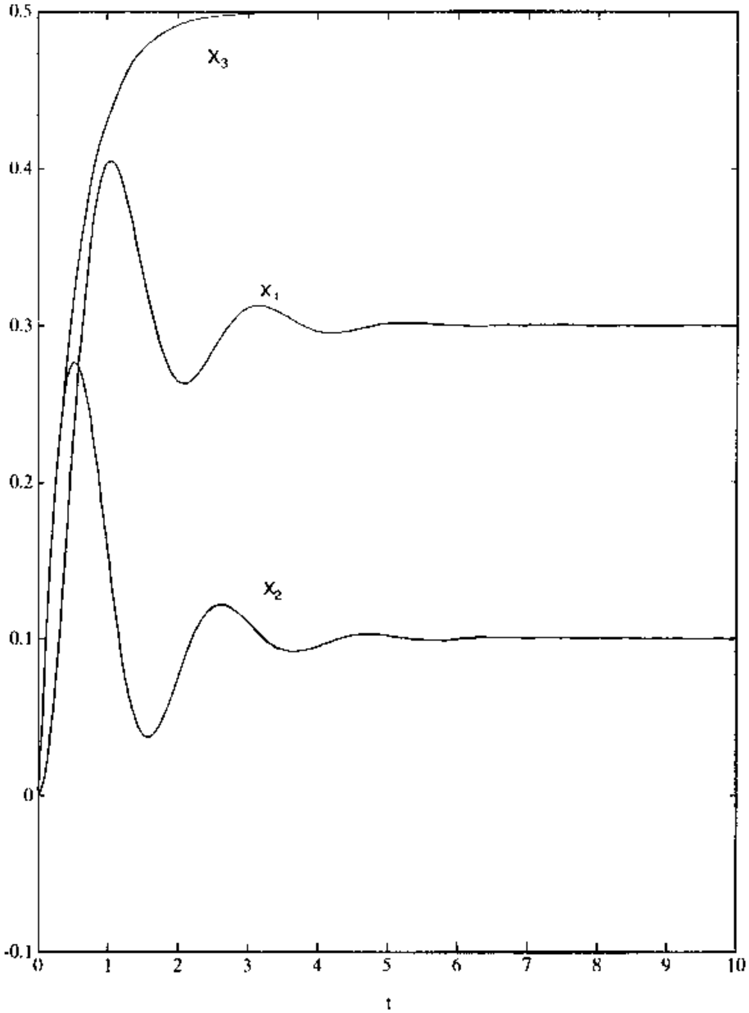


Figure 2.4 The zero-state response of system state and of system output, due to unit step input.

The above result on steady state of system output $\mathbf{y}(t)$ can also be directly derived from the system's transfer function model.

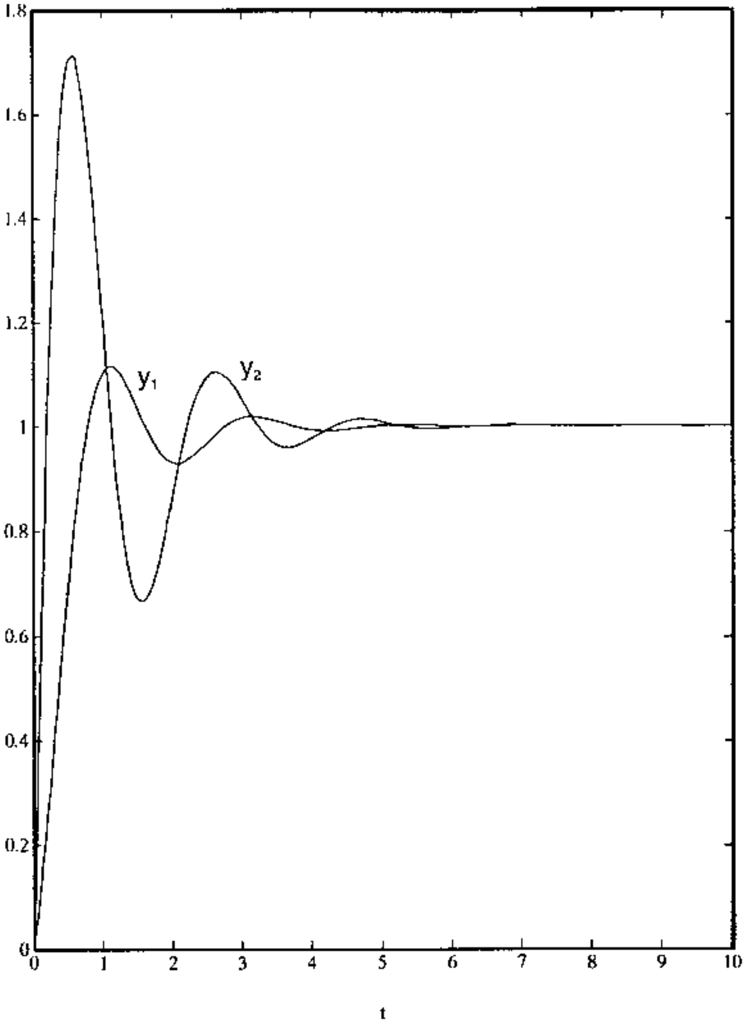


Figure 2.4 (Continued)

From (1.9),

$$G(s) = C(sI - A)^{-1}B$$
$$\triangleq \begin{bmatrix} g_1(s) \\ g_2(s) \end{bmatrix} = \frac{1}{(s^2+2s+10)(s+2)} \begin{bmatrix} s^2+7s+20 \\ 6s^2+17s+20 \end{bmatrix}$$

From (1.7), $\mathbf{y}(t)$ equals the inverse Laplace transform of $Y(s) = G(s)U(s) = G(s)/s$ (for unit step input). In addition, from the final value theorem of the Laplace transform, the constant steady state of $\mathbf{y}(t)$ can be derived directly as

$$\mathbf{y}(t \Rightarrow \infty) = \lim_{s \Rightarrow 0} sY(s) = [1 \quad 1]'$$

This result is in accordance with Fig. 2.4. This derivation shows that the classical control theory, which concentrates on system input/output relations especially at steady state, is easier than the state space control theory for deriving steady state response.

However, in measuring the transient part of this input/output relation, the bandwidths of $g_1(s)$ and $g_2(s)$ (3.815 and 9.21, respectively) are *incorrect* because $y_1(t)$ and $y_2(t)$ reach their steady state at about the same time. In addition, the waveform of $y_1(t)$ is noticeably smoother than that of $y_2(t)$ in Fig. 2.4. Overall, based on the actual step responses $y_1(t)$ and $y_2(t)$, system $g_1(s)$ is certainly much more preferable than system $g_2(s)$, yet the corresponding BW_1 is two and a half times narrower than BW_2.

2.2 SYSTEM SENSITIVITY AND ROBUSTNESS

Whereas the previous section showed the critical importance of system poles (eigenvalues of system dynamic matrix) on system performance, this section is based on a basic result of numerical linear algebra that the sensitivity of eigenvalues is determined by their corresponding eigenvectors.

Numerical linear algebra, which has not been commonly used in the existing textbooks on control systems, is a branch of study which concentrates on the sensitivity of linear algebraic computation with respect to the initial data variation and computational round-off errors [Fox, 1964]. Because linear algebra is the basic mathematical tool in linear control systems theory, the results of numerical linear algebra can be used directly in analyzing linear system sensitivities. Some basic results of numerical linear algebra have been introduced in Appendix A.

Let us first define the norm $\|A\|$ of a matrix

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

The norm of a matrix can provide a scalar measure to the magnitude of the matrix.

Before establishing the matrix norm, it is necessary to establish the norm $\|\mathbf{x}\|$ of a vector $\mathbf{x} = [x_1, \ldots, x_n]'$, where the vector elements x_i $(i = 1, \ldots, n)$ can be complex numbers. Like the absolute value of a scalar variable, the vector norm $\|\mathbf{x}\|$ must have the following three properties [Chen, 1984]:

1. $\|\mathbf{x}\| \geqslant 0$ and $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = 0$
2. $\|a\mathbf{x}\| \leqslant |a| \|\mathbf{x}\|$, where a is a scalar
3. $\|\mathbf{x} + \mathbf{y}\| \leqslant \|\mathbf{x}\| + \|\mathbf{y}\|$, where $\mathbf{y}$ is also an n-dimensional vector

The third property is also called "triangular inequality."

Definition 2.3

The vector norm $\|\mathbf{x}\|$ is defined as follows:

1. $\|\mathbf{x}\|_1 = |x_1| + \cdots + |x_n|$
2. $\|\mathbf{x}\|_2 = (|x_1|^2 + \cdots + |x_n|^2)^{1/2} = (\mathbf{x}^* \mathbf{x})^{1/2}$ ("*" stands for transpose and complex conjugate operation)
3. $\|\mathbf{x}\|_\infty = \max_i |x_i|$

In most cases only the norm $\|\mathbf{x}\|_2$ is being used. Therefore $\|\mathbf{x}\|$ is the default of vector norm $\|\mathbf{x}\|_2$ in this book unless specified otherwise.

Vector norms have the following common and important property (Cauchy–Schwartz inequality) [Chen, 1984]:

$$|\mathbf{x}^* \mathbf{y}| = |\mathbf{y}^* \mathbf{x}| \leqslant \|\mathbf{x}\| \|\mathbf{y}\| \tag{2.7}$$

The matrix norm $\|A\|$, where the entries of matrix A can be complex numbers, must also have the following four properties:

1. $\|A\| \geqslant 0$ and $\|A\| = 0$ if and only if $A = 0$
2. $\|aA\| = |a| \|A\|$, where a is a scalar
3. $\|A + B\| \leqslant \|A\| + \|B\|$, where B is a matrix of same dimension
4. $\|A\mathbf{x}\| \leqslant \|A\| \|\mathbf{x}\|$ (2.8)

Based on the above properties, especially (2.8), there can be three different definitions of matrix norm $\|A\|$ according to the three different vector norms of Definition 2.3, respectively [Chen, 1984].

Definition 2.4

$$\begin{aligned}
&1. \quad \|A\|_1 = \max_{\mathrm{j}} \{|a_{1j}| + \cdots + |a_{mj}|\} \\
&2. \quad \|A\|_2 = \max\{(\text{eigenvalue of}(A^* A))^{1/2}\} \\
&\qquad\qquad = \max\{\text{singular value of A}\} \qquad (2.9) \\
&3. \quad \|A\|_\infty = \max_i \{|a_{i1}| + \cdots + |a_{in}|\}
\end{aligned}$$

Unless specified otherwise, $\|A\|$ is the default of $\|A\|_2$, which is also called the “spectrum norm.”

There is another commonly used matrix norm $\|A\|_F$, which is called the “Frobenius norm” and is defined as follows:

$$4. \quad \|A\|_F = \left(\sum_{i,j} |a_{ij}|^2\right)^{1/2} = [\mathrm{Trace}(A^* A)]^{1/2} \qquad (2.10)$$

where the matrix operator “Trace” stands for the sum of all diagonal elements.

Based on the singular value decomposition of a matrix with $m \geqslant n$ (see Appendix A, Sec. A.3),

$$A = U\Sigma V^* = U \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^*$$

where $\Sigma_1 = \mathrm{diag}\{\text{singular values of} A : \sigma_i (i = 1, \ldots, n)\}$, $U^* U = I$, $V^* V = I$, and $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n \geqslant 0$. Then from (2.9–2.10),

$$\begin{aligned}
\|A\|_F &= [\mathrm{Trace}(\Sigma^* \Sigma)]^{1/2} \\
&= (\sigma_1^2 + \cdots + \sigma_n^2)^{1/2} \begin{cases} \leqslant \sqrt{n}\sigma_1 = \sqrt{n}\|A\|_2 & (2.11\text{a}) \\ \geqslant \sigma_1 = \|A\|_2 & (2.11\text{b}) \end{cases}
\end{aligned}$$

Equation (2.11) is useful in estimating the matrix spectrum norm.

Definition 2.5

Condition number of a computational problem:

Let A be data and $f(A)$ be the result of a computational problem $f(A)$. Let ΔA be the variation of data A and Δf be the corresponding variation of result $f(A)$ due to ΔA such that

$$f(A+\Delta A) = f(A) + \Delta f$$

Then the condition number $\kappa(f)$ of the computational problem $f(A)$ is defined by the following inequality:

$$\frac{\|\Delta f\|}{\|f\|} \leqslant \frac{\kappa(f)\|\Delta A\|}{\|A\|} \tag{2.12}$$

Therefore, $\kappa(f)$ is the relative sensitivity of problem f with respect to the relative variation of data A. A small $\kappa(f)$ implies low sensitivity of problem f, which is then called a "well-conditioned problem." On the other hand, a large $\kappa(f)$ implies high sensitivity of the problem f, which is then called an "ill-conditioned problem" [Wilkinson, 1965].

Example 2.3 [Wilkinson, 1965; Tsui, 1983b]

Let the computational problem be the computation of solution $\mathbf{x}$ of a set of linear equations $A\mathbf{x} = \mathbf{b}$, where A and $\mathbf{b}$ are given data.

Let $\Delta\mathbf{b}$ be the variation of $\mathbf{b}$ (no variation of A). Then $A(\mathbf{x}+\Delta\mathbf{x}) = (\mathbf{b}+\Delta\mathbf{b})$ implies that

$$\|\Delta\mathbf{x}\| = \|A^{-1}\Delta\mathbf{b}\| \leqslant \|A^{-1}\|\|\Delta\mathbf{b}\|$$

Thus from (2.8),

$$\|A\|\|\mathbf{x}\| \geqslant \|\mathbf{b}\|,$$
$$\frac{\|\Delta\mathbf{x}}{\|\mathbf{x}\|} \leqslant \|A\|\|A^{-1}\| \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}$$

From Definition 2.5, this inequality implies that the condition number of this problem is $\|A\|\|A^{-1}\|$.

Suppose in the same problem that ΔA is the variation of A (no variation of $\mathbf{b}$). Then $(A+\Delta A)(\mathbf{x}+\Delta\mathbf{x}) = \mathbf{b}$ implies (assuming $\|\Delta A\Delta\mathbf{x}\|$ is very small):

$$\Delta\mathbf{x} = A^{-1}(-\Delta A\mathbf{x})$$

Thus from (2.8), $\|\Delta\mathbf{x}\|/\|\mathbf{x}\| \leqslant \|A\|\|A^{-1}\|\|\Delta A\|/\|A\|$. From Definition 2.5, this inequality again implies that the condition number of this problem is $\|A\|\|A^{-1}\|$.

Because of the result of Example 2.3, we define the condition number of a matrix A as

$$\kappa(A) = \|A\|\|A^{-1}\| \tag{2.13}$$

In the following we will first analyze the sensitivity of the eigenvalues of system dynamic matrix, and then use this result to analyze the sensitivity of system stability property.

2.2.1 The Sensitivity of Eigenvalues (Robust Performance)

Robust performance is defined as the low sensitivity of system performance with respect to system model uncertainty and terminal disturbance. Because Sec. 2.1 indicated that the eigenvalues of system dynamic matrix (or system poles) most directly and explicitly determine system performance, it is obvious that the sensitivities of these eigenvalues most directly determine a system's robust performance.

From (1.10), $V^{-1}AV = \Lambda$, where matrix Λ is a Jordan form matrix with all eigenvalues of matrix A. Therefore, if A becomes $A + \Delta A$, then

$$V^{-1}(A + \Delta A)V = \Lambda + V^{-1}\Delta A V \underset{=}{\Delta} \Lambda + \Delta\Lambda \tag{2.14}$$

$$\|\Delta\Lambda\| \leqslant \|V\|\|V^{-1}\|\|\Delta A\| \underset{=}{\Delta} \kappa(V)\|\Delta A\| \tag{2.15a}$$

Inequality (2.15a) indicates that the condition number $\kappa(V)$ of eigenvector matrix V can decide the magnitude of $\|\Delta\Lambda\|$. However, $\Delta\Lambda$ is not necessarily in Jordan form, and hence may not accurately indicate the actual variation of the eigenvalues.

Based on (2.14), a result using $\kappa(V)$ to indicate the variation of eigenvalues was derived by Wilkinson (1965):

$$\min_i \{|\lambda_i - \lambda|\} \underset{=}{\Delta} \min_i \{|\Delta\lambda_i|\} \leqslant \kappa(V)\|\Delta A\| \tag{2.15b}$$

where $\lambda_i \, (i = 1, \ldots, n)$ and λ are eigenvalues of matrices A and $(A + \Delta A)$, respectively. Because the left-hand side of (2.15b) takes the minimum of the difference $\Delta\lambda_i$ between the eigenvalues of A and $A + \Delta A$, the upper bound on the right-hand side of (2.15b) does not apply to other $\Delta\lambda_i$'s.

To summarize, from (2.15), it is still reasonable to use the condition number of eigenvector matrix V of matrix A, $\kappa(V)$, to measure the sensitivity of all eigenvalues (Λ) of matrix A, $s(\Lambda)$. In other words, we define

$$s(\Lambda) \underset{=}{\Delta} \kappa(V) = \|V\|\|V^{-1}\| \tag{2.16}$$

even though $s(\Lambda)$ is not an accurate measure of the variation (sensitivity) of each individual eigenvalue. The advantage of this measure is that it is valid for large $\|\Delta A\|$ [Wilkinson, 1965].

In order to obtain a more accurate measure of the sensitivity of individual eigenvalues, first-order perturbation analysis is applied and the following result is obtained under the assumption of small $\|\Delta A\|$ [Wilkinson, 1965]:

Theorem 2.1

Let $\lambda_i, \mathbf{v}_i$, and $\mathbf{t}_i$ be the i-th eigenvalue, right and left eigenvectors of matrix A, respectively ($i = 1, \ldots, n$). Let $\lambda_i + \Delta\lambda_i$ be the i-th eigenvalue of matrix $A + \Delta A$ ($i = 1, \ldots, n$). Then for small enough $\|\Delta A\|$,

$$|\Delta\lambda_i| \leqslant \|\mathbf{t}_i\|\|\mathbf{v}_i\|\|\Delta A\| \underset{=}{\Delta} s(\lambda_i)\|\Delta A\|, \qquad i = 1, \ldots, n \tag{2.17}$$

Proof

Let $\Delta A = dB$, where d is a positive yet small enough scalar variable, and B is an $n \times n$ dimensional matrix. Let $\lambda_i(d)$ and $\mathbf{v}_i(d)(i = 1, \ldots, n)$ be the i-th eigenvalue and eigenvector of matrix $A + dB$, respectively. Then

$$(A + dB)\mathbf{v}_i(d) = \lambda_i(d)\mathbf{v}_i(d) \tag{2.18}$$

Without loss of generality, we assume $i = 1$. From the perturbation theory,

$$\lambda_1(d) = \lambda_1 + k_1 d + k_2 d^2 + \cdots \tag{2.19a}$$

and

$$\mathbf{v}_1(d) = \mathbf{v}_1 + (l_{21}\mathbf{v}_2 + \cdots + l_{n1}\mathbf{v}_n)d + (l_{22}\mathbf{v}_2 + \cdots + l_{n2}\mathbf{v}_n)d^2 + \cdots \tag{2.19b}$$

where k_j and l_{ij} ($i = 2, \ldots, n,\ j = 1, 2, \ldots$) are constants. For small enough d

(or ΔA), (2.19) can be simplified as

$$\lambda_1(d) \approx \lambda_1 + k_1 d \stackrel{\Delta}{=} \lambda_1 + \Delta\lambda_1 \tag{2.20a}$$

and

$$\mathbf{v}_1(d) \approx \mathbf{v}_1 + (l_{21}\mathbf{v}_2 + \cdots + l_{n1}\mathbf{v}_n)d \tag{2.20b}$$

Substituting (2.20) into (2.18) and from $A\mathbf{v}_i = \lambda_i \mathbf{v}_i$ and $d^2 \ll 1$, we have

$$[(\lambda_2 - \lambda_1)l_{21}\mathbf{v}_2 + \cdots + (\lambda_n - \lambda_1)l_{n1}\mathbf{v}_n + B\mathbf{v}_1]d = k_1\mathbf{v}_1 d \tag{2.21}$$

Multiplying $\mathbf{t}_1(\mathbf{t}_i\mathbf{v}_j = \delta_{ij})$ on the left of both sides of (2.21), we have

$$\mathbf{t}_1 B\mathbf{v}_1 = k_1$$

From (2.20a) and (2.8),

$$|\Delta\lambda_1| = |\mathbf{t}_1 B\mathbf{v}_1 d| \leqslant \|\mathbf{t}_1\|\|\mathbf{v}_1\|\|dB\| = \|\mathbf{t}_1\|\|\mathbf{v}_1\|\|\Delta A\|$$

The derivation after (2.18) is valid for other eigenvalues and eigenvectors. Hence the proof.

This theorem shows clearly that the sensitivity of an eigenvalue is determined by its corresponding left and right eigenvectors. From now on, we will use the notation $s(\lambda_i)$ to represent the sensitivity of λ_i, even though $s(\lambda_i)$ is not the condition number of λ_i as defined in (2.13).

Example 2.4

Consider the following two matrices:

$$A_1 = \begin{bmatrix} n & 1 & 0 & & \ldots & 0 \\ 0 & n-1 & 1 & 0 & & \vdots \\ \vdots & 0 & n-2 & 1 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & 2 & 1 \\ 0 & 0 & 0 & \ldots & 0 & 1 \end{bmatrix}$$

and

$$A_2 = \begin{bmatrix} n & n & 0 & & \dots & 0 \\ 0 & n-1 & n & 0 & & \vdots \\ \vdots & 0 & n-2 & n & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & 2 & n \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

Clearly, the two matrices have the same set of eigenvalues $\{n, n-1, \dots, 1\}$. The right and left eigenvector matrices are:

$$V = \begin{bmatrix} 1 & -x_2 & x_3 & -x_4 & \dots & (-1)^{n-1}x_n \\ 0 & 1 & -x_2 & x_3 & \dots & (-1)^{n-2}x_{n-1} \\ 0 & 0 & 1 & -x_2 & \dots & (-1)^{n-3}x_{n-2} \\ \vdots & & & \ddots & \ddots & \vdots \\ \vdots & & & & 1 & -x_2 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

and

$$T = \begin{bmatrix} 1 & x_2 & x_3 & \dots & x_n \\ 0 & 1 & x_2 & \dots & x_{n-1} \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & 1 & x_2 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

where

$$\begin{aligned} &\text{for } A_1 : x_i = x_{i-1}/(i-1) = 1/(i-1)!, i = 2, \dots, n, (x_1 = 1), \text{or} \\ &\qquad x_2 = 1, x_3 = 1/2!, x_4 = 1/3!, \dots, x_n = 1/(n-1)!; \\ &\text{for } A_2 : x_i = nx_{i-1}/(i-1) = n^{i-1}/(i-1)!, i = 2, \dots, n, (x_1 = 1), \\ &\qquad \text{or } x_2 = n, x_3 = n^2/2!, \dots, x_n = n^{n-1}/(n-1)!. \end{aligned}$$

The eigenvector matrix parameters $(x_i, i = 1, \dots, n)$ are much greater for A_2 than for A_1. From (2.17), the sensitivity of the eigenvalues of A_2 is much higher than that of A_1.

For example,

$$\begin{aligned} s(\lambda_1 = n) &= \|\mathbf{t}_1\|\|\mathbf{v}_1\| = \|\mathbf{v}_n\|\|\mathbf{t}_n\| = s(\lambda_n) \\ &= (1 + x_2^2 + \cdots + x_n^2)^{1/2}(1) \\ &= \begin{cases} (1 + (1/2)^2 + \cdots + 1/(n-1)!^2)^{1/2} \approx 1 & \text{(for } A_1\text{)} \\ (1 + (n/2)^2 + \cdots + [n^{n-1}/(n-1)!]^2)^{1/2} & \\ \approx n^{n-1}/(n-1)! & \text{(for } A_2\text{)} \end{cases} \end{aligned} \tag{2.22a}$$

$$\begin{aligned} s(\lambda_{n/2} = n/2) &= \|\mathbf{t}_{n/2}\|\|\mathbf{v}_{n/2}\| = \|\mathbf{v}_{(n/2)+1}\|\|\mathbf{t}_{(n/2)+1}\| = s(\lambda_{(n/2)+1}) \\ &= (1 + \cdots + x_{(n/2)+1}^2)^{1/2}(1 + \cdots + x_{n/2}^2)^{1/2} \\ &\approx \begin{cases} (1)(1) = 1 & \text{(for } A_1\text{)} \\ (n^{n/2}/(n/2)!)^2 & \text{(for } A_2\text{)} \end{cases} \end{aligned} \tag{2.22b}$$

The values of $s(\lambda_i)$ are much greater for A_2 than for A_1. For A_1, all $s(\lambda_i)$ values are close to 1 $(i = 1, \ldots, n)$. Thus every eigenvalue of A_1 is almost the least possibly sensitive to the parameter variation of A_1, and the computations of these eigenvalues are therefore all well conditioned. On the other hand, the $s(\lambda_i)$ for A_2 equals 5.2, 275, and 2.155×10^6 for $i = 1$ and $n = 5$, 10, and 20 respectively, and equals 6.944×10^6 and 8×10^{12} for $i = n/2$ and $n = 10$ and 20, respectively. Thus the eigenvalues (especially $\lambda_{n/2}$) of A_2 are very sensitive to the parameter variation of A_2. Therefore the computations of the eigenvalues of A_2 are ill conditioned.

The difference between matrices A_1 and A_2 is at the upper diagonal line. From Example 1.5 and (2.4c), the upper diagonal elements of A_1 and A_2 are the coupling links between the eigenvalues of A_1 and A_2. Therefore the weaker these coupling links, the smaller the norm of each row of matrix $T\,(= V^{-1})$ computed from all columns of matrix V, and the lower the sensitivity of each eigenvalue.

From another point of view, the weaker the coupling links, the weaker the effect of the matrix parameter variation on the corresponding eigenvalues (see Gerschgorin's theorem [Wilkinson, 1965]). An even more direct inspection of the original matrices A_1 and A_2 shows that the smaller these upper diagonal elements, the closer the matrices to Jordan form, and therefore the lower the sensitivity of their Jordan forms to the variation of these two matrices. This observation is not generally valid for other matrices.

To summarize, this example shows that decoupling is extremely effective in lowering eigenvalue sensitivity. It is common sense that if a system relies more heavily on more system components in order to run, then that system has higher sensitivity with respect to these system components. Although the coupled systems usually have higher performance.

The theoretical analysis on the sensitivity of eigenvalues of A_1 and A_2 can be shown by the following example of ΔA [Wilkinson, 1965; Chen, 1984]. Let

$$\Delta A = dB = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \\ d & 0 & \dots & 0 \end{bmatrix}$$

Then

$$\det[\lambda I - (A_1 + \Delta A)] = (\lambda - n) \cdots (\lambda - 2)(\lambda - 1) + d(-1)^{n-1}$$

and

$$\det[\lambda I - (A_2 + \Delta A)] = (\lambda - n) \cdots (\lambda - 2)(\lambda - 1) + d(-n)^{n-1}$$

Hence the constant coefficient of characteristic polynomial is affected by the above data variation d (or ΔA), and this effect is much more serious for A_2 than for A_1. A root locus plot (with respect to d) in Chen [1984] demonstrates the sensitivity of the eigenvalues of A_2 vs. d.

Readers can also refer to Wilkinson [1965] for more theoretical discussions on A_2. However, the comparison of A_1 and A_2 in this book offers a clearer explanation for understanding the eigenvalue sensitivity of this example.

2.2.2 The Sensitivity of System Stability (Robust Stability)

Stability is the foremost system property. Therefore the sensitivity of this property (called “robust stability”) with respect to system model uncertainty is also critically important. Consequently, a generally accurate quantitative measure of this sensitivity is also essential to guide robust stability analysis and design.

From Conclusion 2.1, the most basic and direct criterion of system stability is that every dynamic matrix eigenvalue has a negative real part. Hence the sensitivity of these eigenvalues with respect to system model uncertainty (or dynamic matrix variation) should be the most direct and critical factor in measuring the sensitivity of system stability (robust stability).

Let us compare the Routh–Hurwitz criterion of system stability, where the system characteristic polynomial must be first computed. The sensitivity of this step of computation can be as high as the direct computation of eigenvalues (see Wilkinson, 1965 and Examples 1.6 and 2.4). The Routh–Hurwitz criterion requires additional determination based on the characteristic polynomial coefficients and on the basic stability criterion of Conclusion 2.1. This indirectness will inevitably reduce the accuracy of both the stability determination and the measure of robust stability.

Let us compare another stability criterion, the Nyquist criterion. This criterion also requires two general steps. The first step plots system frequency response $G(j\omega)$ $(\omega = 0 \rightarrow \infty)$. The second step applies the Nyquist stability criterion, which is based on the basic criterion of Conclusion 2.1 and on the Cauchy integral theorem, on the plot of step one. Both steps are indirect with respect to Conclusion 2.1 and will cause inaccuracy in each step. Stability is an *internal* system property about the convergence of a *time domain* response, while the Nyquist criterion determines this property based on the information of system's *input/output terminal* relation in *frequency domain*. Because of this fundamental reason, the Nyquist criterion is very difficult to apply to multivariable systems [Rosenbrock, 1974; Hung et al., 1979; Postlethwaite et al., 1982; Doyle et al., 1992], and its corresponding robust stability measures (gain margin and phase margin) are not generally accurate [Vidyasagar, 1984].

In this book, the result of sensitivity of system poles of Sec. 2.2.1 is used to measure robust stability. Compared to the above two robust stability measures of classical control theory, this measure has not only the apparent advantage of general accuracy, but also another critical advantage—the ability to accommodate pole assignment and thus to guarantee performance. The analysis in Sec. 2.1 shows that system poles can most directly and explicitly determine the corresponding system performance.

As stated in the beginning of this chapter, performance and robustness are the two *contradictory* yet critical properties of a practical engineering system. Therefore, it would be very impractical to concentrate on only one of these two properties [such as pole assignment only or sensitivity function $[I - L(s)]^{-1}$ (see Sec. 3.1) only]. The main purpose of this book is to introduce a new design approach which can really and fully consider *both* properties.

There are three existing robust stability measures using the sensitivity of system poles. In this book they are called M_1, M_2, and M_3. Among the three measures, M_1 and M_2 were developed in the mid 1980s [Kautsky et al., 1985; Qiu and Davidson, 1986; Juang et al., 1986; Dickman, 1987; Lewkowicz and Sivan, 1988], and M_3 was developed in the early 1990s [Tsui, 1990, 1994a]. We will analyze and compare the general accuracy and the optimization feasibility of these three measures.

Let us first introduce these three measures.

$$M_1 = \min_{0 \leqslant \omega < \infty} \{\underline{\sigma}(A - j\omega I)\}, (\underline{\sigma} \text{ equals the smallest singular value}) \tag{2.23}$$

$$M_2 = s(\Lambda)^{-1} |\mathrm{Re}\{\lambda_n\}|, (|\mathrm{Re}\{\lambda_n\}| \leqslant \cdots \leqslant |\mathrm{Re}\{\lambda_1\}|) \tag{2.24}$$

$$M_3 = \min_{1 \leqslant i \leqslant n} \{\mathrm{s}(\lambda_i)^{-1} |\mathrm{Re}\{\lambda_i)|\} \tag{2.25}$$

where all eigenvalues are assumed stable ($\mathrm{Re}\{\lambda_i\} < 0, \forall i$). In addition, we assume all eigenvalues are already arbitrarily assigned for guaranteed performance.

We will analyze these three measures in the following. All three measures are defined such that the more robustly stable the system, the greater the value of its robust stability measure.

Because $\underline{\sigma}$ indicates the smallest possible norm of matrix variation for a matrix to become singular (see Theorem A.8), M_1 equals the smallest possible matrix variation norm for the dynamic matrix A to have an unstable and pure imaginary eigenvalue $j\omega$. Therefore M_1 should be a generally accurate robust stability measure.

The main drawback of M_1 seems to be its difficulty to design. For example, it is very difficult to design a matrix K such that the M_1 of matrix $A - BK$ is maximized, where matrices (A, B) are given and the eigenvalues of $A - BK$ are also prespecified to guarantee the desired performance. In the existing analysis about maximizing M_1, the only simple and analytical result is that M_1 will be at its maximum possible value ($= |\mathrm{Re}\{\lambda_n\}|$) if $\mathrm{s}(\lambda_n)$ is at its minimal value ($= 1$) [Lewkowicz and Sivan, 1988]. Unfortunately, this is impossible to achieve in most cases.

In the measure M_2, the term $|\mathrm{Re}\{\lambda_n\}|$ is obviously the shortest distance between the unstable region and the eigenvalues λ_i on Fig. 2.1. M_2 equals this distance divided (or weighted) by the sensitivity of all eigenvalue matrix Λ. The lower the sensitivity $s(\Lambda)$, the greater M_2. In other words, M_2 may be considered as the weighted distance for λ_n to become unstable, or as the likelihood margin for λ_n to become unstable.

There exist several general and systematic numerical algorithms which can compute matrix K such that the value of $s(\Lambda)^{-1}$ or M_2 is maximized, with arbitrarily assigned eigenvalues in matrix $A - BK$ [Kautsky et al., 1985; MATLAB, 1990]. However, M_2 seems to be less accurate in measuring the likelihood margin for λ_n to become unstable, because $s(\Lambda)$ is not an accurate measure of the sensitivity of λ_n [see the discussion of (2.14)].

In the definition of measure M_3, the likelihood margins for *every* eigenvalue to become unstable are considered. Here the likelihood margin for each λ_i equals $|\mathrm{Re}\{\lambda_i\}|$ divided by its corresponding sensitivity $s(\lambda_i), i = 1, \ldots, n$. In practice, the algorithms for maximizing M_2 (or $s(\Lambda)^{-1}$ $= \kappa(V)^{-1}$) can also be used to maximize M_3, after adding a weighting factor $|\mathrm{Re}\{\lambda_i\}|^{-1}$ on each column $\mathbf{v}_i$ of matrix $V, i = 1, \ldots, n$.

Based on the above analysis and some basic principles, there are two obvious reasons that M_3 is generally more accurate than M_1 and M_2.

First, M_1 and M_2 consider *only* the likelihood margin for λ_n to become unstable, while the instability of *any* eigenvalue can cause system instability (Conclusion 2.1). Therefore M_3 measures the robust stability more completely and more rigorously than M_1 and M_2.

Second, the $s(\Lambda)$ of M_2 is generally not an accurate measure of individual eigenvalue sensitivity and is obviously not as accurate as the sensitivity $s(\lambda_i)$ of λ_i itself in measuring the sensitivity of $\lambda_i, \forall_i$ (including $i = n$). Hence M_2 is too conservative compared to M_3. This is reflected in the following lower bound of M_3, even though M_3 more completely and rigorously reflects the instability likelihood of all eigenvalues.

$$\because s(\Lambda) \triangleq \|V\|\|V^{-1}\| > \|\mathbf{v}_i\|\|\mathbf{t}_i\| \overset{\Delta}{=} s(\lambda_i) \geqslant 1, i = 1, \ldots, n \tag{2.26}$$

$$\therefore M_2 = s(\Lambda)^{-1}|\mathrm{Re}\{\lambda_n\}| \leqslant M_3 \leqslant |\mathrm{Re}\{\lambda_n\}| \tag{2.27}$$

It has been proved that M_1 shares the same upper and lower bounds with M_3 [Kautsky et al., 1985; Lewkowicz and Sivan, 1988].

From (2.26–2.27), if the overall eigenvalue sensitivity $s(\Lambda) = \kappa(V)$ is at the lowest possible value ($= 1$), then all three measures M_i ($i = 1, 2, 3$) will reach their common highest possible value $|\mathrm{Re}\{\lambda_n\}|$. However, it is impossible to make $s(\Lambda) = 1$ for most cases. In those cases, a lower $s(\Lambda)$ does not necessarily imply a higher M_1 or M_3 [Lewkowicz and Sivan, 1988]. Furthermore, in those cases, (2.27) implies that M_1 and M_3 have higher resolution and therefore higher accuracy than M_2.

Example 2.5 [Lewkowicz and Sivan, 1988; Tsui, 1994a]

Let

$$A_1 = \begin{bmatrix} -3 & 0 & 0 \\ 4.5 & -2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} -3 & 0 & 0 \\ 1.5 & -2 & 0 \\ 3 & 0 & -1 \end{bmatrix}$$

The two matrices have same eigenvalues but different eigenvectors. Hence the eigenvalue sensitivity as well as the robust stability are different for these two matrices.

The eigenstructure decomposition of these two matrices are

$$\begin{aligned} A_1 &= V_1 \Lambda_1 T_1 \\ &= \begin{bmatrix} -0.217 & 0 & 0 \\ -0.976 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -3 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 4.61 & 0 & 0 \\ 4.5 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} (\|\mathbf{t}_1\| = 4.61) \\ (\|\mathbf{t}_2\| = 4.61) \\ (\|\mathbf{t}_3\| = 1) \end{matrix} \end{aligned}$$

and

$$\begin{aligned} A_2 &= V_2 \Lambda_2 T_2 \\ &= \begin{bmatrix} 0.4264 & 0 & 0 \\ -0.6396 & 1 & 0 \\ -0.6396 & 0 & 1 \end{bmatrix} \begin{bmatrix} -3 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 2.345 & 0 & 0 \\ 1.5 & 1 & 0 \\ 1.5 & 0 & 1 \end{bmatrix} \begin{matrix} (\|\mathbf{t}_1\| = 2.345) \\ (\|\mathbf{t}_2\| = 1.803) \\ (\|\mathbf{t}_3\| = 1.803) \end{matrix} \end{aligned}$$

In the above result, the norm of every right eigenvector in V matrix equals one. Thus from (2.17), the eigenvalue sensitivity $s(\lambda_i)$ equals the norm of the corresponding left eigenvector $\|\mathbf{t}_i\|$, which has been listed along with the corresponding vector above.

Based on this result, the values of $M_i \, (i = 1, 2, 3)$ are calculated in Table 2.1.

The inspection of the two matrices shows that unlike in A_2, the $\lambda_n \, (= -1)$ in A_1 is completely decoupled and thus has sensitivity $s(-1) = 1$. This feature is reflected by M_1, which reaches its maximal value for A_1 and is considered by M_3 also. Also, unlike A_2, A_1 has a large element (4.5) which causes higher sensitivity of other two adjacent eigenvalues $(-2, -3)$ of A_1 as well as a higher value of $s(\Lambda)$. This feature is reflected by a smaller value of M_2 for A_1 and is considered by M_3 also. Therefore, *only* M_3 can comprehensively reflect these two conflicting features about robust stability.

Table 2.1 Robust Stability Measurements of Two Dynamic Matrices

	A_1	A_2
M_1	1	0.691
$M_2 = s(\Lambda)^{-1}\lvert -1 \rvert$	0.1097	0.2014
$s(-1)^{-1}$	1	0.5546
$s(-2)^{-1}$	0.2169	0.5546
$s(-3)^{-1}$	0.2169	0.4264
M_3	$s(-2)^{-1}\lvert -2 \rvert = 0.4338$	$s(-1)^{-1}\lvert -1 \rvert = 0.5546$

From the definition of M_3, for matrix A_1, eigenvalue -2 has the shortest likelihood margin (0.4338) to instability and therefore is most likely to become unstable, even though the eigenvalue $\lambda_n = -1$ is closest to unstable region. Thus for matrix A_1, M_3 has considered accurately the low sensitivity of its λ_n while M_2 has not, and M_3 has considered the high sensitivity of other eigenvalues while M_1 has not.

Overall, matrices A_1 and A_2 are quite similar—with one element 4.5 in A_1 being divided into two elements (1.5 and 3) in A_2. Hence a reasonable robust stability measure should not differ too much for these two matrices. We notice that this is the case for M_3 but not for M_1 or M_2.

This example shows quite convincingly that M_3 is considerably more accurate than M_1 and M_2.

Although maximizing M_2 or minimizing $s(\Lambda)(= \|V\|\|V^{-1}\|)$ may not improve robust stability as directly as maximizing M_3, it also implies in a simple, scalar, and unified sense the improvement of other system aspects such as the lowering of feedback control gain $\|K\|$ and the smoothing of transient response (see Chap. 8 and Kautsky, 1985). Both aspects are very important, especially when the dynamic matrix eigenvalues are already assigned.

We have mentioned that the numerical algorithms used to minimize $s(\Lambda)$ can also be used to maximize M_3. In addition, there is an analytical method for improving M_3. This method is based on the possibility of simple decoupling of the feedback system eigenstructure into p blocks (p = number of system inputs). The decoupling is extremely effective in improving the system's robustness. For example, the eigenvalue -1 of Example 2.5 is completely decoupled in matrix A_1 and thus has the lowest possible sensitivity. Example 2.4 also shows convincingly the strong effect of coupling on eigenvalue sensitivity.

CONCLUSION

State space control theory provides distinctly general, accurate, and clear analysis on linear time-invariant systems, especially their performance and sensitivity properties. *Only* this kind of analysis and understanding can be used to guide generally and effectively the design of complex control systems. This is the reason that linear time-invariant system control results form the basis of the study of other systems such as nonlinear, distributive, and time-varying systems, even though most practical systems belong to the latter category.

This is also the reason that the development of state space control theory has always been significant and useful. For example, because of the lack of accurate measure of system performance and robustness, the direct design of loop transfer function has not been generally effective (see also the end of Secs. 3.1 and 9.3). Starting with the next chapter, we will see that there are basic, practical, and significant design problems which can only now be solved satisfactorily using state space techniques.

EXERCISES

2.1 Let the dynamic matrices of two systems be

$$A_1 = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & -2 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -2 \end{bmatrix}$$

(a) Based on the eigenstructure decomposition of Exercise problem 1.7 and based on (2.4), derive the time function $e^{A_i t}, i = 1, 2$.

(b) Derive $e^{A_i t}$ using $e^{A_i t} = \mathscr{L}^{-1}\{(sI - A_i\}^{-1}\}(i = 1, 2)$.

(c) Derive zero-input response $e^{A_i t}\mathbf{x}_i(0)$ with $\mathbf{x}_i(0) = [1 \quad 2 \quad 3]'$ $(i = 1, 2)$. Plot and compare the wave forms of these two responses.

2.2 Repeat 2.1 for the two matrices from Example 2.5.

2.3 Consider the system

$$A_1, B = [0 \quad 1 \quad -1]' \quad \text{and} \quad C = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}$$

where matrix A_1 is similar to that of 2.1 above.

(a) Using (2.1) and the result of 2.1, derive the zero-state response of the two outputs of this system for unit step input.
(b) Using (1.7), $\mathbf{y}(t) = \mathscr{L}^{-1}\{Y_{zs}(s) = G(s)U(s) = G(s)/s\}$ derive $\mathbf{y}(t)$, where $G(s)$ is derived based on (1.9).
(c) Compute the bandwidth for the two elements of $G(s)$.
(d) Plot and compare the waveforms of the two outputs.

2.4 Analyze the robust stability of the two systems from Example 2.1.

Notice that for eigenvalue λ within a Jordan block larger than 1×1, the corresponding sensitivity $s(\lambda)$ should be modified from (2.17) [Golub and Wilkinson, 1976]. A simple method is to add together sensitivities (2.17) of all eigenvalues within a same Jordan block.

For example, in matrix A_1, suppose the first two left and right eigenvectors are $\mathbf{t}_1, \mathbf{t}_2, \mathbf{v}_1, \mathbf{v}_2$ and correspond to a multiple eigenvalue $\lambda_1\ (= -1)$ in a 2×2 Jordan block, then

$$s(\lambda_1) = \|\mathbf{t}_1\|\|\mathbf{v}_1\| + \|\mathbf{t}_2\|\|\mathbf{v}_2\|(= (1)(\sqrt{3}) + (\sqrt{8})(\sqrt{1/2}))$$

2.5 Repeat 2.4 for the two dynamic matrices from 2.1.
2.6 Verify the expression (2.22) from Example 2.4.
2.7 Verify the conclusion from (2.26) to (2.27).
2.8 Repeat Exercises 2.1 and 2.4 for the following dynamic matrices. Compare the results.

$$\begin{bmatrix} -3 & 0 & 0 \\ 3 & -2 & 0 \\ 1.5 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -3 & 0 & 0 \\ -3 & -2 & 0 \\ 1.5 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -3 & 0 & 0 \\ 2 & -2 & 0 \\ 2.5 & 0 & -1 \end{bmatrix},$$
$$\begin{bmatrix} -3 & 0 & 0 \\ 2.5 & -2 & 0 \\ 2 & 0 & -1 \end{bmatrix}, \begin{bmatrix} -3 & 0 & 0 \\ -2 & -2 & 0 \\ 2.5 & 0 & -1 \end{bmatrix}$$

3 Feedback System Sensitivity

The feedback system discussed in this book consists of two basic subsystem components—an “open-loop system,” which contains the given “plant system,” and a feedback controller system, called a “compensator.” Hence the analysis of such feedback systems is different from that of a single system.

Of the two critical properties of performance and low sensitivity (robustness) of feedback systems, sensitivity has been less clearly analyzed in state space control theory. It is analyzed in this chapter, which is divided into two sections.

Section 3.1 highlights a concept in classical control theory about feedback system sensitivity—the decisive role of loop transfer function in the sensitivity of feedback systems. This concept will guide the design throughout this book, even though the focus remains on state space models of the systems.

Section 3.2 analyzes the sensitivity properties of three basic and existing feedback control structures of state space control theory—direct state feedback, static output feedback, and observer feedback. The emphasis is on the observer feedback structure, which is more commonly used than other two structures. A key design requirement on the robustness property of this structure, called loop transfer recovery (LTR), is introduced.

3.1 SENSITIVITY AND LOOP TRANSFER FUNCTION OF FEEDBACK SYSTEMS

The basic feedback control structure studied by control systems theory is shown in Fig. 3.1.

In this system structure, there is a feedback path from the plant system output $Y(s)$ to input $U(s)$ through a general feedback controller system, called "compensator" $H(s)$. Here $R(s)$ and $D(s)$ are Laplace transforms of an external reference signal $\mathbf{r}(t)$ and an input disturbance signal $\mathbf{d}(t)$, respectively.

The plant system, which is subject to control, is either $G(s)$ itself or a component system of $G(s)$ and with output $Y(s)$. In this book, we will generally treat the plant system as $G(s)$. Hence the controller to be designed is $H(s)$.

The structure of Fig. 3.1 is very basic. For more complicated control system configurations, the analysis and design is usually carried out block by block and module by module, with each block (or module) structured like Fig. 3.1.

Because input $U(s)$ can control the behavior of output $Y(s)$, such input is called "control input signal." Because the control signal usually

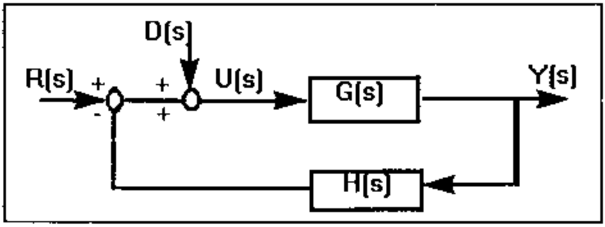


Figure 3.1 The basic structure of feedback (closed-loop) systems.

requires a large amount of power, there will very likely be disturbance associated with the generation of $U(s)$. This disturbance is commonly treated in the system's mathematical model of Fig. 3.1 as an additional signal $D(s)$.

The purpose and requirement of control systems is generally the control of plant system output (or response) $Y(s)$ so that it can quickly reach and stabilize to its desired state, such as the desired vehicle and engine speed, the desired radar and airborne system angle, the desired robot arm position, the desired container pressure and temperature, etc. The desired system output state is usually specified by the reference signal $R(s)$. Hence how well the system output reaches its desired state determines the performance of the system.

The final steady state of system response is relatively easy to analyze (using the final value theorem for example) and relatively easy to satisfy via feedback control design. Hence the transient response properties (such as the convergent speed) are critical factors to system performance and are the main challenges of feedback control system design.

The most basic feature of the feedback control system structure of Fig. 3.1 is that the control signal $U(s)$, which controls signal $Y(s)$, is itself controlled based on $Y(s)$. This feedback of $Y(s)$ to $U(s)$ creates a loop which starts and ends at $U(s)$, and whose transfer function called "loop transfer function" is

$$L(s) = -H(s)G(s) \tag{3.1}$$

We therefore call the feedback system a "closed-loop system." On the other hand, a system without feedback [or $H(s) = 0$] is called an "open-loop system." Figure 3.2 shows a block diagram where the control signal $U(s)$ is not influenced by its control object $Y(s)$. The loop transfer function of this system is

$$L(s) = 0 \tag{3.2}$$

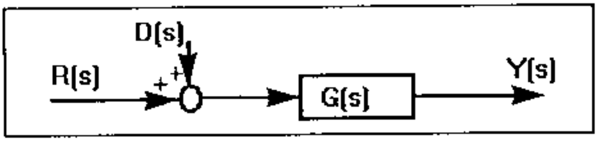


Figure 3.2 The structure of open-loop systems.

A main difference between feedback control and control without feedback concerns the sensitivity to the plant system mathematical model uncertainty, defined as $\Delta G(s)$, and to the control input disturbance $D(s)$. This section shows that this difference is determined almost solely by the loop transfer function $L(s)$, which is created by the feedback configuration itself.

To simplify the description of this concept, only SISO systems are studied in this section. However, this basic and simple concept is general to MIMO systems as well.

3.1.1 Sensitivity to System Model Uncertainty

In most practical situations, the given mathematical model (either state space or transfer function) of the plant system is inaccurate. This is because the practical physical system is usually nonlinear, and its parameters are usually distributive and are difficult to measure accurately. Even for an initially accurate model, the actual plant system will inevitably experience wear-out and accidental damage, both of which can make the mathematical model of the plant system inaccurate.

To summarize, there is an inevitable difference between the actual plant system and its mathematical model $G(s)$. This difference is called "model uncertainty" and is defined as $\Delta G(s)$. Therefore, it is essential that the control systems, which are designed based on the given available mathematical model $G(s)$, have low sensitivity to $\Delta G(s)$.

In single-variable systems, the transfer function from $R(s)$ to $Y(s)$ of control systems of Figs 3.1 and 3.2 are, respectively

$$T_c(s) = \frac{G(s)}{1 + H(s)G(s)} \tag{3.3a}$$

and

$$\mathrm{T}_o(s) = G(s) \tag{3.3b}$$

Let $\Delta T(s)$ be the uncertainty of overall control system $T(s)$ caused by $\Delta G(s)$. We will use relative plant system model uncertainty $\Delta G(s)/G(s)$ and relative control system uncertainty $\Delta T(s)/T(s)$ to measure the overall control system sensitivity vs. plant system model uncertainty.

Definition 3.1

The sensitivity of a control system $T(s)$ to $\Delta G(s)$ is defined as

$$s(T)|_G = \left|\frac{\Delta T(s)/T(s)}{\Delta G(s)/G(s)}\right| \tag{3.4a}$$

For small enough $\Delta G(s)$ and $\Delta T(s)$,

$$s(T)|_G \approx \left|\frac{\partial T(s)}{\partial G(s)}\frac{G(s)}{T(s)}\right| \tag{3.4b}$$

Equation (3.4b) is the general formula for determining $s(T)|_G$.

Substituting (3.3a) and (3.3b) into (3.4b), we have

$$s(T_c)|_G = \left|\frac{1}{1+H(s)G(s)}\right| = \left|\frac{1}{1-L(s)}\right| \tag{3.5a}$$

and

$$s(T_o)|_G = 1 \tag{3.5b}$$

A comparison of (3.5a) and (3.5b) shows clearly that the sensitivity to the plant system model uncertainty of a closed-loop system can be much lower than that of the open-loop system. The difference is determined solely by loop transfer function $L(s)$.

Example 3.1 Sensitivity to the Uncertainty of Some Individual Plant System Parameters

Let

$$G(s) = \frac{K}{s+\lambda}$$

and

$$H(s) = 1$$

Then from (3.3),

$$T_c(s) = \frac{K}{s+\lambda+K}$$

and

$$T_o(s) = \frac{K}{s+\lambda}$$

Thus from (3.4b),

$$\begin{aligned}
s(T_c)|_K &= \left|\frac{\partial T_c(s)}{\partial K}\,\frac{K}{T_c(s)}\right| = \left|\frac{s+\lambda+K-K}{(s+\lambda+K)^2}\,\frac{K}{K/(s+\lambda+K)}\right| \\
&= \left|\frac{1}{1-L(s)}\right| \\
s(T_o)|_K &= \left|\frac{\partial T_o(s)}{\partial K}\,\frac{K}{T_o(s)}\right| = \left|\frac{1}{(s+\lambda)}\,\frac{K}{K/(s+\lambda)}\right| = 1 \\
s(T_c)|_\lambda &= \left|\frac{\partial T_c(s)}{\partial \lambda}\,\frac{\lambda}{T_c(s)}\right| = \left|\frac{-K}{(s+\lambda+K)^2}\,\frac{\lambda}{K/(s+\lambda+K)}\right| \\
&= \left|\frac{-\lambda/(s+\lambda)}{1-L(s)}\right|
\end{aligned}$$

and

$$\begin{aligned}
s(T_o)|_\lambda &= \left|\frac{\partial T_o(s)}{\partial \lambda}\,\frac{\lambda}{T_o(s)}\right| = \left|\frac{-K}{(s+\lambda)^2}\,\frac{\lambda}{K/(s+\lambda)}\right| \\
&= \left|\frac{-\lambda}{(s+\lambda)}\right|
\end{aligned}$$

Therefore, the sensitivity to either plant system parameter K or λ of a closed-loop system equals that of an open-loop system divided by $1 - L(s)$. For open-loop systems, at $s = 0$, this sensitivity equals $1 = 100\%$, which is quite high.

3.1.2 Sensitivity to Control Input Disturbance

As introduced in the beginning of this section, disturbance $D(s)$ associated with the generation of large power control input $U(s)$ is serious and inevitable. Therefore, a practical control system must have low sensitivity to $D(s)$.

In practice, the controller which actually generates and asserts the control is usually called the "actuator."

From the superposition principle of linear systems, in the presence of disturbance $D(s)$ $(\neq 0)$, the closed-loop system (Fig. 3.1) and open-loop system (Fig. 3.2) responses are

$$Y_c(s) = \frac{G(s)}{1 - L(s)} R(s) + \frac{G(s)}{1 - L(s)} D(s) \tag{3.6a}$$

and

$$Y_o(s) = G(s)R(s) + G(s)D(s) \tag{3.6b}$$

respectively.

If among the respective two terms of (3.6a, b) the first term is the desired control system response which follows $R(s)$ and which is the response when $D(s) = 0$, then the second term is the deviation from the desired response and is the sole effect of disturbance $D(s)$. Therefore, the gain (magnitude of transfer function) of this second term represents the sensitivity of the corresponding system to $D(s)$. The higher the gain, the higher the sensitivity to $D(s)$.

Definition 3.2

A system's sensitivity to its control input disturbance is represented by its gain from this disturbance to its output.

Similar to the conclusions from Subsection 3.1.1, a comparison of the second terms of (3.6a) and (3.6b) shows clearly that the sensitivity to control input disturbance of closed-loop systems can be much lower than that of open-loop systems. The difference is an additional denominator $1 - L(s)$, which is determined solely by loop transfer function $L(s)$.

Example 3.2 Sensitivity to Output Measurement Noise

It is important to measure the sensitivity to output measurement noise. In practical feedback control systems, besides the undesirable effect of control input disturbance, there is another common and undesirable effect, caused by output measurement noise. This noise is represented in the mathematical model as an additional signal $N(s)$ to $Y(s)$, and in the block diagram of

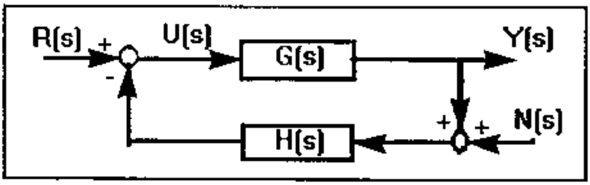


Figure 3.3 Feedback control system with output measurement noise.

Fig. 3.3, which shows a feedback control system with output measurement noise.

In many practical analog systems, especially nonelectrical systems, the signal $Y(s)$, such as velocity, angle, pressure, and temperature, is very difficult to measure accurately. In addition, the implementation of feedback control often requires that the measured analog signal be transformed to a different analog signal such as an electrical signal. The device that performs this operation is called a "transducer." Such operations can also introduce error. Because the presence of output measurement noise is almost inevitable, a feedback system must have low sensitivity to such noise.

The purpose of measuring the feedback system output $Y(s)$ is to help generate a desirable control $U(s)$, so the undesirable effect of system output measurement noise is reflected mainly in its effect on $U(s)$.

Applying Mason's formula to the system in Fig. 3.3, when $R(s) = 0$,

$$U(s) = \frac{-H(s)}{1 + H(s)G(s)} N(s) = \frac{-H(s)}{1 - L(s)} N(s) \tag{3.7}$$

This is the effect of $N(s)$ on $U(s)$. Similar to Definition 3.2, lower magnitude of the transfer function of (3.7) implies lower sensitivity against $N(s)$.

It is clear from (3.7) that the sensitivity to $N(s)$ is very much related to the loop transfer function $L(s)$. For example, from (3.7), in open-loop systems which have no feedback $[H(s) = L(s) = 0]$ and in which the measurement of $Y(s)$ does not affect the system, the sensitivity to the output measurement noise $N(s)$ is zero.

Substituting (3.7) into $Y(s) = G(s)U(s)$,

$$Y(s) = \frac{-G(s)H(s)}{1 - L(s)} N(s) = \frac{L(s)}{1 - L(s)} N(s) \tag{3.8}$$

This is the effect of $N(s)$ on system output $Y(s)$.

In the analysis of feedback system sensitivity to plant system model uncertainty and control input disturbance, it seems that the higher the loop gain $|L(s)|$, the lower the sensitivity. However, Example 3.2 shows that a high $|L(s)|$ or a large $|H(s)|$ does not lower the sensitivity to output measurement noise at all. In fact, it is *equally undesirable* to indiscriminately increase the loop gain $|L(s)|$ because of the following three reasons.

1. A high loop gain is likely to cause feedback system instability, from root locus results. This is especially true for plant systems either with pole-zero excess exceeding two or with unstable zeros.
2. A high loop gain $|L(s)|$ can generally reduce system performance. From (3.3a) and the definition of bandwidth of Sec. 2.1, a higher $|L(s)|$ often implies a lower overall feedback system gain $|T_c(s)|$ and therefore a narrower bandwidth.
3. A high loop gain or a high controller gain $|H(s)|$ is more difficult to implement in practice. A system with higher gain generally consumes more control energy and is more likely to inflict disturbance and cause failure.

Because of the above three reasons, the loop gain $|L(j\omega)|$ is shaped only at certain frequency bands. For MIMO systems, the loop gain is represented by the largest singular value of the $p \times p$ dimensional matrix $L(j\omega)$ [Doyle and Stein, 1981; Zhou et al., 1995].

However, as described in Sec. 2.1, bandwidth is far less direct and far less generally accurate in reflecting system performance. Subsections 2.2.2 and 3.2.1 (at the end) also indicated that robust stability is far less generally accurately measured by the loop transfer function based gain margins and phase margins. In addition, the loop-shaping operation, though it is already very complicated, is less refined than state space design methods in terms of how fully the available design freedom is utilized. For example, it seems that only the gain (but *not* the phase angle) of loop transfer function is considered by this operation.

To summarize, the critical factor of feedback system sensitivity is the system loop transfer function itself, but not the high gain or only the gain, of this loop transfer function.

3.2 SENSITIVITY OF FEEDBACK SYSTEMS OF MODERN CONTROL THEORY

Section 3.1 described the critical importance of loop transfer function for feedback system sensitivity. The same concept will be used to analyze the sensitivity of three existing and basic feedback control structures of state space control theory. These three structures are state feedback, static output feedback, and observer feedback. Of the three structures, observer feedback system structure is much more commonly used than the other two.

Because loop transfer function is determined by the internal feedback system structure, from now on we will let the external system reference signal $\mathbf{r}(t) = 0$. In addition, we will assume that the plant system $G(s)$ is irreducible.

3.2.1 State Feedback Control Systems

The state feedback control systems (or direct state feedback systems) have a control signal $\mathbf{u}(t)$ of

$$\mathbf{u}(t) = -K\mathbf{x}(t) \tag{3.9}$$

where $\mathbf{x}(t)$ is the system state vector, and K, which is called the "state feedback gain" or "state feedback control law," is constant. The block diagram of this feedback control structure is shown in Fig. 3.4.

It is clear from Fig. 3.4 that the loop transfer function of this system is

$$L(s) = -K(sI - A)^{-1}B \underset{=}{\Delta} L_{Kx}(s) \tag{3.10}$$

Substituting (3.9) into (1.1a), the dynamic equation of this feedback system becomes

$$\dot{\mathbf{x}}(t) = (A - BK)\mathbf{x}(t) + B\mathbf{r}(t) \tag{3.11}$$

Hence matrix $A - BK$ is the dynamic matrix of the corresponding direct state feedback system, and its eigenvalues are the poles of that feedback system.

From Sec. 1.1, system state provides the most explicit and detailed information about that system. Therefore state feedback control, if designed

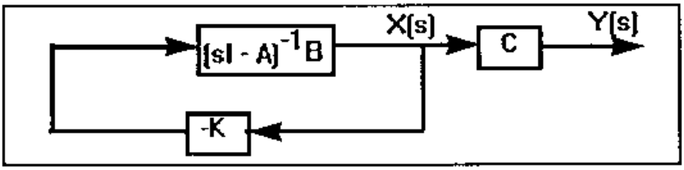


Figure 3.4 Direct state feedback systems.

properly, should be most effective in improving system performance and robustness properties, even though this design is not aimed at shaping the loop transfer function $L_{Kx}(s)$ directly.

Theorem 3.1

For any controllable plant system, the direct state feedback control can assign arbitrary eigenvalues to matrix $A - BK$, and the direct state feedback system remains controllable.

Proof

Any controllable system is similar to its corresponding block-controllable canonical form, which is the dual version (A', C') of its corresponding block-observable canonical form of (A, C) of (1.16).

The form of (1.16) implies that there exists a matrix K' such that all unknown parameters of matrix $A - K'C$ can be arbitrarily assigned, and that $A - K'C$ remains to be in observable canonical form for any K'. Hence the eigenvalues of matrix $A - K'C$ can be arbitrarily assigned and the system $(A - K'C, C)$ remains to be observable for any K'.

From the duality phenomenon, the above conclusions imply that the eigenvalues of matrix $A' - C'K$ can be arbitrarily assigned, and that system $(A' - C'K, C')$ remains to be controllable for any K.

However, matrix $A - BK$ in general cannot preserve the block-observable canonical form of the original matrix A. Hence direct state feedback system cannot preserve the observability property of the original open-loop plant system (A, B, C).

In addition, eigenvectors can also be assigned if $p > 1$, thus achieving robustness (see Sec. 2.2). The explicit design algorithms of state feedback control for eigenvalue/vector assignment will be introduced in Chap. 8.

Besides the ability to assign arbitrary poles and the corresponding eigenvectors to the feedback system, state feedback control can also realize a so called "linear quadratic optimal control," whose design will be introduced in Chap. 9. It has been proved that the loop transfer function $L_{Kx}(s)$ of such control systems satisfies the "Kalman inequality" such that

$$[I - L_{Kx}(j\omega)]^* R[I - L_{Kx}(j\omega)] \geqslant R \quad \forall\omega \tag{3.12a}$$

where R is symmetrical positive definite $(R = R' > 0)$ [Kalman, 1960].

Based on (3.12a), it has been proved that for $R = rI(r > 0)$,

$$\sigma_i[I - L_{Kx}(j\omega)] \geqslant 1 \quad \forall \omega \tag{3.12b}$$

where σ_i $(i = 1, \ldots, p)$ is the i-th singular value of the matrix. From (3.12b), the values of gain margin and phase margin of the feedback system are $1/2 \to \infty$ and $\geqslant 60^\circ$, respectively [Lehtomati et al., 1981].

The SISO case of the above result can be shown in Fig. 3.5.

The shaded area of Fig. 3.5 indicates all possible values of $-L_{Kx}(j\omega)$ that satisfy (3.12b). It is clear that the margin between these values and the -1 point is at least $1/2$ to ∞ in magnitude, and 60° in phase angle. Since according to the Nyquist stability criterion, the number of encirclements of the -1 point determines feedback system stability, this result implies a good robust stability of quadratic optimal feedback systems.

Notice that at this good robust stability, no large gain (distance to the origin) of $L_{Kx}(j\omega)$ is required at all.

However, as will be introduced at the beginning of Chap. 9, the linear quadratic optimal control systems can be formulated to have poor robustness (such as the minimum time problem). Yet the gain margin and phase margin indicate good robustness for all such systems. This is another proof that the gain margins and phase margins are *not* generally accurate measures of system robustness (see Subsection 2.2.2).

The main drawback of direct state feedback control is that it cannot be generally implemented. In most practical plant systems, only the terminal inputs and outputs of the system are directly measurable; not the entire set of internal system states. In other words, the available information about most practical systems cannot be as complete and explicit as for system

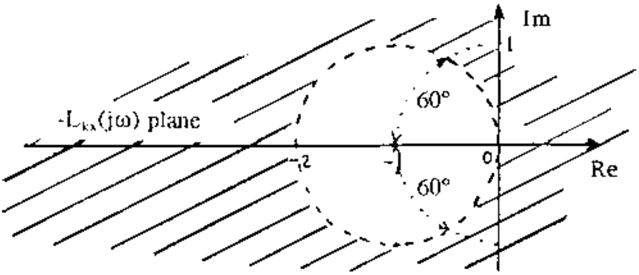


Figure 3.5 Loop transfer frequency response of single-input quadratic optimal control systems.

states. Therefore, direct state feedback control should be considered only as an ideal and theoretical form of control.

3.2.2 Static Output Feedback Control Systems

In static output feedback control systems, the control signal $\mathbf{u}(t)$ is

$$\mathbf{u}(t) = -K_y\mathbf{y}(t) = -K_yC\mathbf{x}(t) \tag{3.13}$$

where $\mathbf{y}(t) = C\mathbf{x}(t)$ is a system output that is directly measurable and K_y is constant. The block diagram of this feedback system is shown in Fig. 3.6.

The loop transfer function and the dynamic matrix of this feedback system are, respectively

$$L(s) = -K_yC(sI - A)^{-1}B \tag{3.14}$$

and $A - BK_yC$, which are very similar to that of the direct state feedback system. The only difference is that the constant gain on $\mathbf{x}(t)$ is K_yC instead of K, where C is a given system matrix. Hence static output feedback implements a constrained state feedback control with constraint

$$K = K_yC \tag{3.15}$$

In other words, K must be a linear combination of the rows of given matrix C, or $K' \in \mathbf{R}(C') \overset{\triangle}{=}$ range space of C' (see Subsection A.1.2). Because the dimension of this space is m, which is usually smaller than n, this constraint can be serious.

Example 3.3

In a second-order SISO system $(n = 2, p = m = 1)$, if C is either $[1 \;\; 0]$ or $[0 \;\; 1]$, then from (3.15) the state feedback control law $K = [k_1k_2]$

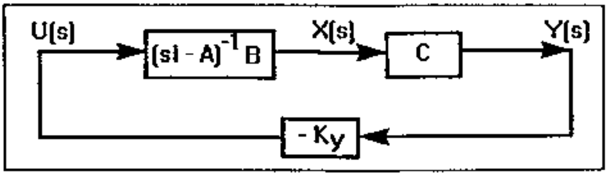


Figure 3.6 Static output feedback systems.

realized by the static output feedback must have either $k_2 = 0$ or $k_1 = 0$, respectively. This situation generally implies a reduction of the effectiveness of the control from dimension 2 to dimension 1.

If $m = n$ and if C is nonsingular, then (3.15) is no longer a constraint, and static output feedback becomes direct state feedback in the sense that $\mathbf{x}(t) = C^{-1}\mathbf{y}(t)$ and $K_y = KC^{-1}$. Therefore, direct state feedback control can be considered a special case of static output feedback control when C is nonsingular, and static output feedback control may be called "generalized state feedback control," as is done in this book.

The advantage of static output feedback control is its generality because $\mathbf{y}(t)$ is directly measurable. Besides, its corresponding loop transfer function is guaranteed to be $-K(sI - A)^{-1}B$ of (3.14) for whatever $K = K_yC$ of (3.15). This property is *not* shared by many other feedback systems (such as observer feedback systems). Finally, from the same argument of Theorem 3.1 and its proof, static output feedback control preserves controllability and observability properties of the original open-loop system.

The main drawback of static output feedback control is that it is usually too weak compared with direct state feedback control. This is because m is usually much smaller than n in practice, which makes the constraint (3.15) of static output feedback control too severe. For example, only when m is large enough (as compared to n) such that $m + p > n$, can arbitrary eigenvalues be assigned to the feedback system dynamic matrix $A - BK_yC$ [Kimura, 1975]. Example 3.3 is another such example.

As a result, the design of static output feedback control is far from satisfactory [Syrmos et al., 1994]. In this book, static output feedback control design algorithms for either pole assignment (Algorithm 8.1) or quadratic optimal control (Algorithm 9.2) are presented in Chaps 8 and 9, respectively.

3.2.3 Observer Feedback Systems—Loop Transfer Recovery

An observer feedback system does not require the direct observation of all system states, and implements a generalized state feedback control which is much stronger than the normal static output feedback control. Therefore observer feedback control structure overcomes the main drawbacks of both direct state feedback control structure and static output feedback control structure; it is the most commonly used control structure in state space control theory.

An observer is itself a linear time-invariant dynamic system, which has the general state space model

$$\dot{\mathbf{z}}(t) = F\mathbf{z}(t) + L\mathbf{y}(t) + TB\mathbf{u}(t) \tag{3.16a}$$

$$-K\mathbf{x}(t) = -K_z\mathbf{z}(t) - K_y\mathbf{y}(t) \tag{3.16b}$$

where $\mathbf{z}(t)$ is the state vector of observer system, B comes from the plant system state space model (A, B, C), and other observer parameters (F, T, L, K_z, K_y) are free to be designed.

This observer definition is more general than the existing ones. It is defined from the most basic and general observer function that it has $\mathbf{y}(t)$ and $\mathbf{u}(t)$ as its inputs and $K\mathbf{x}(t)$ as its output. The many distinct advantages of this general definition will be made obvious in the rest of this book.

Let us first analyze the conditions for an observer of (3.16) to generate a desired state feedback control signal $K\mathbf{x}(t)$.

Because both $\mathbf{x}(t)$ and $\mathbf{y}(t) = C\mathbf{x}(t)$ are time-varying signals, and because K and C are constants, it is obvious that to generate $K\mathbf{x}(t)$ in (3.16b), the observer state $\mathbf{z}(t)$ must converge to $T\mathbf{x}(t)$ for a constant T. This is the foremost important requirement of observer design.

Theorem 3.2

The necessary and sufficient condition for observer state $\mathbf{z}(t)$ to converge to $T\mathbf{x}(t)$ for a constant T, or for observer output to converge to $K\mathbf{x}(t)$ for a constant K, and for any $\mathbf{z}(0)$ and any $\mathbf{x}(0)$, is

$$T\ A - F\ T = L\ C \tag{3.17}$$

where all eigenvalues of matrix F must be stable.

Proof [Luenberger, 1971]

From (1.1a),

$$T\dot{\mathbf{x}}(t) = TA\mathbf{x}(t) + TB\mathbf{u}(t) \tag{3.18}$$

Subtracting (3.18) from (3.16a), we have

$$\dot{\mathbf{z}}(t) - T\dot{\mathbf{x}}(t) = F\mathbf{z}(t) + LC\mathbf{x}(t) - TA\mathbf{x}(t) \tag{3.19}$$

$$= F\mathbf{z}(t) - FT\mathbf{x}(t) + FT\mathbf{x}(t) + LC\mathbf{x}(t) - TA\mathbf{x}(t)$$

$$= F[\mathbf{z}(t) - T\mathbf{x}(t)] \tag{3.20}$$

if and only if (3.17) holds. Because the solution of (3.20) is

$$\mathbf{z}(t) - T\mathbf{x}(t) = \mathrm{e}^{Ft}[\mathbf{z}(0) - T\mathbf{x}(0)]$$

$\mathbf{z}(t)$ converges to $T\mathbf{x}(t)$ for any $\mathbf{z}(0)$ and any $\mathbf{x}(0)$ if and only if all eigenvalues of F are stable.

This proof also shows that it is necessary to let the observer gain to $\mathbf{u}(t)$ be defined as TB in (3.16a), in order for (3.19) to hold.

After $\mathbf{z}(t) = T\mathbf{x}(t)$ is satisfied, replacing this $\mathbf{z}(t)$ into the output part of observer (3.16b) yields

$$K = K_Z T + K_y C = [K_Z : K_y] \begin{bmatrix} T \\ C \end{bmatrix} \overset{\triangle}{=} \overline{K}\,\overline{C} \tag{3.21}$$

Therefore (3.17) (with stable F) and (3.21) together form the necessary and sufficient conditions for observer (3.16) to generate a desired state feedback $K\mathbf{x}(t)$.

The above introduction of (3.17) and (3.21) shows clearly that the two conditions have naturally and completely *separate* physical meanings. More explicitly, (3.17) determines the dynamic part of observer (F, T, L) *exclusively* and guarantees the observer state $\mathbf{z}(t) \Rightarrow T\mathbf{x}(t)$ *exclusively*, while (3.21) presumes that $\mathbf{z}(t) \Rightarrow T\mathbf{x}(t)$ is *already* satisfied and determines the output part of observer ($\overline{K}$ or $K = \overline{KC}$) *exclusively*. This basic design concept has not been applied before (except for a very narrow application of function observer design) and will be emphasized throughout the rest of this book.

There are many design algorithms that can satisfy (3.17) and (3.21) for arbitrary (stable) eigenvalues of F and arbitrary K (assuming observable systems). However, this book will present only one such algorithm (Algorithm 7.1), which has an *additional* feature of minimized observer order. This is because (3.17) and (3.21) have *not* addressed the critical robustness property of the observer feedback systems. This property will be analyzed in the rest of this chapter.

As stated in the beginning of this subsection, observer feedback systems have been the most commonly used control structure in state space control theory. Because an observer can generate the state feedback control signal $K\mathbf{x}(t) = \overline{KC}\mathbf{x}(t)$ [if (3.17) holds] and because the union of observer poles and eigenvalues of $A - BK = A - B\overline{KC}$ forms the entire set of observer feedback system poles [if (3.17) holds, see Theorem 4.1], it has been presumed that observer feedback systems have the same ideal robustness properties as those of the direct state feedback system corresponding $K = \overline{KC}$.

However, in practice since the 1960s, bad robustness properties of observer feedback systems have commonly been experienced, even though the observer implements a state feedback control whose corresponding direct state feedback system is supposed to have ideal robustness properties (see Subsection 3.2.1). Because robustness with respect to model uncertainty and control disturbance is critically important for most practical engineering systems (see Sec. 3.1), state space control theory has not found many successful practical applications since the 1960s.

At the same time, the application of the polynomial matrix and the rational polynomial matrix has extended classical control theory into MIMO systems [Rosenbrock, 1974; Wolovich, 1974; Kaileth, 1980; Chen, 1984; Vidyasagar, 1985]. Using the concept of loop transfer functions, classical control theory clarifies better than modern control theory the analysis of feedback system robustness properties (see Sec. 3.1). Furthermore, matrix singular values which can simply and accurately represent the matrix norm (such as loop transfer function matrix norm or loop gain) have become practically computable by computers (see Sec. A.3). As a result, classical control theory, especially in terms of its robust design, has witnessed significant development during the past two decades [Doyle et al., 1992]. For example, the H_∞ problem, which may be briefly formulated as

$$\min\{\max_{\omega}\{\|[I - L(j\omega)]^{-1}\|_\infty\}\} \quad \text{(see Definition 2.4)}$$

has received much attention [Zames, 1981; Francis, 1987; Doyle et al., 1989; Kwakernaak, 1993; Zhou et al., 1995].

Until the end of 1970s, there was a consensus of understanding on the cause of the problems of bad robustness observer feedback systems. This understanding was based solely on the perspective of loop transfer functions [Doyle, 1978]. We will describe this understanding in the following.

The feedback system of the general observer (3.16) can be depicted as in Fig. 3.7, which shows that an observer can be considered a feedback compensator $H(s)$ with input $\mathbf{y}(t)$ and output $\mathbf{u}(t)$, where

$$\begin{aligned} U(s) &= -H(s)Y(s) \\ &= -[I + K_Z(sI - F)^{-1}TB]^{-1}[K_y + K_Z(sI - F)^{-1}L]Y(s) \end{aligned} \qquad (3.22)$$

It should be noticed from (3.16) that the transfer function from signal $\mathbf{y}(t)$ to

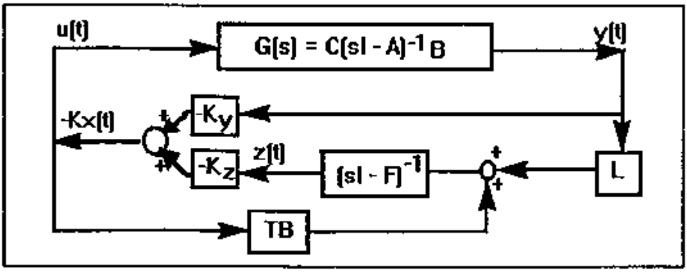


Figure 3.7 Block diagram of general observer feedback systems.

$-K\mathbf{x}(t)$ is

$$H_{Kx}(s) = -[K_y + K_Z(sI - F)^{-1}L] \tag{3.23}$$

which is *different* from $-H(s)$ of (3.22). The difference is caused *solely* by the feedback of signal $\mathbf{u}(t)$ to the observer. If this feedback, which is defined by its path gain TB and its loop gain $K_Z(sI - F)^{-1}TB$, equals zero, then $-H(s) = H_{Kx}(s)$.

Theorem 3.3

The loop transfer function at the break point $-K\mathbf{x}(t)$ of Fig. 3.7, $L_{Kx}(s)$, equals that of the corresponding direct state feedback system (3.10), or

$$L_{Kx}(s) = -K(sI - A)^{-1}B \tag{3.24}$$

Proof [Tsui, 1988a]

From Fig. 3.7,

$$L_{Kx}(s) = H_{Kx}(s)G(s) - K_Z(sI - F)^{-1}TB \tag{3.25a}$$

by (3.23)

$$= -K_yG(s) - K_Z(sI - F)^{-1}[LG(s) + TB] \tag{3.25b}$$

by (1.9)

$$= -[K_y C + K_Z(sI - F)^{-1}(LC + sT - TA)](sI - A)^{-1}B$$

by (3.17)

$$= -[K_y C + K_Z(sI - F)^{-1}(sI - F)T](sI - A)^{-1}B$$

by (3.21)

$$= -K(sI - A)^{-1}B$$

Figure 3.7 also shows that $-K\mathbf{x}(t)$ is only an internal signal of compensator $H(s)$, while $\mathbf{u}(t)$ is the *real* analog control signal that is attributed to the plant system $G(s)$ and which is where the disturbance is introduced (see Subsection 3.1.2). Therefore, the loop transfer function $L(s)$, which really determines the sensitivity properties of the observer feedback system, should be the one at break point $\mathbf{u}(t)$ [Doyle, 1978]. From Fig. 3.7,

$$L(s) = -H(s)G(s)$$

by (3.22)

$$= -[I + K_Z(sI - F)^{-1}TB]^{-1}[K_y + K_Z(sI - F)^{-1}L]G(s) \tag{3.26}$$

Because $L(s) \neq L_{Kx}(s) = -K(sI - A)^{-1}B$ and because loop transfer function plays a critical role in the feedback system sensitivity, the observer feedback system has different robustness properties from that of the corresponding direct state feedback system [Doyle, 1978].

Example 3.4

In order to further understand the difference between the two loop transfer functions of (3.25) and (3.26), we will analyze two more system diagrams of observer feedback systems. The first diagram (Fig. 3.8) is called a "signal flow diagram."

For simplicity of presentation, we may assume the path branch with gain $K_y = 0$ and ignore this path branch. Then Fig. 3.8 shows that at node $\mathbf{u}(t)$ there is only *one* loop path. The loop with gain $-K_Z(sI - F)^{-1}TB$ is attached to this single loop path. In contrast, at node $-K\mathbf{x}(t)$, there are *two*

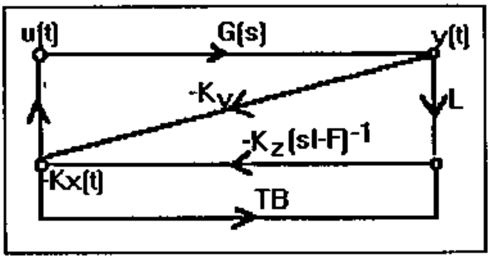


Figure 3.8 Signal flow diagram of observer feedback systems.

loop paths. The loop with gain $-K_Z(sI-F)^{-1}TB$ is an *independent* loop path between the two.

The second block diagram (Fig. 3.9) is also common in literature. In this equivalent block diagram of observer feedback systems,

$$H_y(s) = -[K_y + K_Z(sI-F)^{-1}L] = H_{Kx}(s) \text{ of } (3.23)$$

and

$$H_u(s) = -K_Z(sI-F)^{-1}TB \tag{3.27}$$

We should reach the same conclusion from Figs 3.8 and 3.9 on the loop transfer functions at nodes $\mathbf{u}(t)$ and $-K\mathbf{x}(t)$. They are

$$\begin{aligned} L(s) &= [I - H_u(s)]^{-1}H_y(s)G(s) \\ &= -[I + K_Z(sI-F)^{-1}TB]^{-1}[K_y + K_Z(sI-F)^{-1}L]G(s) \end{aligned} \tag{3.26}$$

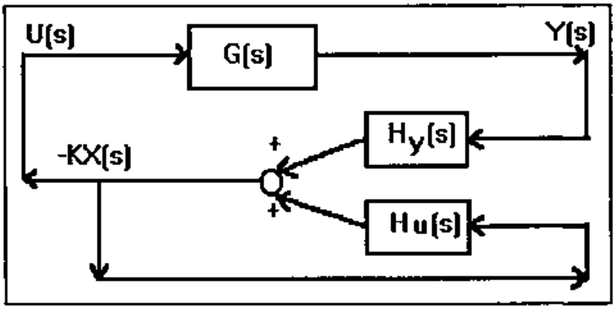


Figure 3.9 An equivalent block diagram of observer feedback systems.

and

$$L_{Kx}(s) = H_y(s)G(s) + H_u(s) \tag{3.25a}$$
$$= -K_yG(s) - K_Z(sI - F)^{-1}[LG(s) + TB] \tag{3.25b}$$

respectively.

Theorem 3.4

The necessary and sufficient condition for observer feedback system loop transfer function $L(s)$ to be the same as that of the corresponding direct state feedback system $L_{Kx}(s)$ is

$$H_u(s) = -K_Z(sI - F)^{-1}TB = 0 \quad \forall s \tag{3.28a}$$

For freely designed state feedback gain K [or K_Z of (3.21)], (3.28a) becomes

$$H_u(s) = -K_Z(sI - F)^{-1}TB = 0 \quad \forall s \text{ and } K_Z \tag{3.28b}$$

The necessary and sufficient condition for (3.28b) is

$$TB = 0 \tag{3.29}$$

Proof

Figure 3.7, Example 3.4, and the comparison between (3.25) and (3.26) all indicate clearly that the difference between $L_{Kx}(s)$ and $L(s)$ is caused *solely* by the feedback loop [with gain $H_u(s)$]. Therefore, the necessary and sufficient condition for $L(s) = L_{Kx}(s)$ is $H_u(s) = 0$ [or (3.28a)].

Because $(sI - F)^{-1}$ should be nonsingular $\forall s$ and K_Z should be freely designed, $TB = 0$ is obviously the necessary and sufficient condition for (3.28b).

Comparing Figs 3.1 and 3.9, this theorem indicates that only the system structure of Fig. 3.1, which does not have the feedback from input $\mathbf{u}(t)$ and which is therefore called the "output feedback compensator" (see Sec. 4.4), can guarantee the same loop transfer function of the direct state feedback system.

In papers [Doyle and Stein, 1979, 1981] subsequent to Doyle [1978], the authors imposed the problem of making $L(s) = L_{Kx}(s)$, which is called "loop transfer recovery" (LTR). This problem is clearly an additional requirement of observer design—the observer is required not only to realize a desired state feedback control signal, but also to have $L(s) = L_{Kx}(s)$. Mathematically speaking, from Theorems 3.2–3.4, the observer is required to satisfy not only (3.17) and (3.21), but also (3.29) (if the state feedback control is freely designed).

The LTR requirement can eliminate the basic cause of sensitivity problems of observer feedback systems and is therefore of great practical importance to the entire state space control theory.

Unfortunately, for *almost all* given plant systems, it is impossible to have an observer that can generate the arbitrarily given state feedback signal $K\mathbf{x}(t)$ while satisfying (3.28a) or (3.29) (see Sec. 4.3). For this reason, this book proposes a new and systematic design approach which is general for *all* plant systems. This new approach can design an observer that generates a constrained state feedback signal $K\mathbf{x}(t) = \overline{K}\overline{C}\mathbf{x}(t)$ ($\overline{K}$ is completely freely designed) that satisfies (3.29) exactly for most plant systems (see Sec. 4.4) and that satisfies (3.29) in a least-square sense for all other plant systems.

Although a state observer that can generate the *arbitrarily* given $K\mathbf{x}(t)$ *cannot* satisfy (3.28a) or (3.29) for almost all plant systems, such an observer is required by all other LTR design methods. At the other extreme, the study of *every possible* $K\mathbf{x}(t)$ that can be generated by an observer [satisfying (3.28a), (3.17), and (3.21)] and that can stabilize the matrix $A - BK$ has been reported [Saberi, 1991]. Obviously, the K (or K_Z) that is constrained on (3.28a), (3.17), (3.21), and stable $A - BK$ is only a theoretical formulation (or reformulation). The K under this formulation (K_Z is *not* free) cannot be systematically designed, in contrast to the K that is constrained only on $K = \overline{K}\overline{C}$ ($\overline{K}$ or K_Z are free) of our design (see Subsection 3.2.2, the paragraph at the end of Sec. 4.2, and the corresponding technical argument in Sec. 4.4).

SUMMARY

Loop transfer function is a critical factor which determines the feedback system sensitivity, and requirement (3.29) is necessary and sufficient to preserve observer feedback system loop transfer function from that of its corresponding direct state feedback system, for either arbitrarily given or freely [but with constraint (3.21)] designed state feedback.

State feedback control, either unconstrained or constrained by (3.21), is the general and the basic form of control of state space control theory, and is by far the best among all existing basic forms of control.

The observer (3.16) is the main feedback compensator structure of state space control theory, but it is required to satisfy (3.17) and nonsingular $\overline{C}$ [or (3.21) for all K] in most of the literature. Observers with additional requirement (3.28a) or (3.29) in the existing literature are very severely limited. This book introduces a fundamentally new observer design approach which can satisfy (3.17), (3.21), and (3.29) much more generally.

4

A New Feedback Control Design Approach

Chapter 3 analyzed the observer design requirements, which can be outlined as follows.

To guarantee observer state $\mathbf{z}(t) \Rightarrow T\mathbf{x}(t)$, we require

$$TA - FT = LC \quad (F \text{ is stable}) \tag{4.1}$$

To guarantee the generation of signal $K\mathbf{x}(t)$, we require [assuming $\mathbf{z}(t) \Rightarrow T\mathbf{x}(t)$]

$$K = [K_Z : K_y]\begin{bmatrix} T \\ C \end{bmatrix} \overset{\Delta}{=} \overline{K}\,\overline{C} \tag{4.2}$$

Finally, to realize the same robustness properties of the state feedback control which can be designed systematically, we require (Theorem 3.3 and 3.4)

$$TB = 0 \tag{4.3}$$

The real challenge is *how* to generally and systematically satisfy these three requirements. A fundamentally new design approach of satisfying these three requirements is proposed in this chapter, which is divided into four sections.

Section 4.1 points out a basic and general observer design concept that (4.1) should be satisfied *separately* and before satisfying (4.2) for arbitrary K (or nonsingular $\overline{C}$). In most existing observer design and in all existing LTR observer design, only state observers are designed which imply the simultaneous satisfaction of (4.1) *and* nonsingular $\overline{C}$. This basic concept implies the generation of $K\mathbf{x}(t)$ directly from $\mathbf{z}(t)$ [$\Rightarrow T\mathbf{x}(t)$] and $\mathbf{y}(t)$ [$= C\mathbf{x}(t)$] instead of from the explicit $\mathbf{x}(t) = \overline{C}^{-1}[\mathbf{z}(t)' : \mathbf{y}(t)']'$. This concept is used throughout the rest of this book.

Section 4.2 analyzes the poles (or performance) of the observer feedback system. It proves a revised version of the "separation property" that (4.1) *alone* (not nonsingular $\overline{C}$) is the sufficient condition for observer feedback system poles being composed of the eigenvalues of F and $A - B\overline{KC}$.

Section 4.3 reviews the current state of existing results of LTR. It points out that while state observers can be designed generally, the LTR state observers are very severely limited.

Section 4.4 summarizes the conclusions of the first three sections and proposes a fundamentally new design approach which satisfies (4.1) and (4.3) *first* (not nonsingular $\overline{C}$), and which satisfies (4.1)–(4.3) much more generally, simply, and systematically. The only tradeoff of this new design approach is that its state feedback $K\mathbf{x}(t)$ can be constrained on (4.2) because $\overline{C}$ may not always be nonsingular. This tradeoff is obviously necessary and worthwhile in light of the severe drawbacks of the results of Sec. 4.3.

4.1 BASIC DESIGN CONCEPT OF OBSERVERS—DIRECT GENERATION OF STATE FEEDBACK CONTROL SIGNAL WITHOUT EXPLICIT SYSTEM STATES

We will use the design examples of three basic observers to explain that satisfying (4.1) first and then (4.2) keeps with the basic physical meanings of these two requirements. Because (4.1) *alone* implies that $\mathbf{z}(t) \Rightarrow T\mathbf{x}(t)$, this separation also implies the direct generation of $K\mathbf{x}(t)$ in (4.2) from $\mathbf{z}(t)$ [$\Rightarrow T\mathbf{x}(t)$] and $\mathbf{y}(t)$ [$= C\mathbf{x}(\mathrm{t})$].

Example 4.1 Full-Order Identity State Observers

Let $T = I$ and $F = A - LC$ in the observer part (3.16a). Then (4.1) is obviously satisfied and (3.16a) becomes

$$\dot{\mathbf{z}}(t) = (A - LC)\mathbf{z}(t) + L\mathbf{y}(t) + B\mathbf{u}(t) \tag{4.4}$$

$$= A\mathbf{z}(t) + B\mathbf{u}(t) + L[\mathbf{y}(t) - C\mathbf{z}(t)] \tag{4.5}$$

Subtracting (1.1a) from (4.4),

$$\dot{\mathbf{z}}(t) - \dot{\mathbf{x}}(t) = (A - LC)[\mathbf{z}(t) - \mathbf{x}(t)] = F[\mathbf{z}(t) - \mathbf{x}(t)]$$

Therefore, $\mathbf{z}(t) \Rightarrow \mathbf{x}(t)$ if F is stable. Thus we have repeated the proof of Theorem 3.2.

In the above argument, (4.2) is not involved and (4.1) *alone* completely determines the observer (4.4)–(4.5), which generates $\mathbf{x}(t)$. Only *after* $\mathbf{z}(t) \Rightarrow T\mathbf{x}(t)$ is generated do we multiply $\mathbf{z}(t)$ by K [or let $[K_Z : K_y] \triangleq \overline{K} = [K : 0]$ in (3.16b) and (4.2)] in order to generate the desired state feedback $K\mathbf{x}(t)$.

Because parameter T has n rows, this observer has n states, and it is therefore called "full order." In addition, if T is not an identity matrix but is nonsingular, then $\mathbf{x}(t)$ does not equal $\mathbf{z}(t)$ but equals $T^{-1}\mathbf{z}(t)$. We define any observer of (3.16) that estimates $\mathbf{x}(t)$ as a "state observer." We therefore call the observer with $T = \mathrm{I}$ an "identity observer" and consider it a special case of full-order state observers.

It is obvious that TB cannot be 0 for a nonsingular T. Therefore, a full-order state observer cannot satisfy LTR (4.3).

The observer structure of (4.5) is also the structure of Kalman filters [Anderson, 1979; Balakrishnan, 1984], where L is the filter gain. The Kalman filter can therefore be considered a special case of full-order identity state observer.

The full-order identity state observer feedback system has the block diagram shown in Fig. 4.1.

Example 4.2 Reduced-Order State Observers

Contrary to full-order state observers, the order of a reduced-order state observer equals $n - m$ and $\mathbf{y}(t)$ is used in (3.16b) ($K_y \neq 0$). Thus the parameter T of this observer has only $n - m$ rows and cannot be square.

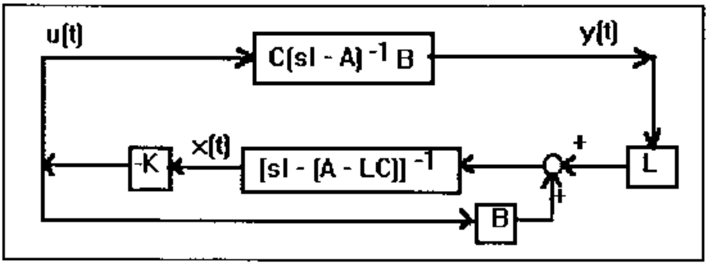


Figure 4.1 Full-order identity state observer feedback system.

As in the design of Example 4.1, (4.1), and $\mathbf{z}(t) \Rightarrow T\mathbf{x}(t)$ must be satisfied first (see Theorem 3.2). Only then, to generate $\mathbf{x}(t) = I\mathbf{x}(t)$ in (3.16b) or to satisfy

$$I\mathbf{x}(t) = [K_Z : K_y][\mathbf{z}(t)' : \mathbf{y}(t)']' = \overline{K}[T' : C']'\mathbf{x}(t) \tag{4.6}$$

matrix $\overline{C} \triangleq [T' : C']'$ must be nonsingular and $\overline{K} = \overline{C}^{-1}$. Therefore, in this design, the requirement (4.2) $(I = \overline{KC})$ again comes *after* (4.1) and is separated from (4.1).

The reason that this observer can have order lower than n comes from the utilization of the information of $\mathbf{y}(t) = C\mathbf{x}(t)$ in (3.16b), (4.2), and (4.6). Mathematically speaking, with the addition of m rows of matrix C in matrix $\overline{C}$, the number of rows of T can be reduced from n to $n - m$ in order to make matrix $\overline{C}$ square and nonsingular. The reduced-order state observer feedback system can be depicted as shown in Fig. 4.2.

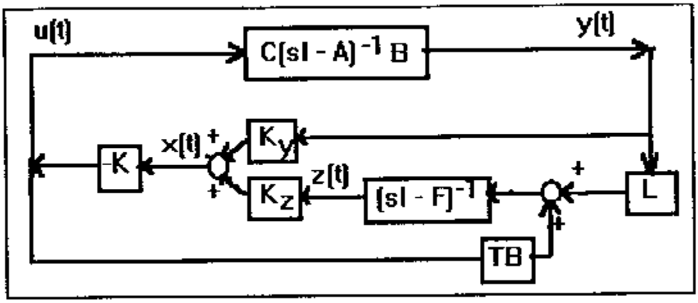


Figure 4.2 Reduced-order state observer feedback system.

In the formulation (3.16) of observers, the signal $K\mathbf{x}(t)$ is estimated, with K being a general matrix. Therefore the state observer that estimates $\mathbf{x}(t) = I\mathbf{x}(t)$ is a special case of the observers of (3.16) in the sense that the general matrix K of the latter becomes a special identity matrix I of the former.

Examples 4.1 and 4.2 also show that because matrix I has rank n, matrix $\overline{C}$ (which equals T in full-order state observers and $[T' : C']'$ in reduced-order state observers) must be a nonsingular square matrix. Therefore the number of rows of matrix T or the order of these two types of state observers must be n and $n - m$, respectively.

However, after $\mathbf{z}(t) \Rightarrow T\mathbf{x}(t)$ is satisfied by (4.1), the desired state feedback $K\mathbf{x}(t)$ can be generated *directly* from $\mathbf{z}(t) = T\mathbf{x}(t)$ and $\mathbf{y}(t) = C\mathbf{x}(t)$ without explicit information on $\mathbf{x}(t)$. From a linear algebraic point of view, Eq. (4.2) ($K = \overline{K}\overline{C}$) can be solved without the computation of $\overline{C}^{-1}$. More important, for $p \ll n$, which is generally true, a very wide range of desirable K can be satisfied by (4.2) *without* a nonsingular $\overline{C}$, as long as $K' \in \mathbf{R}(\overline{C}')$, even though a nonsingular $\overline{C}$ [or the estimation of $\mathbf{x}(t)$] is still required $\forall K$. This basic understanding offers the following two possible significant improvements of observer design.

The first is observer order reduction, because the observer order equals the number of rows of T in matrix $\overline{C}$. We will call the observer that estimates the desired state feedback $K\mathbf{x}(t)$ and with minimal order the "minimal order observer." The design results of this observer will be reviewed in Example 4.3, and the first systematic and general design algorithm of this observer [Tsui, 1985] is presented in Chap. 7.

The second, and even more significant, improvement is that not requiring $\overline{C}$ to be nonsingular implies that the entire remaining observer design freedom after (4.1) can be fully used to satisfy (4.3), or to realize the robustness properties of the state feedback control that the observer is trying to realize. This is the key concept behind the new design approach, which is formally proposed in Sec. 4.4 [Tsui, 1987b]. The exact and analytical solution of (4.1) and (4.3) [Tsui, 1992, 1993b] will be described in Chaps 5 and 6.

It should be emphasized that the *single* purpose of an observer in almost all control system applications is to realize a state feedback control $K\mathbf{x}(t)$ but *not* to estimate explicit plant system state $\mathbf{x}(t)$. When $\mathbf{x}(t)$ is estimated by a state observer, it is multiplied immediately by K (see Figs 4.1 and 4.2).

Definition 4.1

The observer (3.16) that generates the desired $K\mathbf{x}(t)$ directly [without generating explicitly $\mathbf{x}(t)$] is called the "function observer." Obviously, only function observers can have minimal orders that are lower than $n - m$.

Example 4.3 Overview of Minimal Order Function Observer Design

Order reduction has been an important problem in control systems theory [Kung, 1981] and high observer order has been a major cause of impracticality of state space control theory.

Based on the analysis of this section, the *only* difference between the minimal order observer and the other observers is at Eq. (4.2):

$$K = [K_Z : K_y] \begin{bmatrix} T \\ C \end{bmatrix} = \overline{KC}$$

in which the least possible number of rows of matrix T is sought in design computation. To do this computation generally and systematically, every row of matrix T in (4.2) must be completely decoupled from each other and must correspond to only one eigenvalue of matrix F (or only one observer pole). In addition, the complete freedom of T must also be fully used in this design computation. Because T must satisfy (4.1) first, the freedom of T to be used in (4.2) can be considered the remaining freedom of (4.1).

Although there have been many attempts at minimal order observer design [Gopinath, 1971; Fortmann and Williamson, 1972; Gupta et al., 1981; Van Loan, 1984; Fowell et al., 1986], which have been clearly documented in O'Reilly [1983], the above solution matrix T of (4.1) has not been derived [Tsui, 1993a]. As a result, it has been necessary to solve (4.1) and (4.2) *together* and it has not been possible to solve (4.2) *separately* and therefore systematically [Tsui, 1993a]. As a result, the general and systematic minimal order observer design problem has been considered a difficult and unsolved problem [Kaileth, 1980, p. 527; Chen, 1984, p. 371].

The above solution matrix T has been derived by Tsui [1985]. Thus the minimal order observer design has been really and uniquely simplified to the solving of only (4.2), which is only a set of linear equations. A general and systematic algorithm of minimal order observer design [or the solving of (4.2) for minimal number of rows of T] is proposed in Tsui [1985] and is introduced as Algorithm 7.1 in Chap. 7 of this book.

Minimal order function observer is the *only* existing observer that generates the desired $K\mathbf{x}(t)$ signal directly, without the explicit $\mathbf{x}(t)$ [or, satisfying (4.1), without a nonsingular $\overline{C}$], and it is the only application of this basic design concept. This example shows that this observer can be generally and systematically designed, based *only* on a desirable solution of (4.1) [Tsui, 1985].

The above three examples of existing basic observer design demonstrate that satisfying (4.1) first without a nonsingular $\overline{C}$ in (4.2) [or generating $K\mathbf{x}(t)$ directly, without generating $\mathbf{x}(t) = \overline{C}^{-1}[\mathbf{z}(t)' : \mathbf{y}(t)']'$] fits the original physical meanings of these two conditions and is in keeping with the existing basic observer design procedures.

This design concept enables the elimination of the difficult and unnecessary requirement of complete state estimation or the requirement that $\overline{C}$ be nonsingular, and thus enables the possibility of significant improvements on observer design (one of which is observer order reduction).

Example 4.3 also demonstrates that this basic concept has been obscured by the fact that almost all observer results involve state observers *only*, and by the previous unsuccessful attempts at the general and systematic design of minimal order function observers.

4.2 PERFORMANCE OF OBSERVER FEEDBACK SYSTEMS—SEPARATION PROPERTY

In the previous section, we discussed the design concept of satisfying (4.1) separately without satisfying a nonsingular matrix $\overline{C}\triangleq[T' : C']'$.

In this section, we will prove that (4.1) *alone* (*not* with a nonsingular $\overline{C}$) guarantees that the observer feedback system poles be the eigenvalues of F and $A - B\overline{KC}$. Thus (4.1) alone also guarantees explicitly and to a certain degree the observer feedback system's performance (see Sec. 2.1). This is an essential validation of the new design approach of this book, which seeks the satisfaction of (4.1) and (4.3) first, without a nonsingular matrix $\overline{C}$.

Theorem 4.1 (Separation property)

If (4.1) is satisfied, then the poles of the feedback system that is formed by the plant system (1.1) and the general observer (3.16) are composed of the eigenvalues of matrices F of (3.16) and $A - BK$ of (1.1) and (4.2).

Proof [Tsui, 1993b]

Substituting (3.16b) into the plant system input $\mathbf{u}(t)$ and then substituting this $\mathbf{u}(t)$ and $\mathbf{y}(t) = C\mathbf{x}(t)$ into the dynamic part of plant system (1.1a) and observer (3.16a), the dynamic equation of the observer feedback system is

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{z}}(t) \end{bmatrix} = \begin{bmatrix} A - BK_yC & -BK_Z \\ LC - TBK_yC & F - TBK_Z \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{z}(t) \end{bmatrix} \overset{\triangle}{=} A_c \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{z}(t) \end{bmatrix} \tag{4.7}$$

Multiplying

$$Q^{-1} = \begin{bmatrix} I & 0 \\ -T & I \end{bmatrix}$$

and

$$Q = \begin{bmatrix} I & 0 \\ T & I \end{bmatrix}$$

on the left and right side of A_c, respectively, we have

$$\overline{A}_c = Q^{-1}A_cQ = \begin{bmatrix} A - BK_yC - BK_ZT & -BK_Z \\ -TA + FT + LC & F \end{bmatrix}$$

if (4.1)

$$= \begin{bmatrix} A - B\overline{KC} & -BK_Z \\ 0 & F \end{bmatrix} \tag{4.8}$$

The eigenvalues of $A - B\overline{KC}$ and of F will constitute all eigenvalues of $\overline{A}_c$ of (4.8), which has the same eigenvalues of A_c.

In the normal and existing state space design practice, either the eigenvalues of F and $A - B\overline{KC}$ are assigned without considering the overall feedback system poles, or the overall system is designed without considering the poles of its feedback compensator. The separation property guarantees the overall observer feedback system poles once the eigenvalues of F and $A - B\overline{KC}$ are assigned. Therefore from Sec. 2.1, it guarantees explicitly the overall observer feedback system performance to the degree of those assigned poles. It also guarantees the poles and the stability of observer (3.16) from the stability of the overall observer feedback system, in case the design is carried out from the perspective of the overall feedback systems.

The separation property is thus extremely important and has appeared in almost all state space control literature, such as O'Reilly [1983].

However, the general observer (3.16) formulation (with generalized dynamic part and generalized state feedback output) has not really been extended to the existing literature. More important, the property that (4.1) *alone* (not a nonsingular $\overline{C}$) is the sufficient condition of Theorem 4.1 has not really been clarified in the existing literature either [Tsui, 1993b].

Because in the original version of Theorem 4.1 the parameter $\overline{KC}$ is replaced by an arbitrary K, it has been customary to assign the eigenvalues of F in (4.1) and the eigenvalues of $A - BK$ (3.11) *completely* separately. Hence the name "separation property."

However, as will be described in the next section, for most plant systems, an arbitrarily designed state feedback $K\mathbf{x}(t)$ cannot be implemented by an observer with a nonsingular $\overline{C}$ *and* with exact LTR [or (4.3)]. The new design approach of this book fundamentally changes this traditional design practice by designing the state feedback gain K *based* on $K = \overline{K}[T' : C']'$, where observer parameter T satisfies (4.1) *and* (4.3). This new design approach is validated partly by the above revised separation property, which shows that (4.1) alone is the sufficient condition of this property, while the addition of constraint $K = \overline{K}[T' : C']'$ generalizes this property from K to $K = \overline{K}[T' : C']'$.

Finally, for the sake of theoretical integrity, we shall point out that the condition (4.1) is not a necessary condition of Theorem 4.1 for every possible combination of ($\overline{C}$, $\overline{K}$, F, T, L). This point can be simply proved by the following special example.

Example 4.4

Let a matrix $\overline{A}_c$ and its characteristic polynomial be

$$|sI - \overline{A}_c| = \begin{vmatrix} sI - (A - B\overline{KC}) & BK_z \\ TA - FT - LC & sI - F \end{vmatrix}$$

$$= \left|\begin{array}{cc:c} s-a & -b & 1 \\ -b & s-a & -1 \\ \hdashline c & c & s-f \end{array}\right|$$

where parameters (a, b, c, f) are scalars. Then

$$\begin{aligned} |sI - \overline{A}_c| &= (s-f)|sI - (A - B\overline{KC})| + \begin{vmatrix} s-a & -b \\ c & c \end{vmatrix} + \begin{vmatrix} -b & s-a \\ c & c \end{vmatrix} \\ &= (s-f)|sI - (A - B\overline{KC})| \end{aligned} \tag{4.9}$$

The equality of (4.9) (or separation property) holds even if parameter $c \neq 0$, or even if (4.1) is not satisfied.

In any practical design, the parameters of $(\overline{C}, \overline{K}, F, T, L)$ have to be designed to satisfy (4.1), (4.3), and a satisfactory $A - B\overline{K}\overline{C}$, but not to fit the special case of Example 4.4. Thus this argument on the necessity of (4.1) to Theorem 4.1 is totally meaningless in practice.

This situation is very similar to the argument of Saberi et al. [1991] that (4.3) is not necessary for exact LTR and for every possible combination of parameters $(\overline{K}, F, T)$. This is because the parameters of $(\overline{K}, F, T)$ have to be designed to satisfy (4.1) and a satisfactory $A - B\overline{K}\overline{C}$, but not to fit those special cases that satisfy (3.28a) but not (4.3).

4.3 THE CURRENT STATE OF LTR OBSERVER DESIGN

As discussed in Sec. 4.1, besides observer order reduction, a much more important observer design improvement is the significantly more general and systematic robustness preservation (or LTR) of observer feedback systems.

From Theorems 3.3 and 3.4, the requirement of LTR [or (4.3)] can eliminate the basic cause of sensitivity problems of observer feedback systems and is therefore of great practical importance. As a result, this problem has received much attention since its proposition [Sogaard-Andersen, 1986; Stein and Athans, 1987; Dorato, 1987; Tsui, 1987b; Moore and Tay, 1989; Saberi and Sannuti, 1990; Liu and Anderson, 1990; Niemann et al., 1991; Saeki, 1992; Tsui, 1992, 1993b; Saberi et al., 1993; Tsui, 1996a, b; Tsui, 1998b].

However, the mere proposition and formulation of a problem does not imply that the problem is solved, and experience shows that the latter can be much more difficult than the former. Even the derivation of some initial solutions of a problem does not imply the problem is solved satisfactorily, and experience also shows that the latter can be much more difficult than the former. Furthermore, only the theoretical problem with a really satisfactory solution can have real practical value.

This section shows that all other existing LTR observers are state observers. While a state observer without the LTR requirement (4.3) can be generally designed, the state observer with (4.3), which is called the "exact LTR state observer," is very severely limited.

It has been proved that to have an exact LTR state observer or to satisfy (4.1) and (4.3) with arbitrarily given K [or to satisfy (4.1) and (4.3) with a nonsingular $\overline{C}$], the plant system must satisfy either one of the following two restrictions [Kudva et al., 1980]. These two restrictions are originally derived for the existence of the "unknown input observers." An unknown input observer is a state observer with zero gain to the plant system's unknown input [Wang et al., 1975]. Hence it is equivalent to an exact LTR state observer, if we consider the plant system gain to the unknown input signal as matrix B.

The first restriction is that the plant system must have $n - m$ stable transmission zeros. This is extremely restrictive because most systems with $m \neq p$ do not have that many transmission zeros in the first place (see Example 1.8 and Davison and Wang, 1974).

The second is a set of three restrictions: (1) minimum-phase (all transmission zeros are stable), (2) rank $(CB) = p$, and (3) $m \geqslant p$. This again is extremely restrictive because it is *very hard* to require *all* existing transmission zeros be stable (see Exercises 4.2 and 4.6), and rank $(CB) = p$ is also not satisfied by many practical systems such as airborne systems.

The above two restrictions can be related by the following property of transmission zeros [Davison and Wang, 1974]; namely, that almost all systems with $m = p$ have $n - m$ transmission zeros, and that all systems with $m = p$ and with rank $(CB) = p$ have $n - m$ transmission zeros. Therefore the second restriction is a little more general than the first restriction because it admits some additional plant systems with $m > p$.

For plant systems not satisfying the above two restrictions, if they are minimum-phase, then there is an asymptotic LTR state observer for these systems, while there exist no other unknown input observer results for these systems because the above two restrictions are necessary conditions of unknown input observers.

Asymptotic LTR state observers have been widely documented [Doyle and Stein, 1979; Stein and Athans, 1987; Dorato, 1987; Moore and Tay, 1989; Saberi and Sannuti, 1990; Niemann et al., 1991; Saberi et al., 1993] and have been considered the main result of LTR because minimum-phase restriction is less strict than the above two restrictions for exact LTR state observers.

There are mainly two design approaches for asymptotic LTR state observers.

The first is valid for minimal-phase systems only, and is to asymptotically increase the plant system input noise level when designing the Kalman filter [Doyle and Stein, 1979] or to asymptotically increase the time scale of state observer poles [Saberi and Sannuti, 1990]. Unfortunately, this approach inevitably and asymptotically increases the observer gain L. As discussed in Sec. 3.1 and Shaked and Soroka, (1985); Tahk and Speyer, (1987); and Fu, (1990), the large gain L is even more harmful to system sensitivity properties than not having LTR at all.

The second approach is to compute a loop transfer function $L(s)$ whose difference to the target loop transfer function $L_{Kx}(s)$ has an H_∞ norm bound over frequency [Moore and Tay, 1989]. Unfortunately, this bound is itself generally unpredictable. For example, in the actual design it is ever increased until a numerical solution of a bounded value Riccati equation exists—it does not converge to a lower level at all [Weng and Shi, 1998]. Even more critically, at the frequency ω of this bound, no consideration is made and no bound exists for the phase angle of $L(j\omega) - L_{Kx}(j\omega)$.

To summarize, the existing exact LTR state observer is too restrictive, while the existing asymptotic LTR state observers are far from satisfactory.

The main reason for these unsatisfactory LTR results is the requirement of state estimation or the requirement of implementing *arbitrarily* given state feedback control. Mathematically speaking, $\overline{C}$ nonsingular is a difficult yet unnecessary additional requirement [in addition to necessary conditions (4.1) and (4.3)] to satisfy.

For example, most of the existing LTR results involve Kalman filters. The Kalman filter design freedom is used *almost completely* for minimum variance state estimation [Anderson and Moore, 1979; Balakrishnan, 1984] and *not* for LTR. The only remaining design freedom of Kalman filters for LTR is a scalar plant system input noise level q [Doyle and Stein, 1979]. As q is increased asymptotically for achieving LTR, the Kalman filter poles must approach each of the plant system transmission zeros and negative infinity at Butterworth pattern [Anderson and Moore, 1979]. This is the reason that the Kalman filter-based exact LTR observer requires $n - m$ stable plant system transmission zeros [Stein and Athans, 1987; Friedland, 1989], and is the reason that the asymptotic LTR state observer requires that the plant system be minimum-phase [Doyle and Stein, 1979, 1981].

Example 4.5 The Unsatisfactory State of the Existing Asymptotic LTR Result

Let the given plant system be

$$(A, B, C) = \left(\begin{bmatrix} 0 & -3 \\ 1 & -4 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \end{bmatrix}, [0 \quad 1] \right)$$

and

$$G(s) = \frac{s+2}{(s+1)(s+3)}$$

which has $n - m = 2 - 1$ stable transmission zero -2.

Let us design an observer which can implement a quadratic optimal state feedback

$$K = [30 \ -50]$$

whose corresponding loop transfer function (3.10) is

$$L_{Kx}(s) = -K(sI - A)^{-1}B = \frac{-(10s + 50)}{(s + 1)(s + 3)}$$

This example was raised by Doyle and Stein [1979], which provided two types of observer results:

1. A full-order identity state observer with poles $-7 \pm j2$:

   $$(F = A - LC, T, L, K_Z, K_y) = \left(\begin{bmatrix} 0 & -53 \\ 1 & -14 \end{bmatrix}, I, \begin{bmatrix} 50 \\ 10 \end{bmatrix}, K, 0 \right)$$

   whose corresponding loop transfer function is computed as

   $$\begin{aligned} L(s) &= -[1 + K(sI - F)^{-1}B]^{-1}[0 + K(sI - F)^{-1}L]G(s) \\ &= \frac{-100(10s + 26)}{s^2 + 24s - 797} \times \frac{s + 2}{(s + 1)(s + 3)} \end{aligned}$$

   and is very different from $L_{Kx}(s)$.
2. A Kalman filter with asymptotic LTR $(q = 100)$:

   $$\begin{aligned} &(F = A - LC, T, L, K_Z, K_y) \\ &= \left(\begin{bmatrix} 0 & -206.7 \\ 1 & -102.4 \end{bmatrix}, I, \begin{bmatrix} 203.7 \\ 98.4 \end{bmatrix}, K, 0 \right) \end{aligned}$$

whose corresponding loop transfer function is similarly computed as

$$L(s) = \frac{-(1191s + 5403)}{s^2 + 112.4s + 49.7} \times \frac{s + 2}{(s + 1)(s + 3)}$$

This is already the best LTR result of Doyle and Stein [1979]. It is achieved by a high-input noise level $q = 100$ and the associated large filter gain $(\| L \| = 226.2)$, which is extremely undesirable. The poles of this filter are around -2 and -100. Nonetheless, $L(j\omega)$ is still very different from $L_{Kx}(j\omega)$ at $\omega < 10$ (see Fig. 3 of Doyle and Stein, 1979).

The simple and exact LTR result is derived as [Tsui, 1988b]

$$(F, T, L, K_Z, K_y) = (-2, [1 - 2], 1, 30, 10)$$

It can be verified that the corresponding $L(s) = L_{Kx}(s)$, which is guaranteed by $TB = 0$ (see Theorem 3.4). The observer gain $L = 1$ while the output gain of this observer $[K_Z : K_y]$ is less than K. It should be noted that there is no explicit state estimation in this design.

The original example of Doyle and Stein [1979] used the dual controllable canonical form (A', C', B') to represent the same plant system. The corresponding state feedback K is $[\,50 \quad 10\,]$, and the corresponding gains L for the above two observers of Doyle and Stein [1979] were $[\,30 \quad -50\,]$ and $[\,6.9 \quad 84.6\,]$, respectively. Nonetheless, all compatible transfer functions and loop transfer functions of Example 4.5 and Doyle and Stein [1979] remain the same.

This example shows that the asymptotic LTR result is far from satisfactory. It also shows the effectiveness of the design concept of not explicitly estimating plant system states.

The plant system of Example 4.5 has $n - m$ stable transmission zeros and therefore satisfies the first of the above two sets of restrictions for exact LTR. The real advantage of the new design approach (of not requiring state estimation) of this book is for systems *not* satisfying these two sets of restrictions. Several such examples will be illustrated in Sec. 6.2, after the explicit algorithms of (4.1) and (4.3) are described.

A reason that only state observers [satisfying (4.1) and nonsingular $\overline{C}$ *together*] are involved in the existing LTR results concerns the difficulty in deriving a really satisfactory solution of (4.1), as was true in the minimal order observer design (see Example 4.3).

To summarize, the solving of (4.1) *and* (4.3) [but not nonsingular $\overline{C}$] is not a retreat into a simpler design approach nor an avoidance of arbitrary state feedback implementation, but a necessary and difficult step to eliminate the very unsatisfactory state of the existing LTR results, and a novel step which is enabled *only* by a technical breakthrough in the solution of (4.1) [the remaining freedom of (4.1) is fully used to satisfy (4.3)].

4.4 A NEW DESIGN APPROACH AND NEW FEEDBACK STRUCTURE—A DYNAMIC OUTPUT FEEDBACK COMPENSATOR THAT GENERATES STATE/ GENERALIZED STATE FEEDBACK CONTROL SIGNAL

The conclusions of the first three sections of this chapter can be listed as follows.

Conclusion 4.1

Equation (4.1) is a necessary and sufficient condition for an observer (3.16) to generate a signal $K\mathbf{x}(t)$ for a constant K, where $\mathbf{x}(t)$ is the plant system state vector (Theorem 3.2).

Conclusion 4.2

Equation (4.1) is also the sufficient condition for the observer feedback system poles to be composed of the eigenvalues of F and of $A - BK$, where $K\mathbf{x}(t)$ is the state feedback generated by the observer (3.16) (Theorem 4.1). This theorem guarantees the observer feedback system performance.

Conclusion 4.3

For a freely designed state feedback $K\mathbf{x}(t)$ ($K = \overline{KC}, \overline{C} = [T' : C']'$ is determined and $\overline{K}$ is completely free), the necessary and sufficient condition for the observer feedback system to realize the robustness properties of this $K\mathbf{x}(t)$ is (4.3) (or $TB = 0$, Theorem 3.4).

Conclusion 4.4

To satisfy (4.1), (4.3), and a nonsingular $\overline{C}$, or to have an exact LTR state observer, the plant system either must have $n - m$ stable transmission zeros or satisfy (1) minimum-phase, (2) rank $(CB) = p$, and (3) $m \geqslant p$ [Kudva et al., 1980]. Most practical plant systems do not satisfy these restrictions. The other existing asymptotic LTR state observer is far from satisfactory either, mainly because of its asymptotic large gain.

Because of this conclusion, even though the ideally and separately designed state feedback can always be implemented by a state observer, its ideal robustness property is lost in the actual observer feedback system in most cases. This is intolerable because robustness is a key property of most engineering systems. Conversely, even though a state observer has generated the desired state feedback control signal (even optimally in a minimal variance sense), the purpose of this state observer is also lost because it has failed to realize the critical robustness properties of the same state feedback control in a *deterministic* sense.

The reason for this state of existing results of Conclusion 4.4 can be interpreted as follows. Because the state feedback control K is designed *separately* from the state observers, the state observers are expected to implement *arbitrarily given* state feedback. This is proven to be too much of a requirement if the LTR requirement (4.3) is added.

Let us analyze the above situation from another different perspective. The direct (and ideal) state feedback is designed based on the dynamic matrix $A - BK$ or the information of the plant system's input dynamic part (A, B) *only*, and is separated completely from the knowledge of plant system's output observation (with key parameter C) and the knowledge of the observer (with key parameter T) which actually realizes and implements it. Therefore such design cannot be considered mature and is not based on complete information. This immaturity is reflected by the fact that the resulting state feedback control *and* its robustness property cannot be actually realized in most cases if the states are not all directly measurable, even though such a state feedback control is itself ideal and superb (see Subsection 3.2.1).

Based on the above conclusions and analysis, this book proposes a fundamentally new design approach. In this new approach, the state feedback control is designed based on the feedback system dynamic matrix $A - B\overline{KC} \triangleq A - B\overline{K}[T' : C']'$, which comprises the information of not only the plant system's input dynamic part (A, B), but also other plant system parameter C and observer parameter T. The new state feedback control is guaranteed of observer implementation, separation property, and robustness realization for significantly more general cases. Thus this new approach is mature and is divided naturally into the following two major steps.

The first step determines the observer dynamic part (3.16a) by solving (4.1) and using the remaining freedom of (4.1) to best satisfy (4.3). Thus the resulting observer is able to generate a state feedback signal $K\mathbf{x}(t)$ with a constant $K = \overline{KC}$ (see Conclusion 4.1) and, for whatever this K, the feedback system poles of this observer are guaranteed to be the eigenvalues of $A - BK$ and F (see Conclusion 4.2). In addition, every effort has been made to realize the robustness property of this state feedback control (see Conclusion 4.3).

The design algorithms of this step are described in Chaps 5 and 6. Condition (4.1) is satisfied first in Chap. 5 and for all plant systems, and

(4.3) is then best satisfied in Chap. 6. It is proved in Sec. 6.2 that for all plant systems either with at least one stable transmission zero or with $m > p$, the exact solution of (4.1) and (4.3) can be computed, with the rank of matrix $\overline{C}$ also maximized by the available remaining freedom of (4.1) and (4.3). This is significantly more general than the existing exact LTR state observers, and is general for most plant systems (see Exercises 4.3 and 4.7). For all other plant systems, the least square solution of (4.3) can be computed, without large gain.

The second step fully determines the output part of the observer (3.16b) by designing the dynamic matrix $A - B\overline{KC}$, where $\overline{K}$ is the completely free parameter of (3.16b). The loop transfer function $L_{Kx}(s)$ is indirectly (though much more effectively) determined by this design (see Chaps 2, 3, 8 and 9). The explicit design algorithms are described in Chaps 8 and 9.

It should be noted that the design of $A - B\overline{KC}$ is exactly compatible mathematically with the static output feedback design $A - BK_yC$ of Subsection 3.2.2. The only difference is that rank $(C) = m$ while rank $(\overline{C}) = r + m \geqslant m$, where r is the number of rows of T, or the order of the observer of the first step.

In addition, because the rank of $\overline{C}$ of the first step can be between n and m, this new design approach unifies completely the exact LTR state observer, which corresponds to rank $(\overline{C}) =$ maximum n, and the static output feedback, which corresponds to rank $(C) = m =$ minimum of rank $(\overline{C})$. This unification will be discussed in Sec. 6.3. In this sense, we also call the feedback control which is implemented by this new observer as the "generalized state feedback control."

Because (4.3) $(TB = 0)$ is satisfied in the first step of this design approach, the corresponding observer of (3.16) will have the following state space model

$$\dot{\mathbf{z}}(t) = F\mathbf{z}(t) + L\mathbf{y}(t) \tag{4.10a}$$

$$-K\mathbf{x}(t) = -K_Z\mathbf{z}(t) - K_y\mathbf{y}(t) \tag{4.10b}$$

Because this observer (which is also called "feedback compensator") is not involved with the plant system input $\mathbf{u}(t)$, we call it the "output feedback compensator." In addition, compared to static output feedback systems of Sec. 3.2.2, this compensator has an additional dynamic part with state $\mathbf{z}(t) \Rightarrow T\mathbf{x}(t)$, and the control signal produced by this compensator has an additional term $K_Z\mathbf{z}(t) \Rightarrow K_ZT\mathbf{x}(t)$, which is provided by the above additional dynamic part. Therefore this compensator completely unifies the static output feedback as its simplest case and is called a "dynamic output feedback compensator."

The feedback system of this compensator is depicted in the block diagram in Fig. 4.3.

Finally, let us clarify three technical arguments concerning this new design approach.

First, the ideal compensator does not universally exist in practice. The significant advantage of this new design approach in very significantly more general robustness realization (see Exercises 4.2, 4.3, 4.6, and 4.7) certainly has its price—the constrained and therefore weaker state feedback control $K\mathbf{x}(t) = \overline{KC}\mathbf{x}(t)$ (if rank $(\overline{C}) < n$). This is the most serious criticism that has been downgrading the new design approach of this book. The following four points will fully answer this criticism.

1. The existing non-constrained and ideal state feedback control is designed ignoring the key parameters T of the realizing observer and the key parameter C of system output measurement. This is why its critical robustness properties cannot be actually realized for almost all open loop system conditions (see Sec. 4.3 and Part (b) of Exercises 4.2 and 4.6). Then what is the actual advantage of this existing control?
2. Although our generalized state feedback control is a constrained state feedback control, this constraint (based on $\overline{C} \triangleq [T' : C']'$) itself implies that the design of our control does not ignore the realization of this control when not all system state variables are directly measurable. This is why the robustness properties of our control are fully realized.
3. Although our control is a constrained state feedback control, it can achieve the very effective high performance and robustness

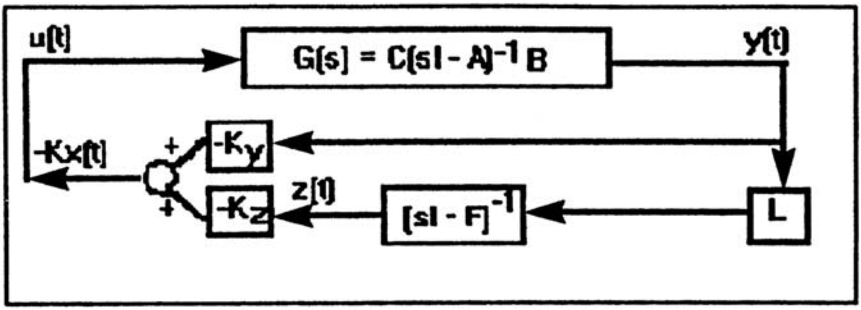


Figure 4.3 Dynamic output feedback compensator which can implement state feedback control—the new result of this book.

control—arbitrary pole assignment and partial eigenvector assignment, for a very large portion of open loop systems (see Exercise 4.8). Notice that the assumption of Exercise 4.8 has been criticized by many as too unfavorable. In addition, our control can virtually guarantee the stability of its feedback system (see Exercise 4.9, which has the same assumption as that of Exercise 4.8.)

4. Although our control is a constrained state feedback control, this constraint itself enabled the complete unification of the well-established state feedback control and static output feedback control (see Sec. 6.3). These two existing controls are the extreme cases of our control in all basic senses such as the control constraint [no constraint and most constraint (rank ($\overline{C}$) = minimum m)] and the controller order (maximum $n - m$ and minimum 0). Then why accepting these very undesirable extremes while rejecting their very reasonable modifications and adjustments (our rank($\overline{C}$) is maximized by all existing remaining design freedom)?

Of course for any theoretical result, no matter how practical, reasonable, and favorable, one can always device a special example to beat it. For example the key proof of the general effectiveness of our control is based on the assumption of $m = p$ and $n - m$ transmission zeros (see Point 3 above). A system of $m = p$ generically have $n - m$ transmission zeros, [Davison and Wang, 1974]. However, this well-known fact does prevent the publication of a comment, which uses a single special system example with $m = p$ but without transmission zeros, to criticize this new design approach. The well known fact that even under this cooked-up special example, our design result is still much better than the existing ones, does not prevent the publication and the subsequent quotation of this comment either [Tsui, 1996b].

To conclude, the criticism of this new design approach that it cannot have both ideal control and full realization of this control under all open loop system conditions, and the rejection of this new design approach because of this criticism, is unrealistic and unfair.

Second, this new design approach naturally answers another design problem, that is, given a plant system (A, B, C), determine the state feedback gain K which can stabilize the matrix $A - BK$ and which can be realized by an observer with $L(s) = L_{Kx}(s)$. The answer provided by this new design approach is that $A - B\overline{KC}$ is stabilizable, where $\overline{C}$ is fully determined. However, another answer to this problem is that K must stabilize matrix $A - BK$ and satisfy (3.28a) [Saberi et al., 1991]. As described in Theorem 3.4, some special K which can stabilize $A - BK$ may satisfy

(3.28a) but not (3.29) = (4.3). In other words, this special K cannot be derived by this new design approach which is based on (4.3), or (4.3) is not as necessary as (3.28a).

However, as discussed following Theorem 3.4, the design of K that stabilizes $A - BK$ must also be constrained by (4.1), (4.2), and (3.28a). Such a design is obviously a theoretical formulation or reformulation only, but impossible to find a direct, systematic, and general solution because K is heavily constrained (K_Z or $\overline{K}$ is constrained). On the other hand, the design of our K to stabilize $A - BK$ is constrained only by $K = \overline{KC}$ ($\overline{K}$ is *not* constrained), and a direct, systematic, and general solution of this design *can* be derived but is also difficult enough (see Syrmos et al., 1994 and Secs 8.1.3 and 8.1.4). In addition, Theorem 3.4 shows that for a free $\overline{K}$, our (4.3) is an equivalent of (3.28a).

This situation is very similar to the theoretical argument on the necessity of (4.1) to Separation Property (Theorem 4.1). In a challenging design, the system parameters (including K) simply cannot be cooked to fit the special case of Example 4.4 in which (4.1) is unnecessary to Theorem 4.1. Similarly, the cooking up of the system parameters to fit a special case that satisfies (3.28a) but not (4.3), although arguable theoretically, is totally meaningless in practical design.

To conclude, the criticism that the formulation (4.3) of this new design approach is not as necessary as (3.28a) for some special cases—and the rejection of this new design approach because of this criticism—ignores the basic difference between the systematic and practical design (in which at least $\overline{K}$ should be free) and the theoretical formulation or reformulation, and is therefore unrealistic and unreasonable.

Third and finally, the dynamic output feedback compensator structure has certainly appeared before. For example, some such compensators have been designed for eigenstructure assignment [Misra and Patel, 1989; Duan, 1993b], and some others have been designed from the perspective of LTR [Chen et al., 1991].

However, none of these existing results satisfies separation property generally. Because (4.1) is the sufficient condition of separation property (Theorem 4.1) and the necessary and sufficient condition for the compensator to generate a state feedback signal $K\mathbf{x}(t)$ for a constant K, the other existing dynamic output feedback compensators cannot generate a signal $K\mathbf{x}(t)$ for a constant K. Not satisfying the separation property also implies the possibility of an unstable compensator, even though the overall feedback system is designed satisfactorily.

To conclude, generating a signal $K\mathbf{x}(t)$ for a constant K is the fundamental feature of existing state space control structures considered to be well established (such as direct state feedback, static output feedback, and observer feedback structures, see Subsections 3.2.1–3.2.3) because it has several inherent properties and advantages. Because the existing dynamic output feedback compensator cannot generate $K\mathbf{x}(t)$ for a constant K, only the dynamic output feedback compensator of this book can be considered as well established [see Tsui, 1998b].

With the design approach of this book now formulated, the design algorithms and solutions to the design problems and formulations imposed by this approach will be described in the next five chapters (especially Chaps 5 and 6). The purpose is to design a controller with general feedback system performance *and* robustness against model uncertainty and input disturbance.

EXERCISES

4.1 Verify the results of Example 4.5, including its dual version.

4.2 It is very useful to measure the strictness of a constraint or a condition, by the probability of the plant systems that satisfy this constraint/condition. To derive a simple expression of this probability, we need to make the following two assumptions on the open-loop systems. These two open-loop system-based assumptions are unbiased in the sense that a good (or bad) assumption is equally good (or bad) to all design algorithms. An assumption that is much more favorable than that of Exercise 4.2 will be used for starting Exercise 4.6.

1. Let p_z be the constant probability for each plant system transmission zero to be stable. We assume $p_z = 1/2$ so that each plant system transmission zero is equally likely to be stable or unstable. This assumption is reasonable because the plant system parameters are supposed to be randomly given (so are the values and positions of plant system transmission zeros), and because the stable and unstable regions are almost equally sized.
2. We assume $m = p$ so that the number of system transmission zeros is simply $n - m$ [Davison and Wang, 1974] and so that the rank $(\overline{C} \underset{=}{\triangle} [T' : C']')$ is simply $m + r$, where r is the number of stable transmission zeros out of the $n - m$ transmission zeros.

(a) Based on this assumption, compute the P_r as the probability of r stable transmission zeros out of $n - m$ transmission zeros. $P_r = [r : n - m](p_z)^r(1 - p_z)^{n-r}$, where $[r : n - m]$ is the combination of r elements out of $n - m$ elements.

Answer:

$n-m=$	1	2	3	4	5	6	7	8	9
r									
0	$1/2$	$1/2^2$	$1/2^3$	$1/2^4$	$1/2^5$	$1/2^6$	$1/2^7$	$1/2^8$	$1/2^9$
1	$1/2$	$2/2^2$	$3/2^3$	$4/2^4$	$5/2^5$	$6/2^6$	$7/2^7$	$8/2^8$	$9/2^9$
2		$1/2^2$	$3/2^3$	$6/2^4$	$10/2^5$	$15/2^6$	$21/2^7$	$28/2^8$	$36/2^9$
3			$1/2^3$	$4/2^4$	$10/2^5$	$20/2^6$	$35/2^7$	$56/2^8$	$84/2^9$
4				$1/2^4$	$5/2^5$	$15/2^6$	$35/2^7$	$70/2^8$	$126/2^9$
5					$1/2^5$	$6/2^6$	$21/2^7$	$56/2^8$	$126/2^9$
6						$1/2^6$	$7/2^7$	$28/2^8$	$84/2^9$
7							$1/2^7$	$8/2^8$	$36/2^9$
8								$1/2^8$	$9/2^9$
9									$1/2^9$

(b) Based on the result of Part (a), find the probability of minimum-phase ($r = n - m$) for $n - m = 1$ to 8.

Answer:

$P_{n-m} = 0.5, 0.25, 0.125, 0.0625, 0.03125, 0.0156, 0.0078, 0.0036$. This probability is too low (and rapidly lower as $n - m$ increases) to be acceptable.

4.3 One of the sufficient conditions of the new design approach of this book is at least one stable transmission zero ($r > 0$, see Conclusion 6.1). Based on the assumption and result of 4.2, calculate the probability of $r > 0$ for $n - m = 1$ to 8, and compare this probability with the probability of minimum-phase P_{n-m} of Part (b) of 4.2.

Answer:

$P(r > 0) = 1 - P_0 = 0.5$, 0.75, 0.875, 0.9375, 0.9688, 0.9844, 0.9922, 0.9964.

The probability $P(r > 0)$ of this new design approach is almost 100% as soon as $n - m$ is > 3, and is very significantly greater than P_{n-m} (probability of minimum-phase, one of the necessary conditions of the existing LTR results).

4.4 The sufficient condition for the generalized state feedback control of this book to assign arbitrarily given poles and some eigenvectors is $r + m + p > n$, or $r > n - m - p$ [see (6.19) or Step 2 of Algorithm 8.1]. Based on the assumption and result of 4.2, calculate the probability of $r > n - m - p$ ($= 100\%$ if $\mathrm{n - m - p} < 0$).

Answer:

$n =$	3	4	5	6	7	8	9	10
$m = p =$	%	%	%	%	%	%	%	%
2	100	75	50	31.25	18.75	10.94	6.25	3.52
3	100	100	100	87.5	68.75	50	34.38	22.66
4	100	100	100	100	100	93.75	81.25	65.63

Compared to the popular static output feedback control, the probability to achieve this arbitrary pole assignment and partial eigenvector assignment is 0% in the above table if the number is not 100%. Thus the improvement of our generalized state feedback control from the static output feedback control is very significant.

4.5 The sufficient condition for the generalized state feedback control of this book to assign arbitrary poles and to guarantee stability is $(r + m)p > n$ or $r > n/p - m$ [see (6.18), Adjustment 2 of Sec. 8.1.4, and Wang, 1996]. Based on the assumption and result of 4.2, calculate the probability of $r > n/p - m$ (= 100% if $\mathrm{n/p - m} < 0$).
Answer:

$n =$	3	4	5	6	7	8	9	10	11	12
$m = p =$	%	%	%	%	%	%	%	%	%	%
2	100	75	88	69	81	66	77	63	75	
3	100	100	100	100	100	100	98	99	99+	98

The probability is very high as soon as m is increased higher than 2, and decreases very slowly so that no substantial decrease can be shown in the above table. For example, it can be calculated that when $n = 16$, the probability is still 98% for $m = 3$, and that when $n = 26$ the probability is still 99.7% for $m = 4$. This result indicates that the great majority of the open loop systems can be guaranteed of arbitrary pole assignment and stabilization by our generalized state feedback control.

Compared to the popular static output feedback control, the probability to achieve this arbitrary pole assignment is 0% in the above table if the number is not 100%. Thus the improvement of our generalized state feedback control from the static output feedback control is very significant.

4.6 Repeat 4.2 by changing p_z to 3/4. This new p_z implies that each plant system transmission zero is three times more likely to be stable than to be unstable. This p_z is significantly more favorable than the half-and-half p_z of 4.2 to 4.5, even though that old p_z is still reasonable (see 4.2). Therefore we may assume that most of the practical values of p_z would fall between these two values of p_z.

(a) *Answer*:
$P_r =$ (assume $p_z = 3/4$)

$n-m=$	1	2	3	4	5	6	7	8
r								
0	$1/4$	$1/4^2$	$1/4^3$	$1/4^4$	$1/4^5$	$1/4^6$	$1/4^7$	$1/4^8$
1	$3/4$	$6/4^2$	$9/4^3$	$12/4^4$	$15/4^5$	$18/4^6$	$21/4^7$	$24/4^8$
2		$9/4^2$	$27/4^3$	$54/4^4$	$90/4^5$	$135/4^6$	$189/4^7$	$252/4^8$
3			$27/4^3$	$108/4^4$	$270/4^5$	$540/4^6$	$945/4^7$	$1{,}512/4^8$
4				$81/4^4$	$405/4^5$	$1{,}215/4^6$	$2{,}835/4^7$	$5{,}670/4^8$
5					$243/4^5$	$1{,}458/4^6$	$5{,}103/4^7$	$13{,}608/4^8$
6						$729/4^6$	$5{,}103/4^7$	$20{,}412/4^8$
7							$2{,}187/4^7$	$17{,}496/4^8$
8								$6{,}561/4^8$

(b) Based on the result of Part (a), find the probability of minimum-phase ($r = n - m$) for $n - m = 1$ to 8.
Answer:
$P_{n-m} = 0.75, 0.56, 0.42, 0.32, 0.24, 0.18, 0.13, 0.1.$

Although much higher than the corresponding probabilities of Exercise 4.2, the P_{n-m} is still lower than $1/2$ as soon as $n - m > 2$, and is decreasing as $n - m$ increases.

4.7 One of the sufficient conditions of the new design approach of this book is at least one stable transmission zero ($r > 0$, see Conclusion 6.1). Based on the assumption and result of 4.6, calculate the probability of $r > 0$ for $n - m = 1$ to 8, and compare this probability with the probability of minimum-phase P_{n-m} of Part b of 4.6.
Answer:
$P(r > 0) = 1 - P_0 = 0.75, 0.9375, 0.9844, 0.9964, \ldots$
The probability $P(r > 0)$ of this new design approach is almost 100% as soon as $n - m > 1$, and is very significantly greater that P_{n-m} (probability of minimum-phase, one of the necessary conditions of the existing LTR designs).

4.8 The sufficient condition for the generalized state feedback control of this book to assign arbitrarily given poles and some eigenvectors is $r + m + p > n$, or $r > n - m - p$ (see (6.19) or Step 2 of Algorithm 8.1). Based on the assumption and result of 4.6, calculate the probability of $r > n - m - p$ ($= 100\%$ if $n - m - p < 0$).

$n =$ $m = p =$	3 %	4 %	5 %	6 %	7 %	8 %	9 %	10 %
2	100	94	84	74	63	53	44	37
3	100	100	100	98	97	90	83	76
4	100	100	100	100	100	99.6	98	96

Compare to the popular static output feedback control, the probability to achieve this arbitrary pole assignment and partial eigenvector assignment is 0% in the above table if the number is not 100%. Thus the improvement of our generalized state feedback control from the static output feedback control is very significant, is much more significant than that of Exercise 4.4, and makes this very effective and difficult design goal (see Chaps 8 and 9) achievable to a very large portion of practical systems. This table of data should be most relevant among all tables, to the practical and current high performance and robustness system design.

4.9 The sufficient condition for the generalized state feedback control of this book to assign arbitrary poles and to guarantee stability is $(r+m)p > n$ or $r > n/p - m$ [see (6.18), Adjustment 2 of Sec. 8.1.4, and Wang, 1996]. Based on the assumption and result of 4.6, calculate the probability of $r > n/p - m$ ($= 100\%$ in $n/p - m < 0$).

Answer:

$n =$ $m = p =$	3 %	4 %	5 %	6 %	7 %	8 %	9 %	10 %	11 %	12 %
2	100	94	98	95	98	96	99	97		
3	100	100	100	100	100	100	99+	99+	99+	99+

The probability is almost all 100%, and does not decrease as n increases. This result indicates that arbitrary pole assignment and stabilization are virtually guaranteed by our generalized state feedback control.

Compare to the popular static output feedback control, the probability to achieve this arbitrary pole assignment is 0% in the above table if the number is not 100%. Thus the improvement of our generalized state feedback control from the static output feedback control is very significant and much more significant than the improvement of Exercise 4.5 (which is based on a less favorable assumption of 4.2).

5

Solution of Matrix Equation $TA - FT = LC$

Chapter 4 proposed the new design approach of satisfying (4.1) and (4.3) first, and explained the necessity and advantages of this approach. The problem of solving (4.1) and (4.3) was first raised in Tsui [1987b]. Its satisfactory solution appeared in Tsui [1992], much delayed from its first verbal presentation at the 1990 American Control Conference. This solution has made this new design approach possible [Tsui, 2000].

The design algorithms of (4.1) and (4.3) are presented in Chaps 5 and 6, respectively. Chapter 5 has two sections.

Section 5.1 introduces the algorithm for computing the block-observable Hessenberg form of the plant system's state space model. Although this computation is unnecessary for the analytical solution of

(4.1), it significantly improves the numerical computation of this solution, and naturally separates the observable part from the unobservable part of the plant system.

Section 5.2 presents the solution of (4.1). It demonstrates also the analytical and computational advantages of this solution over other existing solutions of (4.1).

5.1 COMPUTATION OF A SYSTEM'S OBSERVABLE HESSENBERG FORM

5.1.1 Single-Output Systems

The Hessenberg form matrix is defined as follows:

$$A = \left[\begin{array}{ccccc} x & * & 0 & \dots & 0 \\ x & x & * & 0 & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & & 0 \\ \vdots & & & x & * \\ x & & \dots & \dots & x \end{array}\right] \tag{5.1}$$

where the elements "x" are arbitrary and the elements "*" are nonzero. The matrix of (5.1) is also called the "lower Hessenberg form" matrix. The transpose of the matrix form (5.1) is called the "upper Hessenberg form."

The Hessenberg form is the simplest possible matrix form which can be computed from a general matrix by orthogonal matrix operation without iteration. For example the Schur triangular form, which differs from the Hessenberg form by having all "*" entries of (5.1) equal 0, is computed by iterative methods (QR method).

In the established computational algorithms of some basic numerical linear algebra problems, whether in the QR method of computing matrix eigenstructure decomposition [Wilkinson, 1965] and singular value decomposition [Golub and Reinsch, 1970], or in the computation of solution of the Sylvester equation [Golub et al., 1979] and the Riccati equation (Laub, 1979, Sec. 8.1), the computation of the Hessenberg form has always been the first step [Laub and Linnemann, 1986]. As the first step of the design algorithm for solving (4.1), a special form of system matrix (A, C) called "observable Hessenberg form," in which matrix A is in the lower Hessenberg form of (5.1), is also computed [Van Dooren et al., 1978; Van Dooren, 1981]. The single-output case of this form is

$$\begin{bmatrix} CH \\ \cdots\cdots \\ H'AH \end{bmatrix} = \begin{bmatrix} * & 0 & & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ x & * & 0 & \dots & 0 \\ x & x & * & 0 & : \\ : & & \ddots & \ddots & 0 \\ : & & & x & * \\ x & & \dots & x & x \end{bmatrix} \tag{5.2}$$

where matrix H is an unitary similarity transformation matrix $(H'H = I)$ which transforms the plant system matrix pair $(A,\ C)$ into the form of (5.2).

The matrix H and its result (5.2) can be computed by the following algorithm.

Algorithm 5.1 Computation of Single-Output Observable Hessenberg Form System Matrix

Step 1: Let $j = 1, H = I, \mathbf{c}_1 = C$, and $A_1 = A$.
Step 2: Compute the unitary matrix H_j such that $\mathbf{c}_j H_j = [\,c_j, 0 \dots 0\,]$ (see Appendix A, Sec. 2).
Step 3: Compute

$$H_j' A_j H_j = \begin{bmatrix} a_{jj} & : & \mathbf{c}_{j+1} \\ \dots & \dots & \dots\dots \\ \mathbf{x} & : & A_{j+1} \\ & : & \end{bmatrix} \left.\vphantom{\begin{matrix} a \\ a \end{matrix}}\right\} n - j \tag{5.3}$$

Step 4: Update matrix

$$H = H \begin{bmatrix} I_{j-1} & : & 0 \\ \dots & \dots & \dots \\ 0 & : & H_j \\ & : & \end{bmatrix}$$

where I_{j-1} is an identity matrix with dimension $j - 1$.

Step 5: If $\mathbf{c}_{j+1}$ of (5.3) equals 0, then go to Step 7.

Step 6: Let $j = j + 1$ (so that $\mathbf{c}_{j+1}$ and A_{j+1} of (5.3) become $\mathbf{c}_j$ and A_j, respectively). If $j = n$ then go to Step 7; otherwise return to Step 2.

Step 7: The final result is

$$\begin{bmatrix} CH \\ \cdots\cdots \\ H'AH \end{bmatrix} = \left[\begin{array}{cccccl}
c_1 & 0 & \cdots & \cdots & 0 & :0\ldots 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & :\quad : \\
a_{11} & c_2 & 0 & \cdots & 0 & :\quad : \\
\vdots & a_{22} & \ddots & \ddots & \vdots & :\quad : \\
\vdots & \ddots & \ddots & & 0 & :\quad : \\
\vdots & & \ddots & & c_j & :\quad : \\
x & & & & a_{jj} & :0\ldots 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & :\cdots\cdots \\
 & & & X & & :\overline{A}_o
\end{array}\right]$$

$$\triangleq \left[\begin{array}{ccc}
C_o & : & 0 \\
\cdots & : & \\
A_o & : & 0 \\
\cdots & \cdots & \cdots \\
X & : & \overline{A}_o
\end{array}\right]\begin{array}{l} \\ \\ \}j \\ \\ \}n-j \end{array} \tag{5.4}$$

where the matrix pair (A_o, C_o) is in the observable Hessenberg form of (5.2) and is separated from the unobservable part of the system $\overline{A}_o$. The dimension of this observable part is j but will be replaced by n elsewhere in this book because only observable systems are being considered. (see Exercise 5.4).

5.1.2 Multiple Output Systems

In multi-output systems of m outputs, C is a matrix of m rows and is no longer a row vector. The corresponding observable Hessenberg form in this case is the so-called block-observable Hessenberg form, as in:

$$\begin{bmatrix} CH \\ \cdots\cdots \\ H'AH \end{bmatrix} = \begin{array}{c} \left[\begin{array}{cccccc} C_1 & 0 & \cdots\cdots & \cdots\cdots & \cdots\cdots & 0 \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ A_{11} & C_2 & 0 & \cdots\cdots & \cdots\cdots & 0 \\ \vdots & A_{22} & C_3 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & C_v \\ A_{v1} & \cdots & & & & A_{vv} \end{array}\right] \begin{array}{l} \}m_0 \\ \\ \}m_1 \\ \}m_2 \\ \vdots \\ \vdots \\ \}m_{v-1} \\ \}m_v \end{array} \\ \begin{array}{cccccc} m_1 & m_2 & \cdots\cdots\cdots\cdots\cdots\cdots & & & m_v \end{array} \end{array} \tag{5.5}$$

where the A_{ij} is an $m_i \times m_j$ dimensional arbitrary matrix block, and C_j $(j = 1, \ldots, v)$ is an $m_{j-1} \times m_j$ dimensional "lower-echelon matrix" $(m_0 = m \geqslant m_1 \geqslant m_2 \geqslant \cdots \geqslant m_v > 0)$. We will use the following example to illustrate the lower-echelon-form matrix.

Example 5.1

All lower-echelon-form matrices with three rows are in the following seven different forms:

$$\begin{bmatrix} * & 0 & 0 \\ x & * & 0 \\ x & x & * \end{bmatrix}, \begin{bmatrix} * & 0 \\ x & * \\ x & x \end{bmatrix}, \begin{bmatrix} * & 0 \\ x & 0 \\ x & * \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ * & 0 \\ x & * \end{bmatrix}, \begin{bmatrix} * \\ x \\ x \end{bmatrix}, \begin{bmatrix} 0 \\ * \\ x \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ * \end{bmatrix}$$

where "x" entries are arbitrary and "*" entries are nonzero.

From Sec. A.2 of Appendix A, there exists a unitary matrix H_j such that $\overline{C}_j H_j = [\, C_j, \quad 0 \quad \ldots \quad 0\,]$ for any matrix $\overline{C}_j$ with m_{j-1} rows, where C_j is an $m_{j-1} \times m_j$ dimensional lower-echelon matrix.

From Example 5.1, all m_j columns of C_j are linearly independent of each other, and so are the m_j rows (those with a "*" element) of C_j. Each of the other $m_{j-1} - m_j$ rows (those without a "*" element) can always be expressed as a linear combination of the linearly independent rows which are above this linearly dependent row in C_j (see Sec. A.2 of Appendix A).

Example 5.2

In the last six of the seven matrices of Example 5.1, the linearly dependent rows are, respectively, the 3rd, the 2nd, the 1st, the 3rd and 2nd, the 3rd and 1st, and the 2nd and 1st of the corresponding matrix. All three rows of the first matrix are linearly independent of each other.

For each of the last six of the seven matrices C_j $(j = 2, \ldots, 7)$ of Example 5.1, there exists at least one row vector $\mathbf{d}_j$ such that $\mathbf{d}_j C_j = 0$. For example, $\mathbf{d}_j = [x \quad x \quad 1], [x \quad 1 \quad 0], [1 \quad 0 \quad 0], [x \quad 0 \quad 1]$ or $[x \quad 1 \quad 0], [0 \quad x \quad 1]$ or $[1 \quad 0 \quad 0]$, and $[0 \quad 1 \quad 0]$ or $[1 \quad 0 \quad 0]$, for $j = 2, \ldots, 7$, respectively. It is clear that in these $\mathbf{d}_j$ vectors, the position of element "1" always corresponds to the linearly dependent row of the corresponding C_j, while all "x" elements are the linear combination coefficients for that linearly dependent row. It is also clear that these coefficients correspond only to the linearly independent rows which are above that linearly dependent row.

Without loss of generality, we assume $m_1 = m$, so that all m system outputs are linearly independent [Chen, 1984]. In other words, each row of matrix C corresponds to a linearly independent output. As in the single-output case, during the computation of the block-observable Hessenberg form, if a row of C_j becomes 0 or becomes linearly dependent on its previous rows of C_j, then the corresponding output is no longer influenced by more system states. Thus this row/column will disappear at the subsequent $C_i (i > j)$ blocks (or no longer appear at the observable part of the system). With this adaptation, Algorithm 5.1 can be generalized to the following Algorithm 5.2 for multi-output case.

Algorithm 5.2 Computation of Block-Observable Hessenberg Form

Step 1: Let $j = 1, H = I, \overline{C}_1 = C, A_1 = A, m_0 = m$, and $n_0 = 0$.

Step 2: Compute a unitary matrix H_j such that $\overline{C}_j H_j = [C_j, 0 \ldots 0]$, where C_j is an $m_{j-1} \times m_j$ dimensional lower echelon matrix.

Step 3: Compute

$$H_j' A_j H_j = \begin{array}{cc} \left[\begin{array}{ccc} A_{jj} & : & \overline{C}_{j+1} \\ \cdots & \cdots & \cdots\cdot \\ X & : & A_{j+1} \\ & : & \end{array}\right] & \begin{array}{l} \}m_j \\ \\ \\ \\ \end{array} \\ \begin{array}{ccc} m_j & & \phantom{\overline{C}_{j+1}} \end{array} & \end{array} \qquad (5.6)$$

Step 4: Update matrix

$$H = H \begin{bmatrix} I^j & \vdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \vdots & H_j \\ & \vdots & \end{bmatrix}$$

where I^j is an identity matrix with dimension n_{j-1}.

Step 5: $n_j = n_{j-1} + m_j$. If $n_j = n$ or if $\overline{C}_{j+1} = 0$, then let $v = j$ and go to Step 7.

Step 6: Let $j = j + 1$ (so that the $\overline{C}_{j+1}$ and A_{j+1} of (5.6) become $\overline{C}_j$ and A_j, respectively), and return to Step 2.

Step 7: The final result is

$$\begin{bmatrix} CH \\ \cdots\cdots \\ H'AH \end{bmatrix} = \left[\begin{array}{ccccccccc} C_1 & 0 & \cdots & \cdots & 0 & \vdots & 0 & \cdots & 0 \\ \cdots\cdot & \cdots\cdot & \cdots\cdot & \cdots\cdot & \cdots\cdot & \vdots & & & \\ A_{11} & C_2 & \ddots & & \vdots & \vdots & & & \\ \vdots & A_{22} & \ddots & \ddots & \vdots & \vdots & & & \\ \vdots & \ddots & \ddots & \ddots & 0 & \vdots & & & \\ \vdots & & \ddots & \ddots & C_v & \vdots & & & \\ A_{v1} & & & \ddots & A_{vv} & \vdots & 0 & \cdots & 0 \\ \cdots\cdot & \cdots\cdot & \cdots\cdot & \cdots\cdot & \cdots\cdot & \cdots & \cdots\cdot & & . \\ & & X & & & \vdots & \overline{A}_o & & \end{array}\right]\}m$$

$$= \begin{bmatrix} C_o & \vdots & 0 \\ \cdots & \vdots & \\ A_o & \vdots & 0 \\ \cdots\cdot & \cdots & \cdots \\ X & \vdots & \overline{A}_o \end{bmatrix} \tag{5.7}$$

where the matrix pair (A_o, C_o) is already in the block-observable Hessenberg form (5.5) and is already separated from the unobservable part $\overline{A}_o$ of the system. The dimension of A_o is $n_j = m_1 + \cdots + m_v$ (replaced by n in the rest of this book).

It is clear that Algorithm 5.2 is a generalized version of Algorithm 5.1, when the parameter m is generalized from 1 to m ($\geqslant 1$). The main computation of this algorithm is at Step 3. According to Sec. A.2 of Appendix A, the order of computation of this algorithm (based on Step 3) is about $4n^3/3$.

Definition 5.1

From the description of the block-observable Hessenberg form (5.5), each row of matrix C of (5.5) corresponds to one of system outputs and is linked to one more system state if that row is linearly independent in the next matrix block C_{j+1} of (5.5). Therefore the number of system states which influence the i-th system output equals the number of matrix blocks C_j in which the i-th row of C is linearly independent. We define this number as the i-th observability index $\nu_i, i = 1, \ldots, m$.

It is clear that $\nu_i = j$ if that i-th row becomes linearly dependent in matrix block C_{j+1}, and that $\nu_1 + \cdots + \nu_m = n$. It is also clear that $\max\{\nu_i\} = \nu$ of Step 5, and that all observability indices can be determined by Algorithm 5.2.

Another set of parameters $m_j, j = 1, \ldots, \nu$ of (5.5) can also be used to indicate the observability index. From the description of (5.5) and Definition 5.1, m_j indicates the number of observability indices which are $\geqslant j$.

Example 5.3

Let the block-observable Hessenberg form of a four-output and ninth-order system be

$$\begin{bmatrix} C \\ A \end{bmatrix} = \begin{bmatrix} C_1 & 0 & 0 & 0 \\ \\ A_{11} & C_2 & 0 & 0 \\ A_{21} & A_{22} & C_3 & 0 \\ A_{31} & A_{32} & A_{33} & C_4 \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}$$

$$
= \begin{bmatrix}
* & 0 & 0 & 0: & 0 & 0 & 0 & 0 & 0 \\
x & + & 0 & 0: & 0 & 0 & 0 & 0 & 0 \\
x & x & \& & 0: & 0 & 0 & 0 & 0 & 0 \\
x & x & x & \#: & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & \\
x & x & x & x: & * & 0 & 0: & 0 & 0 \\
x & x & x & x: & x & + & 0: & 0 & 0 \\
x & x & x & x: & x & x & 0: & 0 & 0 \\
x & x & x & x: & x & x & \#: & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots\cdots & \cdots & \cdots & \cdots & \cdots & \cdot\cdot \\
 & & & & & & & & \\
x & x & x & x: & x & x & x: & 0: & 0 \\
x & x & x & x: & x & x & x: & +: & 0 \\
x & x & x & x: & x & x & x: & x: & 0 \\
\cdots & \cdots & \cdots & \cdots\cdots & \cdots & \cdots & \cdots & \cdots & \cdot\cdot \\
x & x & x & x: & x & x & x:x & : & + \\
\cdots & \cdots & \cdots & \cdots\cdots & \cdots & \cdots & \cdots & \cdots & \cdot\cdot \\
x & x & x & x: & x & x & x: & x: & x
\end{bmatrix}
\begin{array}{l}
\left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} m_0 = m = 4 \\
\\
\left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} m_1 = 4 \\
\\
\\
\left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} m_2 = 3 \\
\\
\} m_3 = 1 \\
\\
\} m_4 = 1
\end{array}
\qquad (5.8a)
$$

From Definition 5.1, corresponding to the four system outputs which are represented by the nonzero elements with symbols "*," "+," "&," and "#," respectively, the observability indices are $v_1 = 2, v_2 = 4, v_3 = 1$, and $v_4 = 2$. These indices also equal the number of appearances of the corresponding symbols in (5.8a). We can verify that $v_1 + v_2 + v_3 + v_4 = m_1 + m_2 + m_3 + m_4 = n = 9$, and that $v = v_2 = 4$. We can also verify that m_j equals the number of observability indices which are greater than or equal to j $(j = 1, \ldots, v = 4)$.

In the literature Chen [1984], the block-observable Hessenberg form (5.8a) can be further transformed to the block-observable canonical form (1.16) by elementary similarity transformation:

$$
\begin{bmatrix} CE \\ E^{-1}AE \end{bmatrix} = \begin{bmatrix}
I_1 & 0 & 0 & 0 \\
 & & & \\
A_1 & I_2 & 0 & 0 \\
A_2 & 0 & I_3 & 0 \\
A_3 & 0 & 0 & I_4 \\
A_4 & 0 & 0 & 0
\end{bmatrix}
$$

$$
= \begin{bmatrix}
1 & 0 & 0 & 0 : 0 & 0 & 0 & 0 & 0 \\
x & 1 & 0 & 0 : 0 & 0 & 0 & 0 & 0 \\
x & x & 1 & 0 : 0 & 0 & 0 & 0 & 0 \\
x & x & x & 1 : 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & \\
x & x & x & x : 1 & 0 & 0 : & 0 & 0 \\
x & x & x & x : 0 & 1 & 0 : & 0 & 0 \\
x & x & x & x : 0 & 0 & 0 : & 0 & 0 \\
x & x & x & x : 0 & 0 & 1 : & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots\cdots & \cdots & \cdots & \cdots & \cdot\cdot \\
 & & & & & & & & \\
x & x & x & x : 0 & 0 & 0 : & 0 : & 0 \\
x & x & x & x : 0 & 0 & 0 : & 1 : & 0 \\
x & x & x & x : 0 & 0 & 0 : & 0 : & 0 \\
\cdots & \cdots & \cdots & \cdots\cdots & \cdots & \cdots & \cdots & \cdot\cdot \\
x & x & x & x : 0 & 0 & 0 : & 0 : & 1 \\
\cdots & \cdots & \cdots & \cdots\cdots & \cdots & \cdots & \cdots & \cdot\cdot \\
x & x & x & x : 0 & 0 & 0 : & 0 : & 0
\end{bmatrix} \tag{5.8b}
$$

where matrix E represents elementary matrix operations [Chen, 1984] and is usually not a unitary matrix.

The comparison of (5.8a) and (5.8b) shows that the block-observable canonical form is a special case of the block-observable Hessenberg form, in the sense that in (5.8b), all nonzero elements (those symbols) of C_j blocks of (5.8a) become 1, and all other arbitrary "x" elements of (5.8a) except those in the left m_1 columns become 0.

Although the parameters of a block-observable canonical form system matrix can be substituted directly into the polynomial matrix fraction description of its corresponding transfer function $G(s) = D^{-1}(s)N(s)$ (see Example 1.7), this unique advantage is offset by the unreliability of its computation (matrix E of (5.8b) is usually ill conditioned [Wilkinson, 1965]). For this reason, the actual design algorithms of this book are based only on the observable Hessenberg form.

5.2 SOLVING MATRIX EQUATION $TA - FT = LC$

The computational algorithm for the solution of matrix equation (4.1) ($TA - FT = LC$) is presented in this section. Here the $n \times n$ and $m \times n$ dimensional system matrices (A, C) are given and are observable. The number of rows of solution (F, T, L) is presumed to be $n - m$, although this number is freely adjustable because each row of this solution will be completely decoupled.

To simplify the computation of solution of (4.1), we have computed block-observable Hessenberg form ($H'AH$, CH) in Algorithm 5.2. Substituting ($H'AH$, CH) into (4.1), we have $T(H'AH) - FT = L(CH)$, which implies that the solution matrix T of this equation must be postmultiplied by H', in order to be recovered to the solution (TH'), which corresponds to the original (A, C).

Mathematically, the eigenvalues ($\lambda_i, i = 1, \ldots, n - m$) of matrix F of (4.1) can be arbitrarily given. We will, however, select these eigenvalues based on the following analytical understandings.

First, these eigenvalues must have negative and sufficiently negative real parts in order to achieve observer stability and sufficiently fast convergence of observer output to $K\mathbf{x}(t)$ (Theorem 3.2).

Second, the magnitude of these eigenvalues cannot be too large because it would cause large observer gain L (see Secs. 3.1 and 4.3).

Third, each plant system stable transmission zero must be matched by one of the eigenvalues of F. This is the necessary condition for the corresponding rows of T to be linearly independent if TB=0 (see Sec. 6.2).

Finally, all $n - m$ eigenvalues of F are the transmission zeros of the corresponding observer feedback system [Patel, 1978] and should be selected with the properties of transmission zeros in mind (see Sec. 1.4).

There have been some other suggestions for the selection of eigenvalues of F, but they are unsatisfactory. For example, the suggestion that the eigenvalues of F other than those which matched the stable transmission zeros be negative infinity with Butterworth pattern, is criticized by Sogaard-Andersen [1987]. The other suggestion that all eigenvalues of F be clustered around the plant system stable transmission zeros causes near singular matrix $\overline{C} = [T' : C']'$, and therefore large and unsatisfactory observer output gain $\overline{K}$ in (4.2) [Tsui, 1988b]. Hence the eigenvalues of F should be selected by following the preceding four guidelines.

Once the eigenvalues of matrix F are selected, the matrix F is required in our algorithm to be a Jordan form matrix, with all multiple eigenvalues forming a single Jordan block [see (1.10)]. Hence the matrix F is fully determined. In addition, each row or each block of rows of solution (F, T, L) corresponding to a Jordan block of F is decoupled and can be separately computed. Therefore, our algorithm treats the following two cases of Jordan block size ($= 1$ or > 1), separately.

5.2.1 Eigenstructure Case A

For distinct and real eigenvalue λ_i ($i = 1, \dots, n - m$) of F, (4.1) can be partitioned as

$$\mathbf{t}_i A - \lambda_i \mathbf{t}_i = \mathbf{1}_i C, \qquad i = 1, \dots, n - m \tag{5.9}$$

where $\mathbf{t}_i$ and $\mathbf{l}_i$ are the i-th row of matrix T and L corresponding to λ_i, respectively.

Based on the observable Hessenberg form (5.5) where

$$C = [\, \underset{m}{C_1} \quad 0 \quad \dots \quad 0 \,]$$

Eq. (5.9) can be partitioned as the left m columns

$$\mathbf{t}_i (A - \lambda_i I) \begin{bmatrix} I_m \\ 0 \end{bmatrix} = \mathbf{1}_i C \begin{bmatrix} I_m \\ 0 \end{bmatrix} = \mathbf{1}_i C_1 \tag{5.10a}$$

and the right $n - m$ columns

$$\mathbf{t}_i (A - \lambda_i I) \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} = \mathbf{1}_i C \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} = 0 \tag{5.10b}$$

Because C_1 of (5.5) is of full-column rank and $\mathbf{1}_i$ is free, (5.10a) can always be satisfied by $\mathbf{1}_i$ for whatever $\mathbf{t}_i$. Therefore, the problem of (5.9) is simplified to the solving of $\mathbf{t}_i$ of (5.10b) only, which has only $n - m$ columns instead of the n columns of (5.9).

From observability criteria and the form of matrix C, the matrix product on the left-hand side of (5.10b) must have m linearly dependent rows. Furthermore, this matrix

$$(A-\lambda_i I)\begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} = \begin{bmatrix} C_2 & 0 & \cdots\cdots & \cdots\cdots & 0 \\ A_{22}-\lambda_i I & C_3 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & 0 \\ \vdots & & & & C_v \\ A_{v2} & \cdots & & A_{vv} & -\lambda_i I \end{bmatrix} \tag{5.11}$$

maintains the same form of that of the right $n-m$ columns of matrix A, if A is in block-observable Hessenberg form (5.5). From the definition of this form, the C_j matrices $(j=2,\ldots,v)$ and the matrix of (5.11) are in lower echelon form. In other words, the $n-m$ linearly independent rows of matrix (5.11) are clearly indicated as the rows corresponding to the nonzero elements of matrices C_j $(j=2,\ldots,v)$. Each of the rest of m rows of matrix (5.11) can always be expressed as a linear combination of its previous and linearly independent rows in that matrix. Thus we have the following conclusion.

Conclusion 5.1

The solution $\mathbf{t}_i$ of Eq. (5.10b) has m basis vectors $\mathbf{d}_{ij}$ $(j=1,\ldots,m)$. If $(A,\ C)$ is already in block-observable Hessenberg form, then each of these m basis vectors can correspond to one of the m linearly dependent rows of matrix (5.11), each can be formed by the linear combination coefficients of the preceding and linearly independent rows of this linearly dependent row, and each can be computed by back substitution.

Example 5.4 For a Single-Output Case $(m=1)$

From (5.2),

$$(A-\lambda_i I)\begin{bmatrix} 0 \\ I_{n-1} \end{bmatrix} = \begin{bmatrix} * & 0 & \cdots\cdots & \cdots\cdots & 0 \\ x & * & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & 0 \\ \vdots & & & & * \\ x & \cdots & \cdots\cdots & \cdots\cdots & x \end{bmatrix}$$

which has only one ($m = 1$) linearly dependent row (the row without a "*" element). The solution $\mathbf{t}_i$ therefore has only one basis vector and is unique, and can be computed by back substitution.

Example 5.5 For a Multi-Output Case

In Example 5.3 ($m = 4$), for each λ_i, the corresponding solution $\mathbf{t}_i$ of (5.10b) has m ($= 4$) basis vectors as

$$\begin{array}{l}
\mathbf{d}_{i1} = [\,x \quad x \quad 0 \quad x \quad : \quad 1 \quad 0 \quad 0 \quad : \quad 0 \quad : \quad 0\,] \\
\mathbf{d}_{i2} = [\,x \quad x \quad 0 \quad x \quad : \quad 0 \quad x \quad 0 \quad : \quad x \quad : \quad 1\,] \\
\mathbf{d}_{i3} = [\,x \quad x \quad 1 \quad 0 \quad : \quad 0 \quad 0 \quad 0 \quad : \quad 0 \quad : \quad 0\,] \\
\mathbf{d}_{i4} = [\,x \quad x \quad 0 \quad x \quad : \quad 0 \quad x \quad 1 \quad : \quad 0 \quad : \quad 0\,]
\end{array}$$

Each of the above vectors $\mathbf{d}_{ij}$ has a "1" element, whose position corresponds to where the j-th row becomes linearly dependent in (5.8a) ($j = 1, \ldots, m = 4$). The "x" elements of $\mathbf{d}_{ij}$ are the linear combination coefficients of the linearly independent and preceding rows on that j-th linearly dependent row. Because each $\mathbf{d}_{ij}$ vector satisfies (5.10b), the actual solution $\mathbf{t}_i$ of (5.10b) can be an arbitrary linear combination of the $\mathbf{d}_{ij}$'s.

At the same position of each "1" element of $\mathbf{d}_{ij}$, the elements of other three basis vectors are all 0. Therefore the four basis vectors are linearly independent of each other.

From Conclusion 5.1, for multiple ($\leqslant m$) and real eigenvalues (say, $\lambda_i, i = 1, \ldots, m$), it is possible to assign their corresponding rows of T as $\mathbf{t}_i = \mathbf{d}_{ii}$ ($i = 1, \ldots, m$). This way, these multiple eigenvalues become equivalent of the distinct eigenvalues in the sense that their corresponding Jordan block in F becomes $\mathrm{diag}\{\lambda_i, i = 1, \ldots, m\}$. However, by making this assignment, there is certainly no more freedom left for solutions $\mathbf{t}_i$ ($i = 1, \ldots, m$), and hence this possible solution is not recommended for solving (4.1) and (4.3) [but is recommended for solving the dual of (4.1) in eigenstructure assignment problems of Chap. 8].

Replacing the block-observable Hessenberg form (5.8a) by its special case, the block-observable canonical form (5.8b), the four basis vectors of $\mathbf{t}_i$ of (5.10b) are

$$\begin{array}{l}
\mathbf{d}_{i1} = [\,\lambda_i \quad 0 \quad 0 \quad 0 \quad : \quad 1 \quad 0 \quad 0 \quad : \quad 0 \quad : \quad 0\,] \\
\mathbf{d}_{i2} = [\,0 \quad \lambda_i^3 \quad 0 \quad 0 \quad : \quad 0 \quad \lambda_i^2 \quad 0 \quad : \quad \lambda_i \quad : \quad 1\,] \\
\mathbf{d}_{i3} = [\,0 \quad 0 \quad 1 \quad 0 \quad : \quad 0 \quad 0 \quad 0 \quad : \quad 0 \quad : \quad 0\,] \\
\mathbf{d}_{i4} = [\,0 \quad 0 \quad 0 \quad \lambda_i \quad : \quad 0 \quad 0 \quad 1 \quad : \quad 0 \quad : \quad 0\,]
\end{array}$$

These four vectors not only are linearly independent of each other, but also have additional algebraic properties as follows.

Conclusion 5.2

From the above example, for a fixed parameter j $(j = 1, \ldots, m)$, any set of v_j of the n $\mathbf{d}_{ij}$ vectors are linearly independent, because these vectors form a matrix which equals a v_j dimensional Vandermonde matrix added with $n - v_j$ zero columns. This conclusion is valid for block-observable Hessenberg form-based vectors too, because (5.8a) and (5.8b) are similar to each other. This conclusion can also be extended to multiple eigenvalue and generalized eigenvector cases. See the more rigorous proof in Theorem 8.1.

It is also obvious from the above example that for a fixed parameter j $(j = 1, \ldots, m)$, any $v_j - 1$ of the n $\mathbf{d}_{ij}$ vectors are also linearly independent of matrix C of (5.5).

5.2.2 Eigenstructure Case B

For complex conjugate or multiple eigenvalues of F, the results of Case A can be generalized.

Letting λ_i and λ_{i+1} be $a \pm jb$, and their corresponding Jordan block be

$$F_i = \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$$

as in (1.10), the corresponding Eqs of (5.9), (5.10a), and (5.10b) become

$$\begin{bmatrix} \mathbf{t}_i \\ \mathbf{t}_{i+1} \end{bmatrix} A - F_i \begin{bmatrix} \mathbf{t}_i \\ \mathbf{t}_{i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{1}_i \\ \mathbf{1}_{i+1} \end{bmatrix} C \tag{5.12}$$

$$\begin{bmatrix} \mathbf{t}_i \\ \mathbf{t}_{i+1} \end{bmatrix} A \begin{bmatrix} I_m \\ 0 \end{bmatrix} - F_i \begin{bmatrix} \mathbf{t}_i \\ \mathbf{t}_{i+1} \end{bmatrix} \begin{bmatrix} I_m \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{1}_i \\ \mathbf{1}_{i+1} \end{bmatrix} C_1 \tag{5.13a}$$

and

$$\begin{bmatrix} \mathbf{t}_i \\ \mathbf{t}_{i+1} \end{bmatrix} A \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} - F_i \begin{bmatrix} \mathbf{t}_i \\ \mathbf{t}_{i+1} \end{bmatrix} \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} = 0 \tag{5.13b}$$

respectively.

Because in (5.13a) C_1 is of full-column rank and $\mathbf{1}_i$ and $\mathbf{1}_{i+1}$ are completely free, we need only to solve (5.13b) for $\mathbf{t}_i$ and $\mathbf{t}_{i+1}$. (5.13b) can be written as a set of linear equations:

$$[\mathbf{t}_i : \mathbf{t}_{i+1}] \left(\begin{bmatrix} A\begin{bmatrix}0\\ I_{n-m}\end{bmatrix} & \vdots & 0 \\ \cdots\cdots & \cdots\cdots & \cdots\cdots \\ 0 & \vdots & A\begin{bmatrix}0\\ I_{n-m}\end{bmatrix}\end{bmatrix} - \begin{bmatrix} a\begin{bmatrix}0\\ I_{n-m}\end{bmatrix} & \vdots & -b\begin{bmatrix}0\\ I_{n-m}\end{bmatrix} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots \\ b\begin{bmatrix}0\\ I_{n-m}\end{bmatrix} & \vdots & a\begin{bmatrix}0\\ I_{n-m}\end{bmatrix}\end{bmatrix}\right) = 0 \tag{5.13c}$$

where the two matrices in the bracket have dimension $2n \times 2(n-m)$, and can be expressed as

$$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \otimes A \begin{bmatrix}0\\ I_{n-m}\end{bmatrix} \qquad \text{and} \qquad F_i' \otimes \begin{bmatrix}0\\ \mathrm{I}_{n-m}\end{bmatrix}$$

respectively, where the operator "$\otimes$" stands for "Kronecker product."

It is not difficult to verify that like the matrix (5.10b) of Case A, the whole matrix in the bracket of (5.13c) has $2m$ linearly dependent rows. Therefore the solution $[\mathbf{t}_i : \mathbf{t}_{i+1}]$ of (5.13c) has $2m$ basis vectors $[\mathbf{d}_{ij} : \mathbf{d}_{i+1,j}] (j = 1, \ldots, 2m)$.

Example 5.6 Single-Output Case $(m = 1)$

Let matrix A of (5.2) be

$$\begin{bmatrix} x & * & 0 \\ x & x & * \\ x & x & x \end{bmatrix} \qquad (n = 3)$$

then the matrix of (5.13c) will be

$$\begin{bmatrix} * & 0 & : & 0 & 0 \\ x & * & : & b & 0 \\ x & x & : & 0 & b \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & : & * & 0 \\ -b & 0 & : & x & * \\ 0 & -b & : & x & x \end{bmatrix}$$

Clearly, this matrix has $2m$ $(= 2)$ linearly dependent rows which do not have "*" elements. Therefore the solution $[\mathbf{t}_i : \mathbf{t}_{i+1}]$ has $2m$ $(= 2)$ basis vectors of the following forms:

$$[\,\mathbf{d}_{i1} \quad : \quad \mathbf{d}_{i+1,1}\,] = [\,x \quad x \quad 1 \quad : \quad x \quad x \quad 0\,]$$

and

$$[\,\mathbf{d}_{i2} \quad : \quad \mathbf{d}_{i+1,2}\,] = [\,x \quad x \quad 0 \quad : \quad x \quad x \quad 1\,]$$

where the position of element "1" corresponds to one of the two linearly dependent rows, and "x" elements are the linear combination coefficients of all linearly independent rows. These two basis vectors can be computed separately, either by modified back substitution method or by Givens' rotational method [Tsui, 1986a].

For a multiple of q eigenvalues λ_i and their corresponding q-dimensional Jordan block

$$F_i' = \begin{bmatrix} \lambda_i & 1 & 0 & \cdots & \cdots & 0 \\ 0 & \lambda_i & 1 & \ddots & & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \lambda_i & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & \lambda_i \end{bmatrix}$$

where "$'$" stands for transpose, its corresponding (5.12) and (5.13a,b,c) are, respectively:

$$\begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_q \end{bmatrix} A - F_i \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_q \end{bmatrix} = \begin{bmatrix} \mathbf{1}_1 \\ \vdots \\ \mathbf{1}_q \end{bmatrix} C \tag{5.14}$$

$$\begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_q \end{bmatrix} A \begin{bmatrix} I_m \\ 0 \end{bmatrix} - F_i \begin{bmatrix} \mathbf{t}_i \\ \vdots \\ \mathbf{t}_q \end{bmatrix} \begin{bmatrix} I_m \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{1}_1 \\ \vdots \\ \mathbf{1}_q \end{bmatrix} C_1 \tag{5.15a}$$

$$\begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_q \end{bmatrix} A \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} - F_i \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_q \end{bmatrix} \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} = 0 \tag{5.15b}$$

and

$$[\mathbf{t}_1 \ \vdots \ \dots \ \vdots \ \mathbf{t}_q]\left(I_q \otimes A \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} - F_i' \otimes \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} \right) = 0 \tag{5.15c}$$

where $\mathbf{t}_i$ $(i = 1, \dots, q)$ are the q rows of solution matrix T corresponding to the Jordan block F_i.

Because C_1 is of full-column rank and $\mathbf{1}_i$ $(i = 1, \dots, q)$ are free in (5.15a), we need to solve (5.15b,c) only for $\mathbf{t}_i$ $(i = 1, \dots, q)$.

It is not difficult to verify that like (5.13c), the whole matrix in the bracket of (5.15c) has qm linearly dependent rows. Thus the solution $[\mathbf{t}_1 \ \vdots \ \dots \ \vdots \ \mathbf{t}_q]$ of (5.15c) has qm basis vectors $[\mathbf{d}_{1j} \ \vdots \ \dots \ \vdots \ \mathbf{d}_{qj}], j = 1, \dots, qm$.

Because of the simplicity of bidiagonal form of the Jordan block F_i, (5.15b,c) can be expressed as

$$\mathbf{t}_j(A - \lambda_i I) \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} = \mathbf{t}_{j-1} \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix}, \qquad j = 1, \dots, q, \mathbf{t}_0 = 0 \tag{5.15d}$$

Equation (5.15d) shows that all $\mathbf{t}_j$ vectors except $\mathbf{t}_1$ are computed based on its previous vector $\mathbf{t}_{j-1}$. These vectors are called "generalized" or "defective." Because the vectors $\mathbf{t}_j$ are also the left eigenvectors [see (1.10)], we also call $\mathbf{t}_j$ $(j = 2, \dots, q)$ of (5.15d) "generalized/defective eigenvectors" [Golub and Wilkinson, 1976b].

The above cases of A and B can be summarized in the following algorithm for solving (4.1) [Tsui, 1987a].

Algorithm 5.3 Computation of Solution of Matrix Equation $TA - FT = LC$

Step 1: Based on each eigenvalue of F (say, λ_i, which is distinct real, complex conjugate, or multiple [of q]), compute the m, $2m$, and qm basis vectors of the corresponding $\mathbf{t}_i, [\mathbf{t}_i : \mathbf{t}_{i+1}]$, and $[\mathbf{t}_i \ : \ \ldots \ : \ \mathbf{t}_{i+q-1}]$, according to (5.10b), (5.13c), and (5.15c), respectively.

Step 2: The row (or rows) $\mathbf{t}_i, [\mathbf{t}_i : \mathbf{t}_{i+1}]$, and $[\mathbf{t}_i \ : \ \ldots \ : \ \mathbf{t}_{i+q-1}]$ equal an arbitrary linear combination of their corresponding set of m, $2m$, and qm basis vectors, respectively $(i = 1, \ldots, n)$. There are a total of nm free linear combination coefficients.

Step 3: After all $\mathbf{t}_i$ rows and the corresponding matrix T are fully determined in Step 2, satisfy the left m columns of $TA - FT = LC$ [or (5.10a), (5.13a), and (5.15a)] by solving

$$(TA - FT)\begin{bmatrix} I_m \\ 0 \end{bmatrix} = LC_1 \tag{5.16}$$

The solution L is unique because C_1 has m linearly independent columns.

Conclusion 5.3

The above Algorithm 5.3 computes (F, T, L) which satisfies (4.1). It is clear that the first two steps of the algorithm satisfy the right $n - m$ columns of (4.1), and Step 3 satisfies the left m columns of (4.1). This solution does not assume any restrictions and is therefore completely general. The complete remaining freedom of (4.1) is also expressed explicitly (as the linear combination coefficients) in Step 2.

Let us analyze the computational reliability and efficiency of this algorithm.

Because the initial step of an algorithm affects the computation reliability of that algorithm most, and because most of the computation of Algorithm 5.3 concerns Step 1, the analysis will concentrate on this step only.

This step can be carried out by back substitution (see Sec. A.2 of Appendix A), which is itself numerically stable [Wilkinson, 1965]. However, this operation requires repeated divisions by those "*" nonzero elements of the Hessenberg form matrix (5.5). Therefore this step can be ill conditioned if these nonzero elements are not large enough in magnitude.

According to the Householder method (see Sec. A.2 of Appendix A), these nonzero elements (computed at Step 2 of Algorithm 5.2) equal the norm of the corresponding row vector. This step also consists of the determination of whether that norm is zero or nonzero. Therefore, to improve the condition of Step 1 of Algorithm 5.3, it is plausible to admit only the large enough vector norms as nonzero. From the description of Example 1.5, each of these nonzero elements is the only link between one of the plant system states to system output. Thus admitting only the large enough elements as nonzero implies admitting only the strongly observable states as observable states.

However, reducing the dimension of a system's observable part also implies the reduction of system information. This tradeoff of accuracy and solution magnitude is studied in depth in Lawson and Hanson [1974], Golub et al. [1976a]. To best handle this tradeoff, the singular value decomposition (SVD) method can be used to replace the Householder method in Step 2 of Algorithm 5.2 [Van Dooren, 1981; Patel, 1981]. However, the SVD method cannot determine at that step which row among the m_{j-1} rows of matrix $\overline{C}_j$ is linearly dependent or independent, and thus cannot determine the observability index, which is the analytic information about multi-output system and is as important as the system order of single-output systems. In addition, the SVD method cannot result in echelon form matrix C_j—the form which made the simple back substitution operation of Step 1 of Algorithm 5.3 possible.

The distinct advantage of the computational efficiency of Step 1 of Algorithm 5.3 is that this computation can be carried out in *complete parallel*. This advantage is *uniquely* enabled by the distinct feature that all basis vectors $\mathbf{d}_{ij}$ are completely decoupled for all j $(j = 1, \ldots, m)$ and for all i $(i = 1, \ldots, n)$ as long as the λ_i's are in different Jordan blocks of F. In other words, the computation of $\mathbf{d}_{ij}$ does not depend on the information of other $\mathbf{d}_{ij}$'s. Only the $\mathbf{d}_{ij}$'s corresponding to the same Jordan block and the same j are coupled [see (5.13c) and (5.15c)]. In addition, the back substitution operation is itself very simple and efficient (see Sec. A.2 of Appendix A).

The basic reason for the good computational properties of Algorithm 5.3 is the Jordan form of matrix F. It should be noticed that simplicity and decoupling are the fundamental features and advantages of eigenstructure decomposition. This is the reason that the eigenstructure decomposition (or Jordan form) is computed from a given matrix in the first place. In the particular problem of solving (4.1), the eigenvalues are *given* and are *unnecessary* to be computed. Therefore it is certainly plausible to set matrix F in Jordan form—the form that is much sought after in other problems.

Conclusion 5.4

The computation of Algorithm 5.3 is very reliable and very efficient, as compared with other algorithms for solving (4.1).

The much more important advantage of the solution of Algorithm 5.3 concerns its analytical aspects.

Equation (4.1) is not only the most fundamental equation of observer feedback compensator (3.16) design (see Chaps 3 and 4), but also the most fundamental equation of state/generalized state feedback design. The dual of (4.1) is

$$AV - V\Lambda = B\check{K}, \tag{5.17}$$

which implies $A - BK = V\Lambda V^{-1}$, where $K = \check{K}V^{-1}$ is the state feedback control gain, and V and Λ are the right eigenvector matrix and the Jordan form matrix of the state feedback system dynamic matrix $A - BK$, respectively.

Because of these reasons, if Lyapunov/Sylvester equations

$$AV - VA' = B \quad / \quad AV - V\Lambda = B \tag{5.18}$$

are considered fundamental in system analysis, and if the algebraic Riccati equation is considered fundamental in quadratic optimal control system design, then Eqs (4.1) and (5.17) should be considered the most fundamental equations in state space control system design.

However, the really general solution of (4.1), with really fully usable remaining freedom and with fully decoupled properties, was not derived until 1985 [Tsui, 1987a, 1993a]. For example, the solution of the Sylvester equation (5.18) has generally been used as the substitute of the general solution of (5.17) [Tsui, 1986c]. Because (5.18) lacks the free parameter $\check{K}$ at its right-hand side as compared with (5.17) [or lacks parameter L of (4.1)], it cannot be simplified to the form of (5.10b), (5.13c), or (5.15c). Thus the existence of solution of (5.18) is questionable when A and Λ share common eigenvalues [Gantmacher, 1959; Chen, 1984; Friedland, 1986]. Such a solution is certainly not a general solution of (4.1) or (5.17).

From Conclusion 5.3, the general solution of (4.1) or (5.17) has been derived, with explicitly and fully expressed remaining freedom and with completely decoupled rows corresponding to the different Jordan blocks of F. Such a solution to such a fundamental equation of design will certainly have great impact on state space control system design and on the practical value of state space control theory. In fact, as will be shown in the rest of this book, this solution has uniquely enabled the dynamic output feedback compensator design [Tsui, 1992, 1993b] (Sec. 6.1), the systematic minimal order observer design [Tsui, 1985] (Chap. 7), the systematic eigenvalue assignment [Tsui, 1999a] (Sec. 8.1) and eigenvector assignment [Kautsky et al., 1985; Tsui, 1986a] (Sec. 8.2), and the robust failure detector design [Tsui, 1989] (Sec. 10.1).

Figure 5.1 outlines the sequential relationship of these design results.

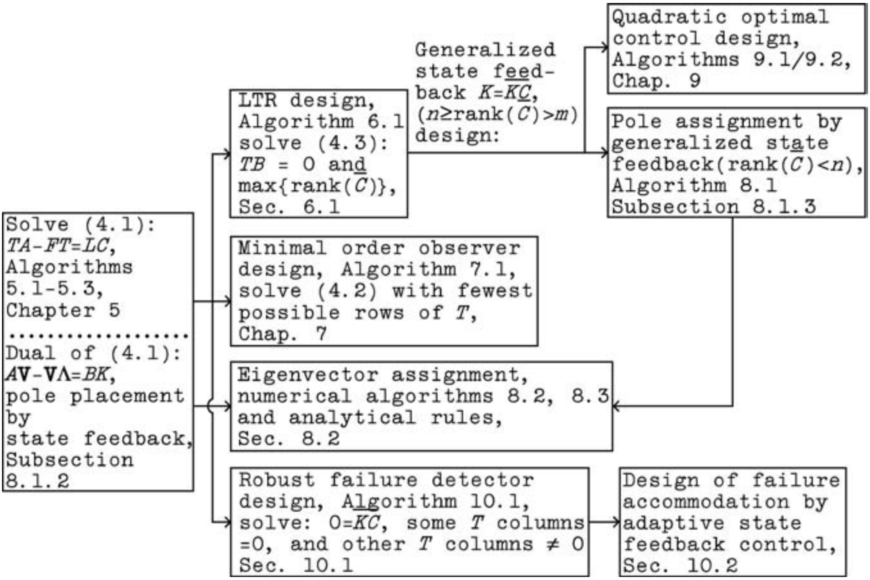
EXERCISES

5.1 Repeat the computation of similarity transformation to block-observable Hessenberg form of Example 6.2, according to Algorithm 5.2 (also Algorithm A.1 for QR decomposition).

5.2 Repeat 5.1 for Example 8.7 (dual version).

5.3 Repeat the computation of satisfying (4.1) for Examples 6.1, 6.2, 6.3, 7.3 and 8.1, 8.2, 8.3, 8.4 (Step 1), according to Algorithm 5.3 (first two steps mainly). Verify (4.1) for these results.

5.4 Partitioning the state of system (5.7) as $[\mathbf{x}_o(t)' : \bar{\mathbf{x}}_o(t)']'$, the system's block diagram can be depicted as in Fig. 5.2 which shows that



Quadratic optimal control design, Algorithms 9.1/9.2, Chap. 9
Generalized state feed-back $K=\underline{KC}$, ($n \geq$rank(C)>m) design:
LTR design, Algorithm 6.1 solve (4.3): TB = 0 and max{rank($\underline{C}$)}, Sec. 6.1
Pole assignment by generalized state feedback(rank(C)<n), Algorithm 8.1 Subsection 8.1.3
Solve (4.1): $TA-FT=LC$, Algorithms 5.1-5.3, Chapter 5
Dual of (4.1): $AV-V\Lambda=BK$, pole placement by state feedback, Subsection 8.1.2
Minimal order observer design, Algorithm 7.1, solve (4.2) with fewest possible rows of T, Chap. 7
Eigenvector assignment, numerical algorithms 8.2, 8.3 and analytical rules, Sec. 8.2
Robust failure detector design, Algorithm 10.1, solve: 0=KC, some T columns =0, and other T columns ≠ 0 Sec. 10.1
Design of failure accommodation by adaptive state feedback control, Sec. 10.2

Figure 5.1 Sequence of design algorithms of this book.

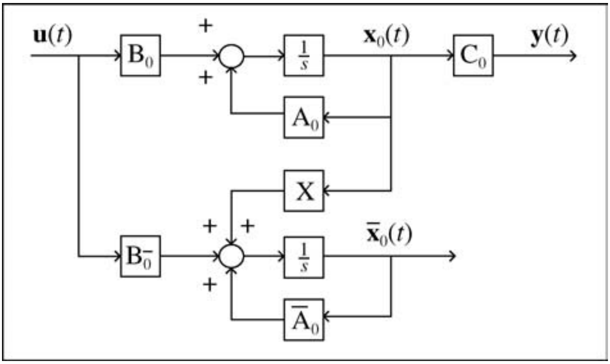


Figure 5.2 Block diagram of systems with observable and unobservable parts.

(A_o, B_o, C_o) is observable, while the other part of the system $(\overline{A}_o, \overline{B}_o, 0)$ is not. Repeat this proof for its dual case (controllable Hessenberg form).

5.5 Compute the solution $(T \triangleq [t_1 \; t_2], L)$ which satisfies the matrix equation (4.1) $(TA - FT = LC)$, where [Chen, 1993]

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \qquad C = [1 \quad 0]$$

and

$$F = -4 \text{ and } -1, \text{ respectively}$$

Answer: For $F = -4 : T[1 \quad 3]' = 0 \Rightarrow T = [-3t_2 \quad t_2]$(arbitrary $t_2 \neq 0$), then, $L = T[4 \quad 0]' = -12t_2$.

For $F = -1 : T[1 \quad 0]' = 0 \Rightarrow T = [0 \quad t_2]$(arbitrary $t_2 \neq 0$), then, $L = T[1 \quad 0]' = 0$.

6

Observer (Dynamic Part) Design for Robustness Realization

Step 2 of Algorithm 5.3 revealed the remaining freedom of (4.1). This freedom will be fully used for the various design applications listed at the end of Chap. 5.

This chapter describes the first of such applications—observer design for the guaranteed full realization of the robustness properties of state feedback control. Failure to realize the robustness properties of this control is perhaps the drawback that has limited the practical applications of state space control theory. This chapter will demonstrate for the first time, that with the full use of the remaining freedom of (4.1), this drawback can be effectively overcome for most open loop system conditions.

The design of this chapter will fully determine the dynamic part of the observer, which can also be considered as a feedback compensator.

This chapter consists of four sections.

Section 6.1 presents the design algorithm that uses the remaining freedom of (4.1) to best satisfy equation $TB = 0$ (4.3). As described in Chap. 3, $TB = 0$ is the key requirement of realizing the robustness properties of state feedback control [Tsui, 2000].

Section 6.2 analyzes the generality of the above solution of (4.1) and (4.3), and illustrates this design algorithm with six numerical examples.

Section 6.3 demonstrates a theoretical significance of this design algorithm—the complete unification of exact LTR state observer feedback system and the static output feedback system.

Section 6.4 describes the adjustment of observer order, which is completely adjustable under the design algorithm of this book. The higher observer order implies a less constrained and therefore a more powerful generalized state feedback control, while the lower observer order implies an easier realization of robustness properties of this control.

6.1 SOLUTION OF MATRIX EQUATION $TB = 0$

Let us first summarize the results at Step 2 of Algorithm 5.3. For distinct and real eigenvalue λ_i,

$$\mathbf{t}_i = \mathbf{c}_i D_i \tag{6.1a}$$

For complex conjugate λ_i and λ_{i+1},

$$[\mathbf{t}_i : \mathbf{t}_{i+1}] = [\mathbf{c}_i : \mathbf{c}_{i+1}][D_i : D_{i+1}] \tag{6.1b}$$

For multiple of q eigenvalues λ_j $(j = i, \ldots, i+q-1)$,

$$[\mathbf{t}_i : \ldots : \mathbf{t}_{i+q-1}] = [\mathbf{c}_i : \ldots : \mathbf{c}_{i+q-1}][D_i : \ldots : D_{i+q-1}] \tag{6.1c}$$

The dimensions of each row vector $\mathbf{t}_i$ and $\mathbf{c}_i$ $(i = 1, \ldots, n-m)$ are n and m, respectively.

Algorithm 6.1 Solve $TB = 0$ (4.3)[*Tsui*, 1992, 1993b]

Step 1: Substitute (6.1) into (4.3), we have

$$\mathbf{c}_i[D_iB] = 0 \tag{6.2a}$$

$$[\mathbf{c}_i : \mathbf{c}_{i+1}][D_iB : D_{i+1}B] = 0 \tag{6.2b}$$

and

$$[\mathbf{c}_i : \ldots : \mathbf{c}_{i+q-1}][D_iB : \ldots : D_{i+q-1}B] = 0 \tag{6.2c}$$

respectively.

Step 2: Compute the solution $\mathbf{c}_i$ $(i = 1, \ldots, n - m)$ of (6.2).

Equation (6.2) is only a set of linear equations (see Appendix A). Nonetheless, there are two special cases of (6.2) which will be treated separately in the following. To simplify the description, only the distinct and real eigenvalue case (6.2a) will be described.

Case A

If the exact nonzero solution of (6.2a) does not exist (this usually happens when $m < p + 1$), then compute the least square solution of (6.2a):

$$\mathbf{c}_i = \mathbf{u}'_m \tag{6.3}$$

where $\mathbf{u}_m$ is the m-th column of matrix U, and where $U\Sigma V' = D_iB$ is the singular value decomposition of D_iB (with nonzero singular values $\sigma_i > 0$, $i = 1, \ldots, m$) of (A.21). The corresponding right-hand side of (6.2a) will be $\sigma_m\mathbf{v}_m$, where $\mathbf{v}_m$ is the m-th row of matrix V' of (A.21) (see Example A.6).

Because the solution of case A implies that $TB \neq 0$, the corresponding observer (3.16) cannot be considered as a dynamic output feedback compensator (4.10), even though this observer approximates the dynamic output feedback compensator requirement $(TB = 0)$ in least-square sense.

Case B

If the exact solution of (6.2a) is not unique (this usually happens when $m > p + 1$), then the remaining freedom of (6.2a) [and (4.1)] exists. This freedom will be fully used to maximize the angles between the rows of matrix $\overline{C} = [T' : C']'$ by the following three substeps. The purpose of this operation is to best strengthen the state and generalized state feedback control $K\mathbf{x}(t) = \overline{KC}\mathbf{x}(t)$ which is eventually implemented by this observer.

Step 2a: Compute all $m - p$ possible and linearly independent solutions $\mathbf{c}_{ij}$ of (6.2a) such that

$$\mathbf{c}_{ij}[D_i B] = 0, \; j = 1, \ldots, m - p \tag{6.4}$$

Step 2b: Compute matrix

$$\overline{D}_i = \begin{bmatrix} \mathbf{c}_{i1} & D_i \\ \vdots & \\ \mathbf{c}_{i,m-p} & D_i \end{bmatrix} \tag{6.5}$$

Step 2c: Compute the $m - p$ dimensional row vector $\mathbf{c}_i$ $(i = 1, \ldots, n - m)$ such that the angles between the rows

$$\mathbf{t}_i \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} = \mathbf{c}_i \overline{D}_i \begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} \tag{6.6}$$

are maximized (as close to $\pm 90^\circ$ as possible). The explicit algorithms of Substep 2c will be described as Algorithms 8.2 and 8.3 in Chap. 8. Because of the special form of matrix C in (5.5), Substep 2c implies the maximization of the angles between the rows of matrix $[T' : C']'$. In addition, maximizing row vector angles also implies maximizing the row rank of the same matrix. The second maximization is much easier than the first (only nonzero angles between the vectors are required), and is guaranteed to be achieved by Algorithms 8.2 and 8.3 even though the first maximization may not be.

It is obvious that $TB = 0$ is satisfied or best satisfied by Algorithm 6.1 [after (4.1) is satisfied].

6.2 ANALYSIS AND EXAMPLES OF THIS DESIGN SOLUTION

Design algorithm 5.3 [for (4.1)] and design algorithm 6.1 [for (4.3)] completely determine the dynamic part of observer feedback compensator and essentially define the new design approach of this book. This design is analyzed theoretically and is illustrated by six numerical examples in this section.

Conclusion 6.1

A sufficient condition to satisfy (4.1) and (4.3) exactly is $m > p$. It is obvious that Algorithm 5.3 satisfies (4.1) for all observable plant systems (A, B, C), while (4.3) [or (6.2)] can always be satisfied by the remaining freedom of (4.1) (the $\mathbf{c}_i$ parameters) if $m > p$.

Another sufficient condition to satisfy (4.1) and (4.3) exactly is that the plant system has at least one stable transmission zero. This is because from the property of transmission zeros (say z_i) described after Definition 1.5, there exists at least one vector (say, $[\mathbf{t}_i : \mathbf{l}_i]$) such that

$$[\mathbf{t}_i : \mathbf{l}_i]S = [\mathbf{t}_i : \mathbf{l}_i]\begin{bmatrix} A - z_i I & : & B \\ -C & : & 0 \end{bmatrix} = 0 \tag{6.7}$$

if $m \not> p$. Because z_i is matched by an eigenvalue λ_i of F (see the beginning of Sec. 5.2), the comparison between (4.1) and the left n columns of (6.7) and the comparison between (4.3) and the right p columns of (6.7) indicate that $\mathbf{t}_i$ and $\mathbf{l}_i$ of (6.7) are the i-th row of T and L of (4.1) and (4.3), respectively. In other words, (4.1) and (4.3) are automatically satisfied together if z_i is matched by λ_i.

It should be noticed that the number of rows of solution (F, T, L) of (4.1) is freely adjustable and can be as low as one. Therefore the existence of at least one stable transmission zero z_i implies the existence of solution of (4.1) and (4.3).

A sufficient condition for $m \not> p$ is also a sufficient condition for $m > p$, because the former case is more difficult (has less output measurement information but more controls to realize) than the latter case, as proved by the first part of this conclusion. Definition 1.5 also implies that the existence of stable transmission zeros is also a necessary condition to satisfy (4.1) and (4.3) exactly if $m \not> p$.

Conclusion 6.2

It is obvious that Algorithms 5.3 and 6.1 fully used the entire design freedom of observer dynamic part (F, T, L) (after the eigenvalues of F are determined) to satisfy (4.1) and (4.3) and to maximize the angles between the rows of matrix $\overline{C} = [T' : C']'$ [see Conclusion 5.3 and Algorithm 6.1 (Case B)].

Conclusion 6.3

If the plant system (A, B, C) either has $n - m$ stable transmission zeros or satisfies (1) minimum-phase; (2) $\text{rank}(CB) = p$; and (3) $m \geqslant p$, then the resulting matrix $\overline{C} = [T' : C']'$ of Algorithms 5.3 and 6.1 is nonsingular. In other words, Algorithms 5.3 and 6.1 will result in an exact LTR state observer if the plant system satisfies the above conditions.

Proof

The proof is divided into two parts, A and B.

Part A: The plant system has $n - m$ stable transmission zeros

From Conclusion 6.1, for general plant system (A, B, C) with $m \ngtr p$, there exists an additional linearly independent row of solution (F, T, L) of (4.1) and (4.3) if and only if there exists an additional plant system stable transmission zero. From Conclusion 5.2, the $n - m$ rows of T corresponding to the $n - m$ stable transmission zeros can always be made linearly independent of each other and of the rows of matrix C. Thus the necessary and sufficient condition for the plant system $G(s)$ with $m \ngtr p$, to have an exact solution of (4.1), (4.3) and a nonsingular matrix $\overline{C} = [T' : C']'$, is that $G(s)$ has $n - m$ stable transmission zeros.

Similar to the last argument of Conclusion 6.1, the sufficient condition for $m \ngtr p$ is also a sufficient condition for $m > p$.

Part B: The plant system satisfies (1) *minimum-phase*, (2) *rank*$(CB) = p$, *and* (3) $m \geqslant p$

First, because $m = p$ and $\text{rank}(CB) = p$ guarantee $n - m$ plant system transmission zeros [Davison and Wang, 1974], the additional condition of minimum-phase guarantees $n - m$ stable plant system transmission zeros. Thus the proof of Part A of this conclusion can be used to prove Part B for the case of $m = p$.

For the case of $m > p$ of Part B, the proof is indirect via the proof that the above three conditions are sufficient conditions for the existence of unknown input observers or exact LTR state observers which satisfy (4.1), (4.3), and rank $(\overline{C}) = n$ (see Sec. 4.3). Because Conclusion 6.2 shows that Algorithms 5.3 and 6.1 *fully* used the remaining observer dynamic part design freedom to satisfy (4.1), (4.3) and maximized rank of $\overline{C}$ after the eigenvalues of F are assigned, and because the eigenvalues of F and the poles of unknown input observers are similarly assigned, matrix $\overline{C}$ of Algorithms 5.3 and 6.1 will have the maximum rank n and will be nonsingular if the unknown input observer exists.

There have been a number of such proofs in the literature [Kudva et al., 1980; Hou and Muller, 1992; Syrmos, 1993b]. It seems that the proof in Hou and Muller [1992] is most complete and explicit. This proof is presented in the following, with minor revision.

Let a nonsingular matrix

$$Q = [B : \overline{B}] \tag{6.8}$$

where $\overline{B}$ is an arbitrary matrix which makes Q nonsingular. Then make a similarity transformation on the plant system (A, B, C):

$$\bar{\mathbf{x}}(t) = Q^{-1}\mathbf{x}(t) \underset{=}{\triangle} [\bar{\mathbf{x}}_1(t)' : \bar{\mathbf{x}}_2(t)']' \tag{6.9}$$

and

$$(Q^{-1}AQ, Q^{-1}B, CQ) \underset{=}{\triangle} \left(\begin{bmatrix} A_{11} & : & A_{12} \\ A_{21} & : & A_{22} \end{bmatrix}, \begin{bmatrix} I_p \\ 0 \end{bmatrix}, [CB : C\overline{B}] \right) \tag{6.10}$$

From (6.9) and (6.10),

$$\dot{\bar{\mathbf{x}}}_2(t) = A_{21}\bar{\mathbf{x}}_1(t) + A_{22}\bar{\mathbf{x}}_2(t) \tag{6.11a}$$

$$\mathbf{y}(t) = CB\bar{\mathbf{x}}_1(t) + C\overline{B}\bar{\mathbf{x}}_2(t) \tag{6.11b}$$

Because $m > p$ and Rank $(CB) = p$, all columns of CB are linearly independent. Hence we can set a nonsingular matrix

$$P = [CB : \overline{CB}]$$

where $\overline{CB}$ is an arbitrary matrix which makes matrix P nonsingular. Multiplying P^{-1} on the left-hand side of (6.11b) we have

$$P^{-1}\mathbf{y}(t) \underset{=}{\triangle} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \mathbf{y}(t) = \begin{bmatrix} I_p & : & P_1 C\overline{B} \\ 0 & : & P_2 C\overline{B} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}_1(t) \\ \bar{\mathbf{x}}_2(t) \end{bmatrix} \tag{6.12}$$

From the first p rows of (6.12),

$$\bar{\mathbf{x}}_1(t) = P_1[\mathbf{y}(t) - C\overline{B}\bar{\mathbf{x}}_2(t)] \tag{6.13}$$

Substituting (6.13) and (6.12) into (6.11), we have the following system of order $n - p$

$$\begin{aligned}\dot{\overline{\mathbf{x}}}_2(t) &= (A_{22} - A_{21}P_1C\overline{B})\overline{\mathbf{x}}_2(t) + A_{21}P_1\mathbf{y}(t) \qquad (6.14\text{a})\\ &\underset{=}{\triangle}\tilde{A}\overline{\mathbf{x}}_2(t) + \tilde{B}\mathbf{y}(t)\\ \overline{\mathbf{y}}(t)&\underset{=}{\triangle}P_2\mathbf{y}(t) = P_2C\overline{B}\overline{\mathbf{x}}_2(t)\underset{=}{\triangle}\tilde{C}\overline{\mathbf{x}}_2(t) \qquad (6.14\text{b})\end{aligned}$$

Because system (6.14) does not involve the original plant system input $\mathbf{u}(t)$, its corresponding state observer is an unknown input observer. In addition, if $\overline{\mathbf{x}}_2(t)$ is estimated by this observer, then from (6.13) $\overline{\mathbf{x}}_1(t)$ can also be estimated. Thus the sufficient condition for the existence of unknown input observer of the original plant system is equivalent to the detectability of system (6.14), plus the Rank$(CB) = p$ and $m > p$ conditions which made the system formulation (6.14) possible.

Because for the system (6.10),

$$\text{Rank}\begin{bmatrix} sI_p - A_{11} & -A_{12} & : & I_p \\ -A_{21} & sI_{n-p} - A_{22} & : & 0 \\ CB & C\overline{B} & : & 0 \end{bmatrix} \qquad (6.15\text{a})$$

$$= p + \text{Rank}\begin{bmatrix} -A_{21} & sI_{n-p} - A_{22} \\ CB & C\overline{B} \end{bmatrix}$$

$$= p + \text{Rank}\left(\begin{bmatrix} I_{n-p} & 0 \\ 0 & P^{-1} \end{bmatrix}\begin{bmatrix} -A_{21} & sI_{n-p} - A_{22} \\ CB & C\overline{B} \end{bmatrix}\begin{bmatrix} -P_1C\overline{B} & I_p \\ I_{n-p} & 0 \end{bmatrix}\right)$$

$$= p + \text{Rank}\begin{bmatrix} sI_{n-p} - \tilde{A} & -A_{21} \\ 0 & I_p \\ \tilde{C} & 0 \end{bmatrix}\begin{matrix} \}n-p \\ \}p \\ \}m-p \end{matrix}$$

$$= 2p + \text{Rank}\begin{bmatrix} sI_{n-p} - \tilde{A} \\ \tilde{C} \end{bmatrix} \qquad (6.15\text{b})$$

A comparison of (6.15a) and (6.15b) shows that the transmission zeros of system (6.10) equal the poles of unobservable part of system (6.14). Therefore, the necessary and sufficient condition for system (6.14) to be detectable is that all transmission zeros of plant system (6.10) are stable [or that (6.10) is minimum-phase]. Thus the proof.

Conclusion 6.4

The conditions that a plant system is minimum-phase and that $\text{Rank}(CB) = p$ are necessary for the plant system to have exact LTR state observer.

Proof

The proof of Conclusion 6.3 shows the condition that all plant system transmission zeros are stable (minimum-phase) is a necessary condition for the existence of unknown input observers.

For matrix $\overline{C} = [T' : C']'$ be nonsingular, CB must have full-column rank if $TB = 0$ (see Example A.7).

The conditions of Conclusion 6.3 and 6.4 are summarized in the following Table 6.1, which shows that the condition of $n - m$ stable transmission zeros is stronger than the condition of minimum-phase and $\text{rank}(CB) = p$. The two conditions are equivalent (both necessary and sufficient) for the case $m = p$, but the former is *not* a necessary condition, while the latter is if $m > p$; and the latter is *not* a sufficient condition, while the former is if $m < p$. Thus between the two conditions themselves, the former is a sufficient condition of the latter, while the latter is only a necessary condition (but *not* a sufficient condition) of the former. Hence the former condition is even more strict than the latter. This result conforms with the existing properties about transmission zeros [Davison and Wang, 1974].

Table 6.1 also shows that in any case the condition of minimum-phase and $\text{Rank}(CB) = p$ is a necessary condition for the existence of exact LTR state observers. It is difficult to require that *all* existing transmission zeros be

Table 6.1 Necessary and Sufficient Conditions for the Existence of a Dynamic Output Feedback Compensator Which Implements Arbitrarily Given State Feedback Control

Conditions	$m < p$	$m = p$	$m > p$
Has $n - m$ stable transmission zeros	Necessary and sufficient	Necessary and sufficient	Sufficient
Minimum-phase and CB full-column rank	Necessary	Necessary and sufficient	Necessary and sufficient

stable (see Exercises 4.2 and 4.6). In SISO systems, $\text{rank}(CB) = p$ (or $CB \neq 0$) implies the existence of $n - m$ zeros. In MIMO systems, this condition is also closely related to the number of zeros [see Davison and Wang, 1974] and is unsatisfied by many practical systems such as airborne systems. Thus the existing result of LTR is very severely limited and is in fact *invalid* for most plant systems.

From Conclusion 6.1, the new design approach of Algorithms 5.3 and 6.1 requires either the existence of at least one stable transmission zero or $m > p$. Because almost all plants with $m = p$ have $n - m$ transmission zeros [Davison and Wang, 1974], $m = p$ can also be the sufficient condition of (4.1) and (4.3) for most cases (see Exercises 4.3 and 4.7). Thus the restrictions of minimum-phase and $\text{rank}(CB) = p$ of Conclusion 6.3 are almost *completely* eliminated. Thus our new design is valid for *most* practical systems. It is also common to have $m > p$ because it is generally much easier to add measurements (or m) to a system than to add controls (or p) to a system. This *significant generalization* of the critical robust control design is possible because the new design approach of this book avoids the realization of separately designed and arbitrarily given state feedback control.

Example 6.1

This is an example of four plant systems which share a common system matrix pair (A, C)

$$A = \left[\begin{array}{ccccccccc}
x & x & x & : & 1 & 0 & 0 & : & 0 \\
x & x & x & : & 0 & 1 & 0 & : & 0 \\
x & x & x & : & 0 & 0 & 1 & : & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
x & x & x & : & 0 & 0 & 0 & : & 1 \\
x & x & x & : & 0 & 0 & 0 & : & 0 \\
x & x & x & : & 0 & 0 & 0 & : & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
x & x & x & : & 0 & 0 & 0 & : & 0
\end{array}\right]$$

and

$$C = \left[\begin{array}{ccccccccc}
1 & 0 & 0 & : & 0 & 0 & 0 & : & 0 \\
x & 1 & 0 & : & 0 & 0 & 0 & : & 0 \\
x & x & 1 & : & 0 & 0 & 0 & : & 0
\end{array}\right] \tag{6.16}$$

where “x”’s are arbitrary elements. Thus this example is very general.

The matrix pair (A, C) of (6.16) is in observable canonical form (1.16) or (5.8b). The four plant systems are distinguished by their respective B matrices:

$$B_1 = \left[\begin{array}{cc} 1 & 0 \\ 1 & 1 \\ 1 & 0 \\ \hdashline -1 & 1 \\ 1 & -1 \\ 1 & 2 \\ \hdashline -2 & -2 \end{array}\right] \qquad B_2 = \left[\begin{array}{cc} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ \hdashline 1 & 0 \\ 2 & 2 \\ 3 & 1 \\ \hdashline 1 & 1 \end{array}\right] \qquad B_3 = \left[\begin{array}{cc} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \hdashline -1 & 1 \\ 1 & 2 \\ 1 & 1 \\ \hdashline -2 & -2 \end{array}\right]$$

and

$$B_4 = \left[\begin{array}{cc} 1 & 0 \\ 1 & 1 \\ 1 & 0 \\ \hdashline -1 & 1 \\ -2 & -1 \\ -2 & 2 \\ \hdashline -2 & -1 \end{array}\right]$$

Using the method of Example 1.7 we can derive directly the polynomial matrix fraction description of the corresponding transfer function $G(s) = D^{-1}(s)N(s)$, where polynomial matrix $D(s)$ is common for all four systems, and the four different $N(s)$ polynomial matrices are:

$$N_1(s) = \left[\begin{array}{cccc} (s+1) & (s-2) & : & (s-2) \\ (s+1) & & : & (s-1) \\ (s+1) & & : & 2 \end{array}\right]$$

$$N_2(s) = \left[\begin{array}{ccc} (s+1) & : & 1 \\ 2 & : & (s+2) \\ (s+3) & : & 1 \end{array}\right]$$

$$N_3(s) = \begin{bmatrix} (s+1) & (s-2) & : & (s-2) \\ (s+1) & & : & 2 \\ (s+1) & & : & 1 \end{bmatrix}$$

and

$$N_4(s) = \begin{bmatrix} (s-2) & (s+1) & : & (s-1) \\ (s-2) & & : & (s-1) \\ (s-2) & & : & 2 \end{bmatrix}$$

The four $N(s)$ matrices reveal that while all four systems have $m = 3 > 2 = p$, only the first and the third systems have one stable transmission zero -1, and only the fourth system has an unstable transmission zero 2.

Thus the dynamic matrix F of the four corresponding dynamic output feedback compensators can be commonly set as

$$F = \text{diag}\,\{-1,\ -2,\ -3,\ -4\}$$

which includes all possible stable transmission zeros of the four plant systems.

Because Step 1 of Algorithm 5.3 is based on matrices (A, C, F) which are common for the four plant systems, the result of this step is also common for the four systems. The following four basis vector matrices for the four eigenvalues of F are computed according to (5.10b):

$$D_1 = \begin{bmatrix} 1 & 0 & 0 & : & -1 & 0 & 0 & : & 1 \\ 0 & -1 & 0 & : & 0 & 1 & 0 & : & 0 \\ 0 & 0 & -1 & : & 0 & 0 & 1 & : & 0 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 4 & 0 & 0 & : & -2 & 0 & 0 & : & 1 \\ 0 & -2 & 0 & : & 0 & 1 & 0 & : & 0 \\ 0 & 0 & -2 & : & 0 & 0 & 1 & : & 0 \end{bmatrix}$$

$$D_3 = \begin{bmatrix} 9 & 0 & 0 & : & -3 & 0 & 0 & : & 1 \\ 0 & -3 & 0 & : & 0 & 1 & 0 & : & 0 \\ 0 & 0 & -3 & : & 0 & 0 & 1 & : & 0 \end{bmatrix}$$

and

$$D_4 = \begin{bmatrix} 16 & 0 & 0 & : & -4 & 0 & 0 & : & 1 \\ 0 & -4 & 0 & : & 0 & 1 & 0 & : & 0 \\ 0 & 0 & -4 & : & 0 & 0 & 1 & : & 0 \end{bmatrix}$$

The result of Step 2 of Algorithm 5.3 (or Algorithm 6.1) is computed according to (6.2a):

$$\begin{aligned} T_1 &= \begin{bmatrix} [0 & 1 & 1]D_1 \\ [1 & 4/5 & 16/5]D_2 \\ [1 & 5/6 & 25/6]D_3 \\ [1 & 6/7 & 36/7]D_4 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -1 & -1 & 0 & 1 & 1 & 0 \\ 4 & -8/5 & -32/5 & -2 & 4/5 & 16/5 & 1 \\ 9 & -15/6 & -75/6 & -3 & 5/6 & 25/6 & 1 \\ 16 & -24/7 & -144/7 & -4 & 6/7 & 36/7 & 1 \end{bmatrix} \\ T_2 &= \begin{bmatrix} [0 & 1 & -1]D_1 \\ [1 & 1 & -1]D_2 \\ [1 & 1 & 0]D_3 \\ [0 & 1 & 2]D_4 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -1 & 1 & 0 & 1 & -1 & 0 \\ 4 & -2 & 2 & -2 & 1 & -1 & 1 \\ 9 & -3 & 0 & -3 & 1 & 0 & 1 \\ 0 & -4 & -8 & 0 & 1 & 2 & 0 \end{bmatrix} \\ T_3 &= \begin{bmatrix} [0 & 1 & -2]D_1 \\ [1 & 0 & 4]D_2 \\ [1 & 0 & 5]D_3 \\ [1 & 0 & 6]D_4 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -1 & 2 & 0 & 1 & -2 & 0 \\ 4 & 0 & -8 & -2 & 0 & 4 & 1 \\ 9 & 0 & -15 & -3 & 0 & 5 & 1 \\ 16 & 0 & -24 & -4 & 0 & 6 & 1 \end{bmatrix} \end{aligned}$$

and

$$T_4 = \begin{bmatrix} [2 & -1 & 1]D_1 \\ [1 & -1/5 & 6/5]D_2 \\ [1 & 0 & 2]D_3 \\ [1 & 1/7 & 20/7]D_4 \end{bmatrix}$$
$$= \begin{bmatrix} 2 & 1 & -1 & -2 & -1 & 1 & 2 \\ 4 & 2/5 & -12/5 & -2 & -1/5 & 6/5 & 1 \\ 9 & 0 & -6 & -3 & 0 & 2 & 1 \\ 16 & -4/7 & -80/7 & -4 & 1/7 & 20/7 & 1 \end{bmatrix}$$

It can be easily verified that the above four matrices satisfy (4.1) [the right $n - m$ $(= 4)$ columns] and (4.3) $(T_i B_i = 0,\ i = 1, \ldots, 4)$. Because the left m $(= 3)$ columns of (4.1) can always be satisfied by matrix L as shown in (5.16), we consider the above four matrices the exact solution of (4.1) and (4.3). The matrix triples $(F,\ T_i,\ L_i)$ $(i = 1, \ldots, 4)$ fully determine the dynamic part (4.10a) of the four dynamic output feedback compensators. This result conforms with Conclusion 6.1 (the first part).

Let us now analyze the design of output part (4.10b) of these four dynamic output feedback compensators. Because the matrices $\overline{C}_i = [T_i' : C']'$ $(i = 1,\ 2)$ are nonsingular, the first two compensators can generate arbitrary and ideal state feedback control $K_i = \overline{K}_i \overline{C}_i$ $(i = 1,\ 2)$. This result conforms with Conclusion 6.3 and Table 6.1. On the other hand, the matrices $\overline{C}_i = [T_i' : C']'$ $(i = 3,\ 4)$ have rank 6 and are singular. Hence only constrained state feedback control $K_i = \overline{K}_i \overline{C}_i$ $(i = 3,\ 4)$ can be implemented by the last two compensators. This result again conforms to Conclusion 6.4 and Table 6.1 because the third plant system has $\text{rank}(CB) = 1 < 2 = p$ and the fourth plant system has a nonminimum-phase zero (2).

For the third and fourth plant systems, there exists no other general and systematic design method which can fully use the design freedom to achieve feedback system performance and robustness. However, Algorithms 5.3 and 6.1 have systematically and generally designed the dynamic part of the dynamic output feedback compensator for these two plant systems as follows.

Because rank $(\overline{C}_i) = 6 < 7 = n,\ (i = 3,\ 4)$, we can select six out of the seven rows of $\overline{C}_i$ to form a new $\overline{C}_i$ $(i = 3,\ 4)$ so that Rank $(\overline{C}_i)$ still equals 6. Suppose we select the first three rows of matrix T_i and all three rows of matrix C to form $\overline{C}_i$ $(i = 3,\ 4)$. Then the new dynamic part of the corresponding dynamic output feedback compensator would be $(\overline{F}_i,\ \overline{T}_i,\ \overline{L}_i)$, which is formed by the first three rows of original (F_i, T_i, L_i) $(i = 3,\ 4)$, and the state feedback control gain implemented by these two compensators is

$$K_3 = \overline{K}_3[\overline{T}_3' : C']' = \overline{K}_3 \begin{bmatrix} 0 & -1 & 2 & 0 & 1 & -2 & 0 \\ 4 & 0 & -8 & -2 & 0 & 4 & 1 \\ 9 & 0 & -15 & -3 & 0 & 5 & 1 \\ & & & & & & \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & 1 & 0 & 0 & 0 & 0 & 0 \\ x & x & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$\begin{aligned} K_4 &= \overline{K}_4[\overline{T}_4' : C']' \\ &= \overline{K}_4 \begin{bmatrix} 2 & 1 & -1 & -2 & -1 & 1 & 2 \\ 4 & 2/5 & -12/5 & -2 & -1/5 & 6/5 & 1 \\ 9 & 0 & -6 & -3 & 0 & 2 & 1 \\ & & & & & & \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & 1 & 0 & 0 & 0 & 0 & 0 \\ x & x & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

respectively. These two state feedback controls are equivalent to a static output feedback control with six independent outputs. Therefore, they are not much weaker than the ideal state feedback control and are much stronger than the ordinary static output feedback control, which corresponds to only three outputs. For example, these two controls can arbitrarily assign the eigenvalues of the corresponding feedback system matrix $A - B\overline{K}_i\overline{C}_i$ $(i = 3,\ 4)$ because $6 + p = 6 + 2 = 8 > 7 = n$ [Kimura, 1975], while the ordinary static output feedback control cannot because $3 + p = 3 + 2 = 5 < 7 = n$.

More important, all four compensators guarantee that the feedback system poles be the union of $\{-1,\ -2,\ -3\}$ and the eigenvalues of $A - BK_i$ $(i = 1, \ldots, 4)$ (Theorem 4.1), and guarantee that the feedback system loop transfer function equals $-K_i(sI - A)^{-1}B(i = 1, \ldots, 4)$ (Theorem 3.4). This result certainly cannot be achieved systematically by other existing design methods for the third and fourth plant systems, especially the fourth, which is nonminimum-phase.

Example 6.2 The Case When Eigenvalues of F are Complex Conjugate

$$\text{Let}\begin{bmatrix} A & : & B \\ C & : & 0 \end{bmatrix} = \begin{bmatrix} 1.0048 & -0.0068 & -0.1704 & -18.178 & : & 39.611 \\ -7.7779 & 0.8914 & 10.784 & 0 & : & 0 \\ 1 & 0 & 0 & 0 & : & 0 \\ 0 & 0 & 0 & 0 & : & 1 \\ & & & & & \\ 1 & 0 & 0 & 0 & : & 0 \\ 0 & 1 & 0 & 0 & : & 0 \end{bmatrix}$$

This is the state space model of a combustion engine system [Liubakka, 1987]. Its four states are manifold pressure, engine rotation speed, manifold pressure (previous rotation), and throttle position, respectively. Its control input is the throttle position (next rotation) and its two output measurements are manifold pressure and engine rotation speed, respectively.

Apply the operation of Steps 2 and 3 of Algorithm 5.2 ($j = 2$), where the operator matrix

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -0.0093735 & 0.999956 \\ 0 & 0 & -0.99996 & -0.0093735 \end{bmatrix}$$

is determined by the elements $[-0.1704 \quad -18.178]$ of matrix A. The resulting block-observable Hessenberg form system matrices are

$$\begin{bmatrix} H'AH & : & H'B \\ CH & : & 0 \end{bmatrix}$$

$$
= \left[\begin{array}{ccccc}
1.0048 & -0.0068 & 18.1788 & 0 & : & 39.6111 \\
-7.7779 & 0.8914 & -0.1011 & 10.7835 & : & 0 \\
-0.00937 & 0 & 0 & 0 & : & -1 \\
1 & 0 & 0 & 0 & : & -0.0093735 \\
\\
1 & 0 & 0 & 0 & : & 0 \\
0 & 1 & 0 & 0 & : & 0
\end{array}\right]
$$

Because this system does not have any stable transmission zeros, we arbitrarily select matrix

$$
F = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}
$$

with eigenvalues $-1 \pm j$. Substituting matrices $H'AH$ and F into (5.13b) of Step 1, Algorithm 5.3, we have

$$
[D_1 : D_2]\left(I_2 \otimes H'AH \begin{bmatrix} 0 \\ I_2 \end{bmatrix} - F' \otimes \begin{bmatrix} 0 \\ I_2 \end{bmatrix} \right)
$$

$$
= \left[\begin{array}{ccccccccc}
-0.05501 & 0 & 1 & 0 & : & -0.05501 & 0 & 0 & 0 \\
-0.0005157 & -0.092734 & 0 & 1 & : & -0.0005157 & -0.092734 & 0 & 0 \\
\\
0.05501 & 0 & 0 & 0 & : & -0.05501 & 0 & 1 & 0 \\
0.0005157 & 0.092734 & 0 & 0 & : & -0.0005157 & -0.092734 & 0 & 0
\end{array}\right]
$$

$$
\times \left[\begin{array}{ccccc}
18.1788 & 0 & : & 0 & 0 \\
-0.1011 & 10.7835 & : & 0 & 0 \\
1 & 0 & : & 1 & 0 \\
0 & 1 & : & 0 & 1 \\
\\
0 & 0 & : & 18.1788 & 0 \\
0 & 0 & : & -0.1011 & 10.7835 \\
-1 & 0 & : & 1 & 0 \\
0 & -1 & : & 0 & 1
\end{array}\right] = 0
$$

Substituting $[D_1 \colon D_2]$ in (6.2b) of Algorithm 6.1, we have

$$\underset{\mathbf{c}}{[-0.0093735\ \ 1\ \ 0\ \ 0]}\begin{array}{c}\left[\begin{array}{ccc} -3.179 & : & -2.179 \\ -0.029801 & : & -0.02043 \\ & & \\ 2.179 & : & -3.179 \\ 0.02043 & : & -0.029801 \end{array}\right] \\ D_1 B \qquad\qquad D_2 B = 0 \end{array} = 0$$

Thus the result of Step 2 of Algorithm 5.3 is

$$T = \begin{bmatrix} \mathbf{c}D_1 \\ \mathbf{c}D_2 \end{bmatrix} = \begin{bmatrix} 0 & -0.092734 & -0.0093735 & 1 \\ 0 & -0.92734 & 0 & 0 \end{bmatrix}$$

This matrix corresponds to system matrix $(H'AH,\ H'B,\ CH)$. Hence it must be adjusted to correspond to the original system matrix $(A,\ B,\ C)$ (see the beginning of Sec. 5.2):

$$T = TH' = \begin{bmatrix} 0 & -0.92734 & 1 & 0 \\ 0 & -0.92734 & 0 & 0 \end{bmatrix}$$

Substituting this T into (5.16) of Step 3, Algorithm 5.3, we have

$$L = (TA - FT)\begin{bmatrix} I_2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1.721276 & -0.082663 \\ 0.721276 & -0.26813 \end{bmatrix}$$

It can be verified that $(F,\ T,\ L)$ satisfies (4.1) and (4.3), but the matrix $\overline{C} = [T' : C']'$ is singular. This is because the system has a nonminimum-phase zero (0.4589). Nonetheless, matrix $\overline{C}$ has one more linearly independent row than the original matrix C. Hence with the guaranteed robustness realization [by (4.3)], the compensator $F,\ T,\ L)$ of (4.10) realizes a stronger state feedback control $\overline{KC}\mathbf{x}(t)$ than $K_y C\mathbf{x}(t)$ of the ordinary static output feedback control.

In addition, Example 7.3 of Chap. 7 provides a numerical example about the multiple eigenvalue case of Algorithm 5.3. Thus complete eigenvalue cases have been shown by numerical examples in this book.

Example 6.3 A Case with Approximate Solution of (4.1) and (4.3)

Let the system matrix be

$$(A,\ B,\ C) = \left(\begin{bmatrix} x & x & 1 & 0 \\ x & x & 0 & 1 \\ x & x & 0 & 0 \\ x & x & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 3 \\ 1 & 2 \\ 2 & 6 \\ -1 & -2 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \right)$$

where “x”’s are arbitrary entries. Because this system has the same number of inputs and outputs $(m = p)$ and satisfies rank $(CB) = p = 2$, it has $n - m = 4 - 2 = 2$ transmission zeros. Because this system is in observable canonical form, using the procedure of Examples 1.7, matrix $N(s)$ of the polynomial matrix fraction description of the corresponding transfer function $G(s) = D^{-1}(s)N(s)$ can be directly derived as

$$N(s) = \begin{bmatrix} s+2 & 3(s+2) \\ s-1 & 2(s-1) \end{bmatrix}$$

Thus this system has two $(= n - m)$ transmission zeros $(-2$ and $1)$ and is nonminimum-phase.

Let us set $F = \text{diag}\,\{-2, -1\}$, where -2 matches the stable transmission zero of $(A,\ B,\ C)$ and -1 is arbitrarily chosen. Solving (5.10b), we have

$$D_1 = \begin{bmatrix} -2 & 0 & 1 & 0 \\ 0 & -2 & 0 & 1 \end{bmatrix} \quad \text{and} \quad D_2 = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

Substituting this result into (6.2a),

$$\mathbf{c}_1 D_1 B = \mathbf{c}_1 \begin{bmatrix} 0 & 0 \\ -3 & -6 \end{bmatrix} = 0$$

and

$$\mathbf{c}_2 D_2 B = \mathbf{c}_2 \begin{bmatrix} 1 & 3 \\ -2 & -4 \end{bmatrix} = 0 \tag{6.17}$$

imply that $\mathbf{c}_1 = [x \quad 0]$ $(x \neq 0)$, while the exact solution of $\mathbf{c}_2$ does not exist. This is because the corresponding transmission zero (-2) of $\mathbf{c}_1$ is matched by the eigenvalue of F, while the transmission zero (1) corresponding to $\mathbf{c}_2$ is not. This example shows that having $n - m$ stable transmission zeros is a necessary condition for a plant system with $(m = p)$ to have exact solution to (4.1), (4.3) and nonsingular $\overline{C}$. This example also conforms with Conclusion 6.1 (the second part).

To minimize $\mathbf{c}_2 D_2 B$ of (6.17) in a least-square sense, we use (6.3) of Algorithm 6.1 such that

$$\mathbf{c}_2 = \mathbf{u}_2' = [\,0.8174 \quad 0.576\,]$$

Here $\mathbf{u}_2$ is the normalized right eigenvector of matrix $[D_2 B][D_2 B]'$ and its smallest eigenvalue $\sigma_2^2 = 0.13393$. In other words, $\sigma_2 = 0.366$ is the smallest singular value of matrix $D_2 B$ and $\mathbf{u}_2$ is the second column of unitary matrix U of the singular value decomposition of $D_2 B$. It can be verified that $\|\mathbf{c}_2 D_2 B\| = \sigma_2$ which is the least-square residual of (6.17) (see Example A.6).

The above result provides us with two possible feedback compensators, whose dynamic part (3.16a) will be, respectively,

$$(F_1, T_1) = (-2, [-2\mathbf{c}_1 \ : \ \mathbf{c}_1])$$

and

$$(F_2, T_2) = \left(\begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} \multicolumn{2}{c}{-2\mathbf{c}_1} & : & \multicolumn{2}{c}{\mathbf{c}_1} \\ -0.8174 & -0.576 & : & 0.8174 & 0.576 \end{bmatrix} \right)$$

Of these two possible compensators, the first is a dynamic output feedback compensator (4.10) because it satisfies $TB = 0$, while the second does not satisfy $TB = 0$ and hence is an observer (3.16) only. Therefore, the first compensator guarantees that the feedback system loop transfer function equals $-\overline{K}_1[T_1' \colon C']' \ (sI - A)^{-1}B$ for whatever $\overline{K}_1$, while the second compensator does not (for its corresponding freely designed $\overline{K}_2$) (Theorem 3.4), even though the least-square gain TB of (6.17) is used in this observer.

On the other hand, the first compensator can implement only a constrained state feedback $\overline{K}_1[T_1' \colon C']'$ because Rank $[T_1' \colon C']' = 3 < 4 = n$, even though arbitrary eigenvalues can still be assigned to the matrix $A - B\overline{K}_1[T_1' \colon C']'$ because $3 + p = 3 + 2 = 5 > 4 = n$, while the second compensator can implement arbitrary state feedback $\overline{K}_2[T_2' \colon C']'$ because the matrix $[T_2' \colon C']'$ is nonsingular.

We recall for nonminimum-phase plant systems, such as the one in this example, that there is no other design method which can systematically and analytically derive as strong a result as these two compensators.

6.3 COMPLETE UNIFICATION OF TWO EXISTING BASIC MODERN CONTROL SYSTEM STRUCTURES

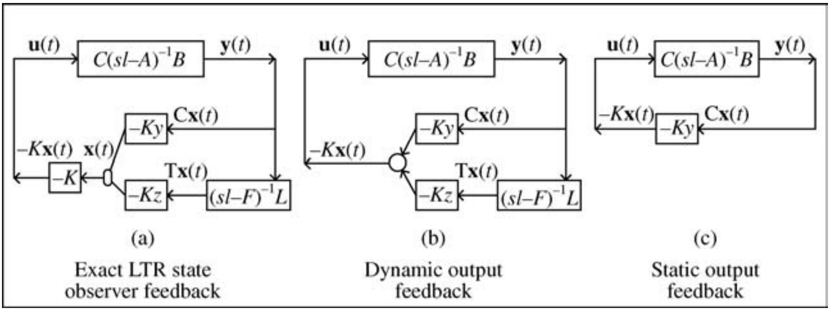
Besides general robustness realization, the new design approach of this book has another major theoretical significance. That is the complete unification of two existing basic control structures of modern control theory. These two basic structures are the exact LTR state observer feedback system and the static output feedback system. State observer and static output feedback have been the main control structures of modern control theory for years, but no attempt has been made to unify these seemingly very different structures.

The new control structure designed in this chapter—the dynamic output feedback controller which can implement state/generalized state feedback control, can completely unify the above two existing control structures as its two extreme cases. This unification can be shown in Fig. 6.1, and the properties of these three structures can be summarized in Table 6.2.

Table 6.2 shows clearly that the new control structure of this book [structure (b)] completely unifies in all aspects the existing two basic control structures of (a) and (c). The common feature which makes this unification *uniquely* possible is the realization of state feedback control [$K\mathbf{x}(t), K$ is a constant] and its robustness properties ($L(s) = -K(sI - A)^{-1}B$).

Table 6.2 shows that control structure (a) exercises the strongest control but is least generally designed, while control structure (c) exercises the weakest control but is general to all plant systems. The table also shows that control structure (b) completely unifies these two extreme properties.

A direct consequence of this unification is that the design of the output part of dynamic output feedback compensator $K = \overline{KC}$ is directly



(a)
Exact LTR state observer feedback

(b)
Dynamic output feedback

(c)
Static output feedback

Figure 6.1 Three modern control structures capable of realizing state/generalized state feedback control *and* their robustness properties.

Table 6.2 Three Control Systems of Modern Control Theory

Control structure	(a)	(b)	(c)
Controller order r	$n - m$	$n - m \geqslant r \geqslant 0$	0
Matrix $\overline{C} = [T' : C']'$	$[T' : C']'$	$[T' : C']'$	C
Rank $(\overline{C}) = q = r + m$	n	$n \geqslant r + m \geqslant m$	m
State feedback gain $K = \overline{KC}$	Arbitrary K ($\overline{C}$ nonsingular)	Arbitrary to severely constrained $K = \overline{KC}$	Severely constrained $K = K_y C$
Dynamic matrix	$A - BK$	$A - B\overline{KC}$	$A - BK_y C$
Loop transfer function	$-K(sI - A)^{-1}B$	$-\overline{KC}(sI - A)^{-1}B$	$-K_y C(sI - A)^{-1}B$
Generality (conditions on plant system)	$n - m$ stable transmission zeros or minimum phase, rank $(CB) = p$, and $m \geqslant p$	At least one stable transmission zero or $m > p$	None

compatible with the existing state feedback design (if $q = n$) and the existing static output feedback design (if $q < n$). This design will be described in Chaps 8 and 9.

6.4 OBSERVER ORDER ADJUSTMENT TO TRADEOFF BETWEEN PERFORMANCE AND ROBUSTNESS [Tsui, 1999c]

One of the main and unique features of the observers based on the result of Algorithm 6.3, is that the observer order r is completely flexible. On the contrary, the existing observer orders are fixed. For example, the state observer orders are fixed to be either n or $n - m$, and the order of a static output feedback controller is 0.

Also because of this unique feature, our observer compensator can completely unify exact LTR state observer and static output feedback control, as described clearly in Sec. 6.3.

The reason behind this unique feature is that the dynamic part of our observer compensator is completely decoupled. This is further enabled, uniquely, by the Jordan form of matrix F in (4.1) and in Algorithm 5.3, and by the design concept that a nonsingular matrix $\overline{C}$ in (4.2) is unnecessary (see Chap. 4).

Example 6.4

Example 6.1 (the third and the fourth compensators) and Example 6.3 (the first compensator) all show that when matrix F is in Jordan form and when a nonsingular matrix $\overline{C}$ is no longer required, the compensator order can be freely adjusted.

More specifically, the third and the fourth compensators of Example 6.1 have order $r = 3$ while $n - m = 4$, and the first compensator of Example 6.3 has order $r = 1$ while $n - m = 2$.

This section deals with the actual determination of this observer compensator order r. Our determination is based on the following two basic and clear understandings.

The first understanding is based on the formulation (4.2) of our control $K = \overline{KC}$, where $\overline{C}$ is formed by the rows of matrices C of (1.1b) and T of (4.1) and (4.3). Equation (4.2) is a constraint on the state feedback gain K (see Subsection 3.2.2). Therefore, the higher the observer order r (which equals the row rank of matrix T), the higher the row rank $(r + m)$ of matrix $\overline{C}$, the less the constraint on K (see Appendix A.1), and the more powerful the corresponding control $K\mathbf{x}(t)$.

The second understanding is based on Eq. (4.3) ($TB = 0$), which is the key condition for realizing the loop transfer function/robustness properties of our control. Because B is given, the smaller the row rank r of matrix T, the easier to satisfy (4.3) (see Appendix A.1).

In addition to these two simple and basic understandings, our observer order determination is further based on the following two obvious system design principles.

The first system design principle is that the system must be stable. Therefore, based on the first of the above two basic understandings, the order r has to be high enough so that the corresponding matrix $A - B\overline{KC}$ is stabilizable.

Stabilization, which only requires all eigenvalues of matrix $A - B\overline{KC}$ be in the stable region rather than in exact locations, is substantially easier than arbitrary eigenvalue assignment of matrix $A - B\overline{KC}$ (see Subsection 8.1.4). Now because rank $(\overline{C}) \times p > n$ is generically sufficient for arbitrary eigenvalue assignment of $A - B\overline{KC}$ [Wang, 1996], this condition should be sufficient for the stabilization of $A - B\overline{KC}$. Therefore, we should have a high enough observer order r such that

$$(r + m) \times p > n \qquad \text{or} \qquad r > \frac{n}{p} - m \tag{6.18}$$

This should be the lower bound of observer order r.

The second system design principle is that the effectiveness (especially the robustness property) of control $K\mathbf{x}(t)$ is totally lost if $TB \neq 0$. Therefore, based on the second of the above two basic understandings, the observer order r should be low enough so that TB can be sufficiently minimized.

Based on Conclusion 6.1 and the second column of Table 6.2, if the open-loop system (A, B, C) has either $m > p$ or at least one stable transmission zero, than $TB = 0$ can be fully satisfied [in addition to (4.1)]. Then from the first of the above two basic understandings, we should have the highest possible observer order r, say r', while keeping $TB = 0$ satisfied.

Definition 6.1

Let $r' \triangleq$ maximal possible rank $(\overline{C} \triangleq [T' : C']') - m$ where matrix T satisfies (4.1) and (4.3).

From Conclusion 6.3 and its proof, r' equals the number of stable transmission zeros of system (A, B, C) if $m \leqslant p$.

What is the value of r' for the cases of $m > p$? It differs from system to system, depends on parameters such as rank(CB) and the numbers of system stable and unstable transmission zeros (even though such systems generically do not have transmission zeros [Davison and Wang, 1974]), and ranges between 0 and $n - m$. There is no simple and general formula for r' directly from the parameters of system (A, B, C). Fortunately, Case B of Algorithm 6.1 guarantees the simple and direct computation of r', as convincingly argued by Conclusion 6.2.

There is another way to compute the value of r' and it computes r' before the computation of the solution of (4.1) and (4.3). This computation is based on a special similarity transformation on the system (A, B, C) called the "special coordinate basis (s.o.b.)" [Saberi et al., 1993]. In the s.o.b., the system is decoupled into five parts with five dimensions such as the number of system stable transmission zeros and the number of system unstable transmission zeros, etc. The value of r' can be determined easily from these five dimensions, because the state observers of some of these decoupled system parts of s.o.b. satisfy automatically $TB = 0$. However, it is obvious and it is accepted that the computation of this s.o.b. itself is very difficult and ill conditioned [Chu, 2000], even though numerically more stable algorithm of computing this s.o.b. is presented in Chu [2000].

In addition to the ill condition of the computation of s.o.b., the corresponding state observer of s.o.b. has order fixed at r' and is not adjustable at all. Then what if this r' cannot satisfy (6.18) or (6.19) (if a higher design requirement is imposed), or what if this r' is too high to be realized? These problems cannot be even discussed based on the state observers since the state observer order is fixed.

If $r' \geqslant r$ of (6.18), then (4.1), (4.3) and the stabilization of matrix $A - B\overline{KC}$ are guaranteed. Because (4.1) implies that the feedback system poles are composed of the eigenvalues of matrices F and $A - B\overline{KC}$ (Theorem 4.1), and (4.3) implies an output feedback compensator [see (4.10)], a solution to the strong stabilization problem is automatically derived by our design. The strong stabilization problem is defined as stabilizing the feedback system [say matrix $A - B\overline{KC}$] by a stable output feedback compensator [Youla et al., 1974 and Vidyasagar, 1985].

In practice a control system design that requires advanced control theory usually deserve both high performance and robustness, in addition to stability only. Therefore the control $\overline{KC}\mathbf{x}(t)$ should be able to assign arbitrary eigenvalues and at least some eigenvectors. Fortunately, such design algorithm is presented in Subsection 8.1.3, and is executable if $\text{rank}(\overline{C}) + p > n$. Therefore in such designs, it is required that at least

$$(r + m) + p > n \qquad \text{or} \qquad r > n - p - m \tag{6.19}$$

is satisfied.

It is proven mainly by the exercises of Chap. 4, and partially by Exercises 8.6 and 8.7, that (6.19) can be satisfied by most open-loop systems, and that (6.18) can be satisfied by a great majority of the open-loop systems.

Comparing the static output feedback controls where $r = 0$ (see Table 6.2 and Subsection 3.2.2), (6.18) and (6.19) cannot be satisfied as soon as $m \times p \leqslant n$ and $m + p \leqslant n$, respectively (see for Example 6.3 and Exercises 6.7 and 8.6).

In case the desired value of r of (6.19) or even (6.18) is higher than the value of r' (which guarantees $TB = 0$), the remaining $r - r'$ rows of T [or their corresponding $\mathbf{c}_i$ vectors of (6.1)] should be computed to make the corresponding matrix $\overline{C} \triangleq [T' : C']'$ full row rank instead of making TB (or $\mathbf{c}_i D_i B, i = r' + 1$ to r) $= 0$. Nonetheless, these $r - r'$ rows of T should still be selected out of the $n - m - r'$ rows of T and should still be computed, so that the product TB (or $\|\mathbf{c}_i D_i B\|, i = r' + 1$ to r) has the smallest possible magnitude.

EXERCISES

6.1 Verify the computation of Algorithm 6.1 to satisfy (4.3) for the four systems of Example 6.1.

6.2 Verify the computation of Algorithm 6.2 to satisfy (4.3) for Example 6.2.

6.3 Verify the computation of Algorithm 6.3 to satisfy (4.3) for Example 6.3.

6.4 Suppose matrices A, C, and D_i (and $\lambda_i, i = 1, \ldots, 4$) are all the same as that of Example 7.3. Let the matrix B be generally given. Repeat Algorithm 6.1.

(a) Let $\mathbf{c}_1 = [1, c_1, c_2]$. Compute $\mathbf{c}_1$ such that $\mathbf{c}_1 D_1 B = 0$.

Answer :

$$[\,c_1 \quad c_2\,] = -[-2 \quad 0 \quad -1 \quad 1 \quad 0 \quad 0 \quad 1\,]B \times \left(\begin{bmatrix} 1 & -1 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} B\right)^{-1}$$

(b) Let $\mathbf{c}_2 = [c_1, 1, c_2]$. Compute $\mathbf{c}_2$ such that $\mathbf{c}_2 D_2 B = 0$.

Answer :

$$[\,c_1 \quad c_2\,] = -[-1 \quad -2 \quad -1 \quad 1 \quad 1 \quad 0 \quad 0\,]B \times \left(\begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & -1 & 0 & 0 & 1 & 0 \end{bmatrix} B\right)^{-1}$$

(c) Let $\mathbf{c}_3 = [c_1, c_2, 1]$. Compute $\mathbf{c}_3$ such that $\mathbf{c}_3 D_3 B = 0$.

Answer :

$$[\,c_1 \quad c_2\,] = -[2 \quad 2 \quad -2 \quad 0 \quad 0 \quad 1 \quad 0\,]B \times \left(\begin{bmatrix} -2 & -2 & 1 & -1 & 0 & 0 & 1 \\ -3 & -3 & -1 & 1 & 1 & 0 & 0 \end{bmatrix} B\right)^{-1}$$

(d) Let $\mathbf{c}_4 = [1, c_1, c_2]$. Compute $\mathbf{c}_4$ such that $\mathbf{c}_4 D_4 B = 0$.

Answer :

$$[c_1 \quad c_2] = -[6 \quad 1 \quad -2 \quad 2 \quad 0 \quad 0 \quad 1]B \times \left(\begin{bmatrix} 3 & 0 & -1 & 1 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} B \right)^{-1}$$

Of course the $\mathbf{c}_i$ vectors do not need to (and some times cannot) be fixed at the above forms.

6.5 Change matrix B of the system of Example 6.1 to

$$\begin{bmatrix} 1 & 1 & 1 & : & -3 & -1 & -1 & : & 2 \\ 0 & 0 & 0 & : & 1 & 2 & 1 & : & -2 \end{bmatrix}'$$

so that $\text{Rank}(CB) = 1 = p - 1$ and the system has one unstable transmission zero 1. What is the value r' of this system?
Answer: $r' = 2 = n - m - 2$.

6.6 Change matrix B of the system of Example 6.1 to

$$\begin{bmatrix} 1 & 1 & 1 & : & -3 & -1 & -1 & : & 2 \\ 0 & 0 & 0 & : & 1 & 0 & 0 & : & -2 \end{bmatrix}'$$

so that $\text{Rank}(CB) = 1 = p - 1$ and the system has two unstable transmission zeros 1 and 2. What is the value r' of this system?
Answer: $r' = 1 = n - m - 3$.

6.7 Repeat Example 6.3 for a similar system

$$(A, B, C) = \left(\begin{bmatrix} x & x & 1 & 0 \\ x & x & 0 & 1 \\ x & x & 0 & 0 \\ x & x & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 3 & 6 \\ -1 & -3 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \right)$$

Instead of having -2 and 1 as transmission zeros of Example 6.3, this new system has -3 and 1 as transmission zeros.

6.8 In single-input and single-output systems $G(s) = D^{-1}(s)N(s)$, the condition $\text{rank}(CB) = p$ (or $CB \neq 0$) implies $N(s)$ has order $n - m = n - 1$. Thus the SISO systems have generically $n - 1$ zeros

[Davison and Wang, 1974]. Using the result of Example 1.7 and Exercises 1.3 to 1.6, repeat the above analysis on how the condition rank$(CB) = p$ will imply to the number of MIMO system transmission zeros.

7

Observer Design for Minimized Order

As stated at the beginning of Chap. 6, Step 2 of Algorithm 5.3 revealed the remaining freedom of (4.1). The first application of using this freedom is to realize the robustness properties of state feedback control, and is presented in Chap. 6. The second application of using this freedom is to minimize the observer order, and is presented in this chapter. The objectives of these two applications are very different.

Like the failure to realize the robustness properties of state feedback control, high observer order has also been a major drawback that has limited the practical application of state space control theory. Lower order observers not only are much easier to realize, but also have generally much smoother corresponding response. Like in Chap. 6, this chapter will demonstrate that with the full use of the remaining freedom of (4.1), this drawback can be effectively overcome.

However, unlike the design of Chap. 6 which determines only the dynamic part of the observer and which results in an output feedback compensator (4.10), the design of this chapter will completely determine the whole observer which cannot qualify as an output feedback compensator.

The design of this chapter is also based on the unique feature of the solution of (4.1) of Algorithm 5.3, that the rows of this solution (F, T, L) are completely decoupled. Thus the number of rows of this solution can be determined freely. From the observer definition of (3.16), this number equals the observer order r (see also Sec. 6.4 for the determination of r, but for a purpose totally different from a minimized r).

Section 7.1 describes the design formulation of this problem, which is claimed in Sec. 7.3 to be far simpler and the simplest possible general design formulation of this problem.

Section 7.2 presents the simple and systematic design algorithm (Algorithm 7.1) based on this formulation, and analyzes the general upper and lower bounds of r which is computed by this algorithm.

Section 7.3 proves that the general observer order bounds of Sec. 7.2 are far lower than the existing ones, are the lowest possible general bounds, and are lower enough to be practically significant even at the computer age. Several examples are presented to demonstrate this significance and Algorithm 7.1.

7.1 DESIGN FORMULATION [Tsui, 1985, 1993a]

As described in Example 4.3, minimal order observer design fully uses the remaining freedom of (4.1) to satisfy (4.2) [but not (4.3)] with arbitrarily given K, with arbitrarily given observer poles for guaranteed rate of observation convergence, and with a minimal value of r.

As reviewed in Example 4.3, minimal order observer design has been attempted for years since 1970 [Gopinath, 1971; Fortmann and Williamson, 1972; Kaileth, 1980, p. 527; Gupta et al., 1981; O'Reilly, 1983; Chen, 1984, p. 371; Van Dooren, 1984; Fowell et al., 1986]. But none has used the solution of (4.1) of Algorithm 5.3. This solution is uniquely decoupled and shows completely and explicitly the remaining freedom of (4.1) (see Sec. 5.2 and the beginning of Chaps 6 and 7). Thus only based on this solution of (4.1), can the minimal order observer design problem be simplified to the solving of (4.2) only and therefore really systematically.

As reviewed in Subsection 3.2.3, only Eq. (4.2) reveals the difference between different types of observers, such as the state observers of Examples 4.1 and 4.2 vs. the function observers of Definition 4.1, and such as the strictly proper type ($K_y = 0$) vs. the proper type ($K_y \neq 0$). The following design formulation (7.1c) and the corresponding design algorithm (Algorithm 7.1) are for proper type observers. However they can be very easily adapted to solve the strictly proper type observer problems.

Based on the block-observable Hessenberg form of (A, C), Eq. (4.2), like (4.1), can be partitioned into its left m columns:

$$K\begin{bmatrix} I_m \\ 0 \end{bmatrix} = [K_Z : K_y]\begin{bmatrix} T \\ C \end{bmatrix}\begin{bmatrix} I_m \\ 0 \end{bmatrix} = K_Z T\begin{bmatrix} I_m \\ 0 \end{bmatrix} + K_y C_1 \tag{7.1a}$$

and its right $n - m$ columns:

$$K\begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} \triangleq \check{K} = K_Z T\begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix} \triangleq K_Z \check{T} \tag{7.1b}$$

Because rank $(C_1) = m$ and K_y is completely free in (7.1a), only (7.1b) need to be satisfied.

To simplify the problem, we assume that all observer poles are distinct and real. Substituting the result of (6.1a) of Algorithm 5.3 (Step 1) into (7.1b), we have

$$\check{K} = K_Z\check{T} = K_Z \begin{array}{cc} \begin{bmatrix} \mathbf{c}_1 & & \\ & \ddots & \\ & & \mathbf{c}_r \end{bmatrix} & \begin{bmatrix} \check{D}_1 \\ \vdots \\ \check{D}_r \end{bmatrix} \\ m \quad \dots \quad m & n-m \end{array} \tag{7.1c}$$

where $\check{K}, \check{T}, \check{D}_i$, are the right $n - m$ columns of K, T, D_i $(i = 1, \dots, r)$ of (4.2) and (6.1a), respectively, and r equals the number of rows of matrix T or the corresponding minimal observer order.

The unknown solution of (7.1c) is K_Z and $\mathbf{c}_i$ $(i = 1, \dots, r)$, where parameter K_Z represents the design freedom of observer output part while $\mathbf{c}_i$ $(i = 1, \dots, r)$ represents the remaining freedom of observer dynamic part. The parameters $\mathbf{c}_i$ can also be considered the remaining freedom of (4.1), or the freedom of observer eigenvector assignment because F is in Jordan form. Hence the observer design freedom is fully used in (7.1c).

In addition, the given row blocks $\check{D}_i$ of (7.1c) are completely decoupled for all i because they are basis vector matrices of observer eigenvectors. Hence unlike *any other* existing minimal order observer design formulations, (7.1c) is truly very similar to a set of linear equations.

As a result, for the first time, (7.1c) can be solved systematically by matrix triangularization operations from the right side of the given matrix of (7.1c), and by back substitution (see Appendix A, Sec. A.2).

7.2 DESIGN ALGORITHM AND ITS ANALYSIS

The following design algorithm solves (7.1c) by matrix triangularization operations from the right side of the given equation of (7.1c), and by back substitution operation following each triangularization (see Sec. A.2).

Algorithm 7.1 Design of Minimal Order Observers [Tsui, 1985]

Step 1: Triangularize the following matrix S until it becomes

$$SH \triangleq \begin{bmatrix} \check{D}_1 \\ \vdots \\ \check{D}_{n-m} \\ \check{K} \end{bmatrix} H$$

$$= \left[\begin{array}{ccccccc} * & & 0 & \vdots & & & \\ & \ddots & & \vdots & & 0 & \\ X & & * & \vdots & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & X & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & \check{D}_{r_1+1}H & & & \\ & & & \vdots & & & \\ & & & \check{D}_{n-m}H & & & \\ & & & X & & & \\ x & \cdots & x & \vdots & 0 & \cdots & 0 \\ & & & X & & & \end{array}\right] \begin{array}{l} \\ \left.\begin{array}{l} \}\ (r_1-1)m+1 \\ \ \ \text{to } r_1m \text{ rows} \end{array}\right\} r_1m \text{ rows} \\ \\ \\ \\ \}m \\ \\ \}m \\ \\ \leftarrow \text{the } q_1\text{-th row } \mathbf{1}_1 \\ \\ \end{array} \tag{7.2}$$

$$\triangleq \overline{S}$$

Step 2: The form of $\overline{S}$ of (7.2) indicates that the q_1-th row of $\check{K}$ is a linear combination of the rows of $\check{D}_i$ $(i = 1, \ldots, r_1)$, or $\mathbf{1}_1 = \Sigma \mathbf{c}_i \check{D}_i H$ $(i = 1, \ldots, r_1)$. Compute $\mathbf{c}_i$ by back substitution. Also set the q_1-th row of matrix K_Z as $[1..1 : 0 \ldots 0]$ with r_1 "1"s.

Step 3: Triangularize the following matrix S_1 until it becomes:

$$
S_1 H_1 \triangleq \begin{bmatrix} \mathbf{c}_1 \check{D}_1 H \\ \vdots \\ \mathbf{c}_{r1} \check{D}_{r1} H \\ \check{D}_{r1+1} H \\ \vdots \\ \check{D}_{n-m} H \\ \check{K} H \end{bmatrix} H_1
= \left[\begin{array}{ccccccc}
* & & 0 & : & & & \\
 & \ddots & & : & & 0 & \\
X & & * & : & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & X & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & \check{D}_{r1+r2+1} H H_1 & & & \\
 & & & \vdots & & & \\
 & & & \check{D}_{n-m} H H_1 & & & \\
 & & & X & & & \\
x & \ldots & x & \vdots & 0 & \ldots & 0 \\
 & & & X & & &
\end{array}\right]
\begin{array}{l}
\left.\begin{array}{l} \\ r_1 + (r_2 - 1)m + 1 \\ \text{to } r_1 + r_2 m \text{ rows} \end{array}\right\} \\
\\
\\
\\
\}m \\
\vdots \\
\}m \\
\\
\leftarrow \text{the } q_2\text{-th row } \mathbf{1}_2 \\
(q_2 \neq q_1)
\end{array}
\qquad (7.3)
$$

$$\triangleq \overline{S}_1$$

Step 4: The form of $\overline{S}_1$ of (7.3) indicates that the q_2-th row of $\check{K}$ is a linear combination of the rows $\mathbf{c}_i \check{D}_i$ $(i = 1, \ldots, r_1)$ and the rows of $\check{D}_i$ $(i = r_1 + 1, \ldots, r_1 + r_2)$, or

$$
\mathbf{1}_2 = \sum_{i=1}^{r1} k_{2i} (\mathbf{c}_i \check{D}_i H H_1) + \sum_{i=r1+1}^{r1+r2} \mathbf{c}_i (\check{D}_i H H_1)
$$

Compute $[k_{21},\ldots,k_{2,r1}] \triangleq \mathbf{k}_2$ and $\mathbf{c}_i$ $(i = r_1 + 1,\ldots,r_1 + r_2)$, and then set the q_2-th row of K_Z as

$$\begin{array}{ccccccccc} [\mathbf{k}_2 & : & 1 & \ldots & 1 & : & 0 & \ldots & 0] \\ r_1 & & & r_2 & & & & & \end{array} \tag{7.4a}$$

Steps 3 and 4 are repeated until each of the p rows of $\check{K}$ is expressed as a linear combination of the rows of $\check{D}_i$ $(i = 1,\cdots,r_1 + .. + r_p \triangleq r)$, where r is the observer order.

Finally, parameters T and L are determined by Step 3 of Algorithm 5.3, and parameter K_y is determined by (7.1a).

Without loss of generality, we assume $q_i = i$ $(i = 1,\ldots,p)$, then the corresponding

$$K_Z = \begin{array}{c} \left[\begin{array}{ccccccccc} 1\ldots1 & : & 0 & : & & & \ldots & : & 0 \\ \mathbf{k}_2 & : & 1\ldots1 & : & 0 & : & \ldots & : & 0 \\ \multicolumn{3}{c}{\mathbf{k}_3} & : & 1\ldots1 & : & \ddots & : & 0 \\ & & & & & & \ddots & & \\ \multicolumn{7}{c}{\mathbf{k}_p} & : & 1\ldots1 \end{array}\right] \\ \begin{array}{ccccc} r_1 & \qquad r_2 & \qquad r_3 & \qquad \ldots & \qquad r_p \end{array} \end{array} \tag{7.4b}$$

It is obvious that observer order is tried and increased one by one starting from 0, in Algorithm 7.1. At any stage of this algorithm, if the calculated $\mathbf{c}_i = 0$, then the corresponding $\check{D}_i$ will be redeployed at the lower part of matrix S to express other rows of $\check{K}$. Therefore it is also obvious that all remaining freedom of (4.1) $(\mathbf{c}_i, i = 1,\ldots,r)$ is *fully* used.

Based on Conclusion 5.2 and the general assumption that

$$v_1 \geqslant v_2 \geqslant \cdots \geqslant v_m, \qquad \text{and that} \qquad r_1 \geqslant r_2 \geqslant \cdots \geqslant r_p \tag{7.5}$$

it is proven that [Tsui, 1986b] in Algorithm 7.1

$$r_i \leqslant v_i - 1, \qquad i = 1,\ldots,p \tag{7.6a}$$

Thus

$$r = (r_1 + \cdots + r_p) \leqslant (v_1 - 1) + \cdots + (v_p - 1) \tag{7.6b}$$

It is also proven that [Tsui, 1986b] in this algorithm

$$r \leqslant n - m \tag{7.7}$$

If parameter K_y is predetermined to be 0, then Eq. (7.1) becomes

$$\begin{array}{c} K = K_Z T = K_Z \begin{bmatrix} \mathbf{c}_1 & & \\ & \ddots & \\ & & \mathbf{c}_r \end{bmatrix} \begin{bmatrix} D_1 \\ \vdots \\ D_r \end{bmatrix} \\ \qquad\qquad\qquad\quad m \quad \ldots \quad m \qquad n \end{array} \tag{7.8}$$

Because the only difference between (7.8) and (7.1c) is that the former has m additional columns, Algorithm 7.1 can be used directly to design this type of minimal order observers, and (7.6a,b) and (7.7) can be replaced by

$$r_i \leqslant v_i, \qquad i = 1, \ldots, p \tag{7.9a}$$

$$r = (r_1 + \cdots + r_p) \leqslant v_1 + \cdots + v_p \tag{7.9b}$$

and

$$r \leqslant n \tag{7.10}$$

respectively. Now we have the complete formula for the general lower and upper bounds of orders of minimal order observers.

Table 7.1 shows that the order of a function observer which can implement arbitrary state feedback control varies between its lower and upper bounds. Unlike state observer orders, the actual value r of this order *depends* on the actual values of K and T (D_{i}'s) in either (7.8) (if $K_y = 0$) or (7.1) (if $K_y \neq 0$).

Table 7.1 Lower and Upper Bounds for Orders of Minimal Order Observers with Arbitrarily Given Poles

Observer type	Stateo observers $p = n, K = I$	Function observers ($p \leqslant n, K$ arbitrary, and $v_1 \geqslant \cdots \geqslant v_m$)
$K_y = 0$	$r = n$	$1 \leqslant r \leqslant \min\{n, v_1 + \cdots + v_p\}$
$K_y \neq 0$	$r = n - m$	$0 \leqslant r \leqslant \min\{n - m, (v_1 - 1) + \cdots + (v_p - 1)\}$

7.3 EXAMPLES AND SIGNIFICANCE OF THIS DESIGN [Tsui, 1998a]

Example 7.1

In the single-output case ($m = 1$), the basis vector matrices D_i of Algorithm 5.3 become row vectors $\mathbf{t}_i (i = 1, \ldots, r)$. Hence the corresponding (7.8) and (7.1) become

$$K = K_Z \underset{n}{\begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_r \end{bmatrix}} \qquad \text{and} \qquad \check{K} = K_Z \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_r \end{bmatrix} \underset{n-m}{\begin{bmatrix} 0 \\ I_{n-m} \end{bmatrix}}$$

respectively. Thus the upper bound of function observer order of this case is n and $n - m$, respectively. The lower bound remains 1 and 0 respectively, because K cannot be 0 while $\check{K}$ can.

The single output (SO) system is a special case of the multiple output (MO, $m \geqslant 1$) system in the sense of $m = 1$. Notice that for such systems $v_1 = n$, which makes the two terms of the upper bounds of r of Table 7.1 well unified with each other when $m = 1$. Example 7.1 shows that the observer order bounds of this special case is well unified by the bounds of Table 7.1.

The single input (SI) system is also a special case of the multiple input ($MI, p \geqslant 1$) system in the sense of $p = 1$. In this special case K is a row vector. The upper bound of r is v_1 and $v_1 - 1$ for the two types of observers respectively because of (7.9a) and (7.6a), respectively. Notice that $v_1 = n$ if $m = 1$. This makes the two terms of upper bounds of r unified with each other for $m = p = 1$. As p increases from 1 [or the problem is changed to generate more signals of $K\mathbf{x}(\mathrm{t})$], the upper bound of r should increase to $v_1 + \cdots + v_p$ or $(v_1 - 1) + \cdots + (v_p - 1)$ but should not exceed the most difficult state observer case n or $n - m$, respectively, for the two types of observers. Because the observability indices satisfy $v_1 + \cdots + v_m = n$ in Definition 5.1, the two terms of the upper bounds of r are also perfectly unified as p is increased up to m. This unification is not achieved by other existing general upper bounds of r such as pv_1 or $p(v_1 - 1)$ [Chen, 1984] because the v_i's may not be all the same.

For all SISO or MIMO cases, the lower bound of r is still 1 and 0 in (7.8) and (7.1c) respectively, also because K of (7.8) cannot be 0 while $\check{K}$ of (7.1c) can. The first case implies that K is a linear combination of the rows of D_1 (see Part (c) of Example 7.3 and Exercise 7.1 or see Part (d) of Exercise

7.1 for variation). The second case implies that the corresponding K is a linear combination of the rows of matrix C.

To summarize, the lower and upper bounds of minimal order observer order of Table 7.1 are perfectly and uniquely unified from SISO cases to the MIMO cases.

The derivation of (7.8) to (7.10) for the strictly proper observers ($K_y = 0$) and the derivation of (7.1) and (7.5) to (7.7) for the proper type observers ($K_y \neq 0$) also show that the bounds of r of Table 7.1 are also perfectly unified for these two types of observers.

For the state observer case when rank $(K = I) =$ maximum n, the upper bounds of r should reach the ultimate high levels n and $n - m$ for the two types of observers, respectively. For $K = I$, the matrix T of (7.8) and the matrix $[T' : C']'$ of (7.1) should be square and nonsingular for the two types of observers. Thus the number of rows of T (r) should be n and $n - m$ for the two types of observers, respectively. This is shown in Table 7.1. Thus Table 7.1 also unifies the state observer case and function observer case perfectly.

From the perfect unification of SISO and MIMO systems, the perfect unification of strictly proper and proper type observers, and the perfect unification of state and function observers, all bounds of observer order of Table 7.1 should be the lowest possible. Any other bound that is lower than any of these bounds of Table 7.1 cannot be general because it cannot unify the special cases.

Although the upper bounds of minimal function observer order is not as simple as that of the state observer order in Table 7.1, it often offers substantial order reduction in practice. The lower bounds (1 and 0) of r are the lowest possible and can be achieved by Algorithm 7.1 systematically whenever it applies (see Example 7.3 and Exercise 7.1). However, it is the upper bound that guarantees the significant order reduction from the state observer orders.

Because the observability indices satisfy $v_1 + \cdots + v_m = n$ in Definition 5.1, the upper bound of r of Table 7.1 is lower than the state observer order whenever $m > p$. In addition, this upper bound can be significantly lower than the state observer order in the situation that $p \ll m \ll n$ and that the v_i's are evenly valued. This situation is indeed common in practice because it is generally much easier to add measurements (or m) to a system than to add controls (of p) to a system.

Example 7.2

In a circuit system with 100 capacitors, 10 current or voltage meters, and 2 controlled-current or voltage sources, $n = 100, m = 10$ and $p = 2$. Given that $v_1 = \cdots = v_{10} = 10$ $(v_1 + \cdots + v_{10} = 100 = n)$, the function observer order of Algorithm 7.1 will not exceed $v_1 + v_2 = 20$ and $(v_1 - 1) + (v_2 - 1) = 18$, respectively (see Table 7.1 and Exercise 7.3).

This is significantly lower than the state observer order. In addition, it is possible that the function observer order can be systematically designed to be even lower than its upper bound of 20 or 18. The improvement from a hundredth-order compensator to a twentieth-order one can hardly be discounted, even by today's computer numerical computation capability.

The development of computer numerical computation capability should only be a challenge, instead of a reason of abandonment, for such research tasks as minimal order observer design. For example, the development of high-speed computers has now made possible the digital realization of a twentieth-order compensator of Example 7.2. In other words, the significance of Example 7.2 is feasible *because* of the computer development. It should be noted that the result of Table 7.1 is analytical and general. Hence the 100-to-20-order reduction of Example 7.2 can easily be a 1000-to-200-order reduction (assuming $n = 1000$ and $v_1 = \cdots = v_{10} = 100$; other parameters of Example 7.2 remain unchanged).

In addition, the unsuccessful past attempts of developing a simple, general, and systematic minimal order observer design algorithm should only be a challenge, instead of a reason of abandonment, for developing one.

Example 7.3 [Tsui, 1985]

Let the block-observable Hessenberg form system matrices be

$$A = \begin{bmatrix} -1 & 0 & 0 & : & 1 & 0 & 0 & : & 0 \\ 2 & 0 & 1 & : & -1 & 1 & 0 & : & 0 \\ 0 & 3 & 0 & : & 0 & 1 & 1 & : & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & : & -3 & 0 & 1 & : & 1 \\ 0 & 0 & 0 & : & 0 & 1 & 0 & : & -1 \\ 1 & 0 & 0 & : & 0 & 0 & -1 & : & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & 0 & : & 0 & 1 & 0 & : & -2 \end{bmatrix}$$

and

$$C = \begin{bmatrix} 1 & 0 & 0 & : & 0 & 0 & 0 & : & 0 \\ 1 & 1 & 0 & : & 0 & 0 & 0 & : & 0 \\ -1 & 0 & 1 & : & 0 & 0 & 0 & : & 0 \end{bmatrix}$$

From Definition 5.1, $v_1 = 3, v_2 = 2, v_3 = 2$. Let us design three minimal order observers for the following three state feedbacks:

$$K_1 = \begin{bmatrix} 3 & -2 & -2 & : & 1 & 2 & 1 & : & 0 \\ 2 & 0 & -1 & : & 1 & 1 & 0 & : & 0 \end{bmatrix}$$
$$K_2 = \begin{bmatrix} 2 & 0 & 2 & : & 1 & 0 & 1 & : & 1 \\ -3 & -3 & -2 & : & 1 & 2 & 0 & : & 0 \end{bmatrix}$$

and

$$K_3 = \begin{bmatrix} 0 & 2 & 3 & : & 0 & 0 & 0 & : & 0 \\ 1 & -1 & -1 & : & 1 & 1 & 0 & : & 0 \end{bmatrix}$$

From Table 7.1, the observer order cannot exceed $(v_1 - 1) + (v_2 - 1) = 3$. But let us first set the $n - m = 7 - 3 = 4$ possible eigenvalues of matrix F as $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} = \{-1, -2, -3, -1\}$.

In Step 1 of Algorithm 5.3 we compute the basis vector matrices D_i $(i = 1, 2, 3)$ from (5.10b) and D_4 from (5.15d) (based on D_1):

$$\begin{bmatrix} D_1 \\ \vdots \\ D_4 \end{bmatrix} = \begin{bmatrix} 2 & 0 & -1 & : & 1 & 0 & 0 & : & 1 \\ 1 & -1 & -1 & : & 1 & 1 & 0 & : & 0 \\ 0 & 0 & 0 & : & 0 & 0 & 1 & : & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & -1 & 0 & : & 0 & 0 & 0 & : & 1 \\ -1 & -2 & -1 & : & 1 & 1 & 0 & : & 0 \\ 1 & 1 & -1 & : & 0 & 0 & 1 & : & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -2 & -2 & 1 & : & -1 & 0 & 0 & & 1 \\ -3 & -3 & -1 & : & 1 & 1 & 0 & & 0 \\ 2 & 2 & -2 & : & 0 & 0 & 1 & : & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 6 & 1 & -2 & : & 2 & 0 & 0 & : & 1 \\ 3 & 0 & -1 & : & 1 & 1 & 0 & : & 0 \\ -1 & -1 & 1 & : & 0 & 0 & 1 & : & 0 \end{bmatrix}$$

We will apply Algorithm 7.1 to each of the three different K's. For simplicity of computation, we use elementary matrix operation (H) instead of orthonormal matrix operation (H) to triangularize the matrix of this example.

For K_1

Step 1:

$$SH = \begin{bmatrix} \check{D}_1 \\ \vdots \\ \check{D}_4 \\ ---- \\ \check{K}_1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \left[\begin{array}{ccccc} 1 & 0 & : & 0 & 0 \\ 1 & 1 & : & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & & 1 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & & 0 & 1 \\ 1 & 1 & & 0 & 0 \\ 0 & 0 & & 1 & 0 \\ -1 & 0 & & 0 & 2 \\ 1 & 1 & & 0 & 0 \\ 0 & 0 & & 1 & 0 \\ 2 & 0 & & 0 & -1 \\ 1 & 1 & & 0 & 0 \\ 0 & 0 & & 1 & 0 \\ & & & & \\ 1 & 2 & & 1 & 1 \\ 1 & 1 & : & 0 & 0 \end{array}\right] \leftarrow q_1 = 2, \mathbf{1}_1 \qquad (7.11)$$

Step 2: $r_1 = 1, \mathbf{c}_1 = [0 \quad 1 \quad 0], \mathbf{c}_1 \check{D}_1 H = \mathbf{1}_1 = [1 \quad 1 \quad 0 \quad 0].$

Step 3:

$$S_1H_1 = \begin{bmatrix} 1 & 1 & & 0 & 0 \\ 0 & 0 & & 0 & 1 \\ 1 & 1 & & 0 & 0 \\ 0 & 0 & & 1 & 0 \\ -1 & 0 & & 0 & 2 \\ 1 & 1 & & 0 & 0 \\ 0 & 0 & & 1 & 0 \\ & & \vdots & & \\ 1 & 2 & & 1 & 1 \\ 1 & 1 & & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & & 0 & 0 \\ 0 & 0 & & 0 & 1 \\ 1 & 0 & & 0 & 0 \\ 0 & 0 & & 1 & 0 \\ -1 & 1 & & 0 & 2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & & \vdots & & \\ & & \vdots & & \\ 1 & 1 & & 1 & 1 \\ x & x & & x & x \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \leftarrow \mathbf{l}_2 \\ \\ \end{matrix}$$

Step 4: $r_2 = 2, \mathbf{c}_2 = [-1 \quad 0 \quad 1], \mathbf{c}_3 = [1 \quad 0 \quad 0]$, and $\mathbf{k}_2 = 2$, so that

$$\mathbf{l}_2 = \mathbf{k}_2(\mathbf{c}_1\check{D}_1HH_1) + \mathbf{c}_2(\check{D}_2HH_1) + \mathbf{c}_3(\check{D}_3HH_1)$$

Finally, the observer for K_1 is $(r = r_1 + r_2 = 3)$, and

$$F = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix}$$

and

$$T = \begin{bmatrix} \mathbf{c}_1 D_1 \\ \mathbf{c}_2 D_2 \\ \mathbf{c}_3 D_3 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & 0 & 0 \\ 2 & 2 & -1 & 0 & 0 & 1 & -1 \\ -2 & -2 & 1 & -1 & 0 & 0 & 1 \end{bmatrix}$$

From (7.4a),

$$K_Z = \begin{bmatrix} \mathbf{k}_2 & : & 1 & 1 \\ 1 & : & 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & : & 1 & 1 \\ 1 & : & 0 & 0 \end{bmatrix}$$

From (5.10a) and (7.1a),

$$L = (TA - FT)\begin{bmatrix} I_3 \\ 0 \end{bmatrix} C_1^{-1} = \begin{bmatrix} 0 & -4 & -2 \\ 7 & 0 & 0 \\ -5 & -2 & 1 \end{bmatrix}$$

and

$$K_y = (K - K_Z T)\begin{bmatrix} I_3 \\ 0 \end{bmatrix} C_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

For K_2

Step 1: The result is similar to that of K_1 in (7.11), except the part

$$\check{K}_2 H = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix} \begin{matrix} \leftarrow q_1 = 1, \mathbf{1}_1 \\ \end{matrix}$$

Step 2: $r_1 = 1, \mathbf{c}_1 = [1 \quad 0 \quad 1]$ so that $\mathbf{c}_1(\check{D}_1 H) = \mathbf{1}_1$.

Step 3:

$$S_1H_1 = \begin{bmatrix} 1 & 0 & & 1 & 0 \\ 0 & 0 & & 0 & 1 \\ 1 & 1 & & 0 & 0 \\ 0 & 0 & & 1 & 0 \\ & & \vdots & & \\ & & \vdots & & \\ 1 & 0 & & 1 & 0 \\ 1 & 2 & & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ & & \vdots & \\ x & x & x & x \\ 1 & 2 & -1 & 1 \end{bmatrix} \leftarrow \mathbf{1}_2$$

Step 4: $r_2 = 1, \mathbf{c}_2 = [1 \quad 2 \quad 1]$, and $\mathbf{k}_2 = -1$ such that

$$\mathbf{1}_2 = \mathbf{k}_2(\mathbf{c}_1\check{D}_1HH_1) + \mathbf{c}_2(\check{D}_2HH_1)$$

Finally, the minimal order observer for K_2 is $(r = r_1 + r_2 = 2)$, and

$$F = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \qquad T = \begin{bmatrix} \mathbf{c}_1D_1 \\ \mathbf{c}_2D_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 & -1 & 1 & 0 & 1 & 1 \\ -2 & -4 & -3 & 2 & 2 & 1 & 1 \end{bmatrix}$$

$$K_Z = \begin{array}{c} \begin{bmatrix} 1 & : & 0 \\ \mathbf{k}_2 & : & 1 \end{bmatrix} \\ r_1 \qquad r_2 \end{array} = \begin{bmatrix} 1 & : & 0 \\ -1 & : & 1 \end{bmatrix}$$

$$L = (TA - FT)\begin{bmatrix} I_3 \\ 0 \end{bmatrix} C_1^{-1} = \begin{bmatrix} 2 & -2 & -1 \\ -3 & -16 & -10 \end{bmatrix}$$

and

$$K_y = (K - K_Z \mathrm{T}) \begin{bmatrix} I_3 \\ 0 \end{bmatrix} C_1^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

For K_3
Because the first row of K_3 is already a linear combination of rows of C, we let $r_1 = 0$ and let the linear combination coefficients be $\mathbf{k}_1$. Then, because the second row of $\check{K}_3$ equals the second row of $\check{K}_1$, we have the following minimal order observer for $K_3 : (r = r_1 + r_2 = 0 + (r_1 \quad \text{for} \quad K_1) = 1)$

$$\begin{aligned} F = -1, T &= \text{the first } r_1 \ (= 1) \text{ rows of } T \text{ for } K_1 \\ &= [1 \quad -1 \quad -1 \quad 1 \quad 1 \quad 0 \quad 0] \end{aligned}$$

$$K_Z = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad K_y = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} & \mathbf{k}_1 & \\ 0 & 0 & 0 \end{bmatrix}$$

$$L = [0 \quad -4 \quad -2]$$

To summarize, the order of the three minimal order observers is 3, 2, and 1, respectively, which is systematically and generally determined by Algorithm 7.1. All three orders do not exceed $(v_1 - 1) + (v_2 - 1) = 3$, which is the upper bound of Table 7.1.

The minimal order observer design problem has been studied using classical control methods also. The most recent result can be found in Chen [1984] and Zhang [1990]. Although for years the upper bound of minimal order observer order from these methods has been $\min\{n - m, \ p(v_1 - 1)\}$ [Chen, 1984], (see Exercise 7.4), the classical control methods differ much from Algorithm 7.1 in determining systematically and generally the lowest possible observer order (see Example 7.3 and the argument between (7.1c) and Algorithm 7.1). The difference appears at how systematically the equation [such as (7.1c)] is being solved, at how the observer dynamic part is decoupled, and at how fully the design freedom (such as the free parameters $\mathbf{c}_i$) is being used. It seems that the classical control methods cannot match Algorithm 7.1 in the above three technical aspects.

The foremost theoretical significance of Algorithm 7.1 is the simplification of the design problem into a *true* set of linear equations (7.1c) or (7.8) with fully usable freedom. The general and lowest possible lower and upper bounds of minimal order observer order (Table 7.1) are also derived simply based on this set of linear equations. Thus it can be claimed with confidence that this set of linear equations is already the simplest possible theoretical and general form of the minimal order observer design problem [Tsui, 1993a]. From Example 4.3 and Algorithm 7.1, this development is enabled solely by the development on the decoupled solution of (4.1) (Algorithm 5.3). Other state space minimal order observer design methods cannot reach this simple form because the result of Algorithm 5.3 has not been used [Van Dooren, 1984; Fowell et al., 1986].

The actual solving of this set of linear equations is technical. Although Algorithm 7.1 is general and systematic, guarantees the upper bound of observer order of Table 7.1 and tries the observer order one by one (starting from 0), it still has room for improvement. This algorithm operates on the D_i matrices in the sequence of $i = 1, 2, \ldots,$ but does not try *different sequences* among these matrices, which may offer additional observer order reduction. For example, if operating in the sequence of (D_1, D_2, D_3), Algorithm 7.1 can detect that K is linearly dependent on the rows of D_1 and D_2 ($r = 2$), but it is still possible that operating on a different sequence of (D_3, D_2, D_1) the Algorithm 7.1 can detect that K is linearly dependent on the rows of D_3 only ($r = 1$) (see Exercise 7.1, Part (d)).

In the literature, there are other reports of minimizing function observer order by observer pole selection [Fortmann and Williamson, 1972; Whistle, 1985]. However, these design methods are much more complicated, while the additional observer order reduction offered by these methods is not generally significant.

Finally, it should be emphasized again that the minimal order observer design (Algorithm 7.1) uses up completely the remaining design freedom of (4.1) (or of the observer) and therefore cannot take care of the robustness of the corresponding observer feedback system [such as (4.3)]. Hence this design is useful only for the situation in which the plant system model and measurements are accurate—and continue to be accurate—and that disturbance and failure are relatively free. In other words, the minimal order observer should be used when only performance (but not robustness) is required.

Although minimal order observer and dynamic output feedback compensator (capable of implementing state feedback control) differ from each other in design priority, both their designs are part of Step 2 of Algorithm 5.3 and both are in the similar form of sets of linear equations. Also, they both are successful and actual attempts of the basic observer design concept—implementing state feedback control directly without explicit information of system states. In addition, both order reduction (which is part of performance) and robustness are important system properties, even though the emphasis of this book is more on robustness properties. Therefore, both results are useful and both may be used in some situations.

EXERCISES

7.1 Repeat Example 7.3 for a modified system of Example 6.1:

$$\text{Let } C = \begin{bmatrix} 1 & 0 & 0 : 0 & 0 & 0 & 0 \\ 2 & 1 & 0 : 0 & 0 & 0 & 0 \\ 3 & 4 & 1 : 0 & 0 & 0 & 0 \end{bmatrix}$$

and let matrices D_i $(i = 1, \ldots, 4)$ be the same as that of Example 6.1. The system has parameters $n = 7, m = 3, p = 2, v_1 = 3,$ and $v_2 = v_3 = 2$.

(a)

$$K = \begin{bmatrix} 1 & -1 & 0 : & 1 & 2 & 3 & 1 \\ 0 & 0 & 1 : & -4 & 3 & -2 & 0 \end{bmatrix}$$

$$\textit{Answer}: \quad r = 3, T = \begin{bmatrix} [\ \ 3 & 2 & 3]D_1 \\ [-2 & 0 & 0]D_2 \\ [\ \ 4 & 3 & -2]D_3 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & -2 & -3 : & -3 & 2 & 3 & 3 \\ -8 & 0 & 0 : & 4 & 0 & 0 & -2 \\ 36 & -9 & 6 : & -12 & 3 & -2 & 4 \end{bmatrix}$$

$$K_Z = \begin{bmatrix} 1 & 1 & : & 0 \\ 0 & 2 & : & 1 \end{bmatrix} \qquad K_y = \begin{bmatrix} 19 & -11 & 3 \\ -63 & 29 & -5 \end{bmatrix}$$

(b)

$$K = \begin{bmatrix} 1 & 0 & 0 : 1 & 2 & 3 & 1 \\ 0 & -1 & 1 : 2 & -4 & -6 & -4 \end{bmatrix}$$

$$\textit{Answer}: \quad r = 2, T = \begin{bmatrix} [\ \ 3 & 2 & 3]D_1 \\ [-2 & 0 & 0]D_2 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & -2 & -3 : & -3 & 2 & 3 & 3 \\ -8 & 0 & 0 : & 4 & 0 & 0 & -2 \end{bmatrix}$$

$$K_Z = \begin{bmatrix} 1 & 1 \\ -2 & -1 \end{bmatrix} \qquad K_y = \begin{bmatrix} 17 & -10 & 3 \\ -17 & 15 & -5 \end{bmatrix}$$

(c) $$K = \begin{bmatrix} 11 & 7 & 1 : 0 & 0 & 0 & 0 \\ -6 & 2 & -4 : 6 & -2 & 4 & -6 \end{bmatrix}$$

Answer : $$r = 1, F = -1, T = [3 \quad 1 \quad -2]\, D_1$$
$$= [3 \quad -1 \quad 2 : -3 \quad 1 \quad -2 \quad 3]$$
$$K_Z = \begin{bmatrix} 0 \\ -2 \end{bmatrix} \qquad K_y = \begin{bmatrix} 2 & 3 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

(d) $$K = \begin{bmatrix} 3 & 2 & -3 : -1 & -2/3 & 1 & 1/3 \\ 10 & 6 & 1 : \;\; 0 & 0 & 0 & 0 \end{bmatrix}$$

Answer : $$r = 1, F = -3, T = [1 \quad -2 \quad 3]\, D_3$$
$$= [9 \quad 6 \quad -9 : -3 \quad -2 \quad 3 \quad 1]$$
$$K_Z = \begin{bmatrix} 1/3 \\ 0 \end{bmatrix} \qquad K_y = \begin{bmatrix} 0 & 0 & 0 \\ 3 & 2 & 1 \end{bmatrix}$$

7.2 Let a system and its state feedback gain be given as

$$(A, B, C, K) = \left(\begin{bmatrix} 0 & 0 & 5 \\ 1 & 0 & -5 \\ 0 & 1 & -2 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, [0 \quad 0 \quad 1], \frac{[0 \quad -5 \quad 3]}{8} \right)$$

Design a minimal order observer according to Algorithm 7.1.

(a) Let $K_y = 0$ and observer poles $= \{-5.25, -2.44, -4\}$.
Answer: $r = 2 < n$.

(b) Let $K_y \neq 0$, and observer poles $= \{-5/3, -10/3\}$.
Answer: $r = 1 < n - m$.

7.3 In Example 7.2, let $n = 1000, m = 100, p = 2, v_1 = \cdots = v_{100} = 10$, and $K_y = 0$. What are the state observer order and the upper bound of minimal observer order of Table 7.1?
Answer: $n = 1000$ and $v_1 + v_2 = 20$.

7.4 Let $n = 21, m = 5, p = 2, v_1 = 9, v_2 = \cdots = v_5 = 3$, and $K_y \neq 0$. What are the state observer order, the upper bound $p(v_1 - 1)$ of the existing minimal observer order, and the upper bound of our minimal observer order of Table 7.1?
Answer: $n - m = 16$; $p(v_1 - 1) = 16$; $v_1 + v_2 - 2 = 10$.

8

Design of Feedback Control—Eigenstructure Assignment

The new design approach of this book is divided into two major steps. The first concerns the dynamic part of the observer/compensator and is covered in Chap. 6 (for robustness realization). The second step, which is covered by Chaps 8 and 9, deals with the design of the output part of the compensator, or the design of the generalized state feedback control $\overline{KC}\mathbf{x}(t)$ with a given $\overline{C}$. This design also fully determines the feedback system loop transfer function because (4.3) is already guaranteed.

Among the existing design results of this control, the eigenvalue and eigenvector assignment (called "eigenstructure assignment") and linear quadratic optimal control are perhaps most commonly known, and are capable of considering effectively both performance and robustness. In particular, according to the analysis of Chap. 2, the eigenvalues and eigenvectors can determine system performance and robustness far more directly and explicitly than other indicators. Hence their assignment should improve feedback system performance and robustness distinctly effectively.

In this book, eigenstructure assignment design methods and linear quadratic optimal control design methods are introduced in Chaps 8 and 9, respectively.

The design of the generalized state feedback control $\overline{KC}\mathbf{x}(t)$ is based on the single overall feedback system matrix $A - B\overline{KC}$. Therefore if the design of Chap. 6 is based mainly on the understanding of feedback systems of Chaps 3 and 4, then the design of Chaps 8 and 9 is based mainly on the analysis of the single overall system of Chap. 2.

According to Table 6.2, the generalized state feedback control $\overline{KC}\mathbf{x}(t)$ unifies the arbitrary state feedback control (or state feedback control) $K\mathbf{x}(t)$ (if rank $(\overline{C}) = n$) and static output feedback control (if rank $(\overline{C} = C) = m$). Both Chaps 8 and 9 present the design methods in these two categories. The arbitrary state feedback control, which is a special case of the generalized state feedback control in the sense of $\overline{C} = I$, is presented first.

8.1 SELECTION AND PLACEMENT OF FEEDBACK SYSTEM POLES

8.1.1 Eigenvalue (Pole) Selection

Although system poles most directly determine system performance, there are no general, explicit and optimal rules for feedback system pole selection. Furthermore, there is no real optimal pole selection without trial and error. This is because plant systems are usually very different and complex, and also because the performance and robustness design requirements are contradictory to each other.

Nonetheless, there are still some basic and general understandings about the relationship between the system poles and the system performance and robustness. The following six general rules of pole selection are guided by these basic understandings (see Truxal, 1955 and Conclusion 2.2).

(a) The more negative the real part of the poles, the faster the speed with which the system reaches its steady state.

(b) In regulator problems, it is often required that the zero frequency response of the control system $T(s = 0)$ be a finite constant. For example, if the unit-step response of a single-input and single-output system $y(t)$ is required to approach 1 at steady state ($t \to \infty$), then $y(t \to \infty) = sY(s \to 0) = sT(s \to 0)/s = T(s \to 0) = 1$.

This implies that $N(s=0)=D(s=0)$ in $T(s)=N(s)/D(s)$. It is well known that $N(s=0)$ equals the product of zeros of $T(s)$ and is invariant under state feedback control [Patel, 1978]. Therefore the relation $N(s=0)=D(s=0)$ imposes a constraint on $D(s=0)$, which equals the product of the poles of feedback system $T(s)$.

(c) From the results of root locus, the further away the feedback system poles from the loop transfer function poles, the higher the loop gain (or feedback control gain) required to place these feedback system poles. The severe disadvantages of high feedback control gain are listed in Subsection 3.1.2.

If rule (a) is concerned mainly with system performance, then rules (b) and (c) are concerned mainly with robustness, and are constraints on rule (a).

(d) If the eigenvalues of a matrix differ too much in magnitude, then the difference between the largest and the smallest singular values of that matrix will also differ too much. This implies the bad condition and the bad robustness of the eigenvalues, of that matrix.

(e) Multiple eigenvalues can cause defective eigenvectors (5.15d), which are very sensitive to matrix parameter variation (see Golub and Wilkinson, 1976b) and which generally result in rough responses (see Example 2.1 and Fig. 2.1). Therefore multiple poles, even clustered poles, should generally be avoided.

(f) For some optimal control systems in the sense of minimal "Integral of time multiplied by absolute error (ITAE)" [Graham and Lathrop, 1953]:

$$J=\int_0^\infty [t|y(t)-1|]dt$$

or in the sense of minimal "Integral of quadratic error (ISE)" [Chang, 1961]:

$$J=\int_0^\infty [q(y(t)-1)^2+ru(t)^2]dt, \qquad q\to\infty$$

the feedback system poles are required to have similar magnitude and evenly distributed phase angles between +90° and −90°. This result conforms with rules (d) and (e).

These six rules are concerned more with the effectiveness and limitations of practical analog control systems. In contrast, the selection of feedback compensator poles (see the beginning of Sec. 5.2) are more specifically and explicitly guided. The feedback compensators are usually digital and can therefore be made ideal and precise, while the analog systems cannot be made ideal and precise.

To summarize, the pole selection rules are neither exhaustive nor generally optimal. This should be a true and reasonable reflection of the reality of practical engineering systems, and should impose a challenge to control engineers.

8.1.2 Eigenvalue Assignment by State Feedback Control

The eigenvalue assignment design methods are presented in this subsection and in Subsection 8.1.3, for arbitrary state feedback control $K\mathbf{x}(t)$ and generalized state feedback control $\overline{KC}\mathbf{x}(t)$, respectively. These design methods have the distinct property that the corresponding eigenvectors are expressed in terms of their corresponding basis vectors, and can therefore be assigned by really systematic and effective numerical methods. These eigenvector assignment design methods will be presented in Sec. 8.2.

Let Λ be the Jordan form matrix that is formed by the selected eigenvalues of Subsection 8.1.1. Then the eigenstructure assignment problem can be formulated as (1.10):

$$(A - BK)\mathrm{V} = V\Lambda \tag{8.1a}$$

or

$$AV - V\Lambda = BK^{\wedge}(K^{\wedge} = KV) \tag{8.1b}$$

Let matrix F of Eq. (4.1) be the same Jordan form matrix (in transpose) as Λ, and be set to have dimension n, then this equation

$$TA - FT = LC$$

becomes the dual of (8.1b). In other words, we can take the transpose of both sides of (8.1b) and then consider the resulting A', B', V', and $K^{\wedge\prime}$ as the matrices A, C, T, and L of Eq. (4.1), respectively.

Therefore, we can use the dual version of Algorithm 5.3 to compute directly the solution $(V, K^{\wedge})$ of (8.1b). Incidentally, Algorithm 5.3 and its dual version were published formally in the same year in Tsui [1985] and Kautsky et al. [1985], respectively.

The only difference between these two design computations is that after $K^{\wedge}$ is computed from (8.1b), it must be adjusted to $K = K^{\wedge}V^{-1}$ because only matrix K corresponds to the original feedback dynamic matrix $A - BK$. This adjustment is unnecessary in observer design because the observer dynamic matrix in (3.16) is matrix F instead of matrix $A - LC$.

The dual version of Algorithm 5.3 is introduced in the following with some simplifications.

Let the Λ_i be an n_i-dimensional Jordan block of Λ.

Let

$$V_i \underset{=}{\triangle} [\mathbf{v}_{i1}|\dots|\mathbf{v}_{\text{ini}}] \qquad \text{and} \qquad K_i \underset{=}{\triangle} [\mathbf{k}_{i1}|\dots|\mathbf{k}_{\text{ini}}]$$

be $n \times n_i$ and $p \times n_i$ dimensional, and be the part of matrices V and $K^{\wedge}$ corresponding to Λ_i in (8.1b), respectively.

Then (8.1b) can be partitioned as

$$AV_i - V_i\Lambda_i = BK_i, \qquad i = 1, \dots, r \tag{8.2}$$

where r is the number of Jordan blocks in Λ and $n_1 + \cdots + n_r = n$.

Using the Kronecker product operator $\otimes$, Eq. (8.2) can be rewritten as

$$[I_{ni}\otimes A - \Lambda_i\otimes I| - I_{ni}\otimes B]\mathbf{w}_i = 0,\ i = 1, \dots, r \tag{8.3a}$$

where

$$\mathbf{w}_i = [\mathbf{v}'_{i1} : \dots : \mathbf{v}'_{ini} : \mathbf{k}'_{i1} : \dots : \mathbf{k}'_{ini}]' \tag{8.3b}$$

For example, when $n_i = 1$, (8.3) becomes

$$[A - \lambda_i I : -B]\begin{bmatrix}\mathbf{v}_i \\ \mathbf{k}_i\end{bmatrix} = 0 \tag{8.4}$$

Because the matrix of (8.3a) has dimension $n_i n \times n_i(n+p)$ [see (5.13c)], and because controllability criterion implies that all rows of this matrix are linearly independent (see Definition 1.2), the vector $\mathbf{w}_i$ of (8.3) has $n_i \times p$ basis vectors and can be set as an arbitrary linear combination of these basis vectors. Naturally, the determination of this linear combination constitutes the assignment of eigenvectors $V_i = [\mathbf{v}_{i1} : \dots : \mathbf{v}_{\text{ini}}]$.

For example, when $n_i = 1$, the matrix of (8.3) or (8.4) has dimension $n \times (n+p)$. Hence eigenvector $\mathbf{v}_i$ of (8.4) can be an arbitrary linear combination of its p corresponding basis vectors $\mathbf{d}_{ij}$ $(j = 1, \ldots, p)$ which also satisfy (8.4):

$$\mathbf{v}_i = [\mathbf{d}_{i1} : \ldots : \mathbf{d}_{ip}]\mathbf{c}_i \underset{=}{\triangle} D_i \mathbf{c}_i \tag{8.5}$$

where $\mathbf{c}_i$ is a p-dimensional free column vector.

The vector $\mathbf{k}_i$ of (8.4) will be the same linear combination (coefficient vector is $\mathbf{c}_i$) of its own corresponding basis vectors.

If $p = 1$ (single-input case) and $n_i = 1$, then the matrix of (8.3) or (8.4) has dimension $n \times (n+1)$. Hence the solution $\mathbf{v}_i$ and $\mathbf{k}_i$ is unique ($\mathbf{c}_i$ is a scalar). This implies that in single-input case, there is no eigenvector assignment freedom, and the eigenvalues alone can uniquely determine the feedback system dynamic matrix.

Equation (8.5) is a uniquely explicit and uniquely decoupled formulation of eigenvector assignment. Only based on this formulation, the general and systematic design algorithms for robust eigenvector assignment are developed in Kautsky et al. [1985]. These methods will be introduced in Sec. 8.2.

Equations (8.3) and (8.4) are the formulas for computing the basis vectors of eigenvector matrix V. Like Step 1 of Algorithm 5.3, this computation can be carried out by direct back substitution if based on the block-controllable Hessenberg form

$$[A : B] = \begin{bmatrix} A_{11} & A_{12} & \ldots & \ldots & A_{1\mu} & : & B_1 \\ B_2 & A_{22} & \ldots & \ldots & : & : & 0 \\ 0 & B_3 & \ldots & \ldots & : & : & 0 \\ \vdots & \ddots & \ddots & & : & : & \vdots \\ 0 & \ldots & 0 & B_\mu & A_{\mu\mu} & : & 0 \end{bmatrix} \tag{8.6}$$

where matrix blocks B_j $(j = 1, \ldots, \mu)$ are the upper echelon-form matrices, and μ is the largest controllability index of the system (A, B).

As the dual of the observability index of Definition 5.1, there are p controllability indices μ_j $(j = 1, \ldots, p)$ of system (A, B) and

$$\mu_1 + \mu_2 + \cdots + \mu_p = n \tag{8.7}$$

In addition, each basis vector of (8.5) $\mathbf{d}_{ij}$ $(i = 1, \ldots, n, j = 1, \ldots, p)$ can be computed corresponding to one of the p inputs which is indicated by j.

If the $\mathbf{d}_{ij}$ vectors are computed this way, then from the dual of Conclusion 5.2, for a fixed value of j, any set of μ_j of the n $\mathbf{d}_{ij}$ vectors are linearly independent of each other (see Example 8.6 and Theorem 8.1). This analytical property is very useful in the analytical rules of eigenvector assignment (Subsection 8.2.2).

If matrix V is computed based on a similarity transformation (HAH', HB) instead of the original (A, B), [one example of (HAH', HB) is the block-controllable Hessenberg form (8.6)], then the corresponding (8.1b) becomes

$$HAH'V - V\Lambda = HBK\hat{} \tag{8.8}$$

A comparison of (8.1b) and (8.8) indicates that the matrix V of (8.8) should be adjusted to $V = H'V$ in order to correspond to the original system matrix (A, B).

As stated following (8.1b), after this adjustment of V, it will then be used to adjust the feedback gain matrix $K = K\hat{}V^{-1}$.

8.1.3 Eigenvalue Assignment by Generalized State Feedback Control

The generalized state feedback control gain is $\overline{KC}$, where $\overline{K}$ is free and rank $(\overline{C}) \triangleq q \leqslant n$ (see Table 6.2). The case for $q = n$ is equivalent of the state feedback control, and is covered in the previous subsection. This subsection deals with the case for $q < n$, which implies additional restrictions $K = \overline{KC}$ to the corresponding state feedback gain K, and whose design can therefore be much more difficult than the case for $q = n$.

Let Λ be a Jordan form matrix which contains the desired eigenvalues of matrix $A - B\overline{KC}$. Then from (1.10), the eigenvalue assignment problem can be expressed in the following dual equations:

$$T(A - B\overline{KC}) = \Lambda T \tag{8.9a}$$

and

$$(A - B\overline{KC})V = V\Lambda \tag{8.9b}$$

where T and $V(TV = I)$ are the left and right eigenvector matrices of $A - B\overline{KC}$ corresponding to Λ, respectively.

This problem has a remarkable property that is not shared by either state feedback design problem or the observer design problem—duality (see Sec. 1.3). Unlike in problem (8.9), in those two problems, the given and dual system matrices B and $\overline{C}$ do not appear in the problem simultaneously.

The following algorithm [Tsui, 1999a] uses two steps (Steps 1 and 2) to satisfy (8.9a) and then (8.9b). One of the unique features of this algorithm, and this feature is also shared by the method of Subsection 8.1.2, is that it allows the corresponding eigenvector assignment to be in the form of assigning the linear combination coefficients of the corresponding basis vectors. See the beginning of Subsection 8.1.2.

Algorithm 8.1

Eigenstructure assignment by generalized state feedback control [Tsui, 1999a].

The algorithm is aimed at partially satisfying (8.9a) and then (8.9b) (and $TV = I$). Because (8.9a) and (8.9b) are redundant, this partial satisfaction of (8.9a) and (8.9b) also implies the complete satisfaction of (8.9a) and (8.9b), as will be evident at Step 2 of the algorithm.

Step 0: Partition the matrix Λ of (8.9) into

$$\Lambda = \mathrm{diag}\{\Lambda_{n-q}, \Lambda_q\}$$

where the eigenvalues in either Λ_{n-q} or Λ_q must be either real or complex conjugate, and the dimensions of these two matrices are $n - q$ and q, respectively.

Step 1: Compute the $(n-q) \times n$-dimensional solution matrix T_{n-q} of the equation

$$T_{n-q}A - \Lambda_{n-q}T_{n-q} = L\overline{C} \tag{8.10a}$$

and

$$\mathrm{rank} \quad [T'_{n-q} : \overline{C}']' = n \tag{8.10b}$$

Because (8.10a) is the same as (4.1) when the dimension is $n - q$, and because the matrix F of (4.1) is also set in Jordan form in Algorithm 5.3, Step 1 can be the same as Steps 1 and 2 of Algorithm 5.3.

Because the above two equations are the necessary and sufficient conditions of the well-known state observer (See Sec. 4.1), the solution T_{n-q} of these two equations always exists for observable systems, and is nonunique if $q > 1$.

Because (8.9a), (8.10a) and the Λ of Step 0 show that T_{n-q} would be the left eigenvector matrix of the feedback system dynamic matrix corresponding to Λ_{n-q}, it is desirable to make its rows as linearly independent as possible (see Sec. 2.2).

Step 2 of Algorithm 5.3 can use the numerical algorithms of Subsection 8.2.1 to make the rows of matrix $[T'_{n-q} : \overline{C}']'$ as linearly independent as possible.

If $q = n$ (state feedback case), then Step 1 is unnecessary and $T_{n-q} = 0$.

Step 2: Compute the $n \times q$ dimensional and full-column rank solution matrix V_q of

$$AV_q - V_q\Lambda_q = BK_q \tag{8.11a}$$

and

$$T_{n-q}V_q = 0 \tag{8.11b}$$

If the i-th eigenvalue in matrix Λ_q is a distinct and real number λ_i, then this equation pair is equivalent of

$$\begin{bmatrix} A - \lambda_i I & -B \\ T_{n-q} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_i \\ \mathbf{k}_i \end{bmatrix} = 0 \tag{8.12}$$

where $\mathbf{v}_i$ and $\mathbf{k}_i$ are the i-th column of matrices V_q and K_q respectively corresponding to λ_i.

The equation pair (8.11) together with (8.10) obviously imply that (8.9) is fully determined and satisfied in the sense that $\Lambda = \text{diag}\{\Lambda_{n-q}, \Lambda_q\}$, the first $n - q$ left eigenvectors of T will be formed by T_{n-q}, and the last q right eigenvectors of V will be formed by V_q, when $\overline{K}$ of (8.9) is computed from K_q of (8.11a) by an appropriate similarity transformation (as will be done in Step 3 of this algorithm).

Because Step 1 of this algorithm is the same as state observer design, Step 2 is the only nontrivial step of Algorithm 8.1.

The similarity between (8.2) and (8.11a) indicates that the solution of (8.11a) can be computed generally using (8.3), while the remaining freedom of (8.11a) can be used to satisfy the set of linear equation (8.11b).

It is very interesting to notice that the equation pair (8.11) corresponding to a different system $(A, B, C \underset{=}{\triangle} T_{n-q})$, is exactly dual to the matrix equation pair (4.1) and (4.3) of order q and corresponding to system (A, B, C). This duality is more clearly revealed by comparing (8.12) with (6.7). In other words, the corresponding dimension m of (6.7) is now $n - q$ for (8.12), and the corresponding condition $m > p$ of (6.7) (see Conclusion 6.1) is now $p >$ new m $(= n - q)$ for (8.12). This new condition is equivalent of $q + p > n$ [Tsui, 2000].

The solution of (4.1) is presented in Algorithm 5.3 (Chap. 5) and the corresponding solution of (4.3) is presented in Algorithm 6.1 (Secs 6.1 and 6.2). From Conclusion 6.1 of Sec. 6.2 and the dimension of (8.11) at the previous paragraph, exact solution of (8.11) exists if and only if either $q+p>n$ [Kimura, 1975] or the eigenvalues of Λ_q are the transmission zeros of system $(A, B, C \underset{=}{\triangle} T_{n-q})$.

If $p+q>n+1$, then the solution of (8.11) is not unique. This freedom can be considered as the freedom of assigning the right eigenvectors of V_q of (8.11a), and is expressed in the form of linear combination of the basis vectors as in (8.5).

This result is compatible to its special case—state feedback case where $q=n$. It is clear that if $q=n$, then $q+p>n$ is guaranteed (arbitrary eigenvalue assignment is guaranteed), and then $p>1$ guarantees $q+p>n+1$ (eigenvector assignment of V_q is possible).

Step 3: The comparison between (8.9b) and (8.11a) shows that

$$\overline{K} = K_q(\overline{C}V_q)^{-1} \tag{8.13}$$

The inverse of matrix $\overline{C}V_q$ is guaranteed because $[T'_{n-q}\overline{C}']'$ is full-row rank (see Step 1) and because of (8.11b) $(T_{n-q}V_q = 0)$.

From the end of Step 2, there is freedom of assigning matrix V_q if $q+p>n+1$. Because T_{n-q} and V_q will be formed respectively by the first $n-q$ left eigenvectors of T and the last q right eigenvectors of V of the feedback system dynamic matrix of (8.9), and because matrix $[T'_{n-q}, \overline{C}]'$ is full-row rank, to make the actual eigenvector matrices T and V as well conditioned as possible so that the eigenvalues can be as robust as possible (Sec. 2.2), V_q may be assigned such that matrix $\overline{C}V_q$ is as well conditioned as possible. The most systematic and effective numerical algorithm for this assignment is presented in Subsection 8.2.1. (such as Algorithm 8.3).

Algorithm 8.1 is uniquely simple, analytical, and reveal the duality property of the original problem (8.9) in Steps 1 and 2. Hence its dual version is directly available as follows:

Step 0 : Divide the n eigenvalues into $\Lambda = \text{diag}\{\Lambda_{n-p}, \Lambda_p\}$ (8.14)

Step 1 : Find V_{n-p} such that $AV_{n-p} - V_{n-p}\,\Lambda_{n-p} = BK_{n-p}$ and
$\text{rank}[B : V_{n-p}] = n$ (8.15)

Step 2 : Find T_p such that $T_pA - \Lambda_pT_p = L_p\overline{C}$ and $T_pV_{n-p} = 0$ (8.16)

Step 3 : $\overline{K} = (T_pB)^{-1}L_p$ (8.17)

Because parameters p and q can be different from each other, the two dual versions of Algorithm 8.1 can complement each other.

For example, if $p+q=n+1$, then neither q nor p can be reduced to make an even q and p from the originally odd q and p (or neither the rows of matrix $\overline{C}$ nor the columns of matrix B can be ignored) for arbitrary eigenvalue assignment. Now if all assigned eigenvalues are in complex conjugate pairs and q is odd (a very common situation), then Step 0 of Algorithm 8.1 cannot be implemented. This difficulty has been studied for years since [Kim, 1975] without simple solution [Fletcher and Magni, 1987; Magni, 1987; Rosenthal and Wang, 1992].

However, because in this very situation both $n-p$ and p are even, the above dual version of Algorithm 8.1 can be applied to solve this problem without a hitch.

Example 8.1

$$\text{Let}(A,B,\overline{C}) = \left(\begin{bmatrix} -4 & 0 & -2 \\ 0 & 0 & 1 \\ 1 & -1 & -2 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 0 & -2 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)$$

Let the assigned eigenvalues be $-1, -2$, and -3. Compute matrix $\overline{K}$ so that matrix $A - B\overline{KC}$ has these eigenvalues, using Algorithm 8.1.

Step 0: We arbitrarily select $\Lambda_{n-q} = -3$ and $\Lambda_q = \text{diag}\{-2, -1\}$

Step 1: The q $(=2)$ basis vectors of T_{n-q} are $D_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

Any linear combination of D_1 would make the first column of (4.1) equal 0 for all L. The free matrix L of (4.1) can be used to satisfy the remaining two columns of (4.1) from any T_{n-q}, but will not be used in the subsequent steps of this algorithm. We arbitrarily select $\mathbf{c}_1 = [1 \quad 0]$ so that $T_{n-q} = \mathbf{c}_1 D_1 = [1 \quad 0 \quad 1]$ is linearly independent of the rows of matrix $\overline{C}$.

Step 2: Because $p+q=n+1$, the solution of Step 2 is unique and can be computed based on Eq. (8.12). In other words, Eq. (8.12) is used twice for the two eigenvalues -2 and -1 and their corresponding columns of matrices V_q and K_q. The solution is

$$V_q = \begin{bmatrix} 0 & 1 \\ -1 & 3 \\ 0 & -1 \end{bmatrix} \qquad \text{and} \qquad K_q = \begin{bmatrix} -1/2 & 1/4 \\ 1 & -1 \end{bmatrix}$$

It can be verified that both (8.11a) and (8.11b) are satisfied.

Step 3: From Eq. (8.13), $\overline{K} = K_q(\overline{C}V_q)^{-1} = \begin{matrix} 1/2 & 5/4 \\ -1 & -2 \end{matrix}$

The corresponding matrix $A - B\overline{K}_y\overline{C}$ has the desired eigenvalues.

Example 8.2 (the dual version of Algorithm 8.1 on the same problem)

Step 0: According to (8.14), we similarly select $\Lambda_{n-p} = -3$ and $\Lambda_p = \text{diag}\{-2, -1\}$

Step 1: The p $(=2)$ basis (column) vectors of solution $[V'_{n-p} : K'_{n-p}]'$ of the first equation of (8.15) are

$$D_1 = \begin{bmatrix} 4 & -3 \\ 1 & 0 \\ -3 & 2 \\ \dots & \dots \\ -1/2 & -1/4 \\ 0 & 1 \end{bmatrix}$$

Any linear combination of D_1 would satisfy the first equation of (8.15). We arbitrarily select $\mathbf{c}_1 = [\,1 \quad 1\,]'$ so that $V_{n-p} = D_1$ (the first three rows) $\mathbf{c}_1 = [\,1 \quad 1 \quad -1\,]'$ so that the second equation of (8.15) is also satisfied. Matrix K_{n-p} is not needed in the subsequent steps of the algorithm.

Step 2: Because $p + q = n + 1$, the solution of (8.16) is unique and can be computed by using Eq. (6.7) [which is equivalent of (8.16) if the matrix B of (6.7) is replaced by matrix V_{n-p}] twice. The solution is:

$$T_p = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \end{bmatrix} \quad \text{and} \quad L_p = \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix}$$

It can be verified that (8.16) is satisfied.

Step 3: $\overline{K} = (T_pB)^{-1}L_p = \begin{bmatrix} 1/2 & 5/4 \\ -1 & -2 \end{bmatrix}$ according to (8.17)

which is the same as the $\overline{K}$ of the basic algorithm version of Example 8.1.

8.1.4 Adjustment of Generalized State Feedback Control Design Priority and Procedure

Algorithm 8.1 assigns the n arbitrarily and previously chosen poles to the system exactly if $q + p > n$. The first group of $n - q$ eigenvalues can always be assigned exactly and their corresponding $n - q$ left eigenvectors T_{n-q} always have q basis vectors for each. These eigenvectors are proposed to be assigned so that the rows of matrix $[T'_{n-q}\overline{C}']'$ are as linearly independent as possible. The second group of q eigenvalues can be assigned exactly if and only if either $q + p > n$ or these eigenvalues are the transmission zeros of system (A, B, T_{n-q}). If $q + p > n + 1$, then there are $q + p - n$ basis vectors for each of the corresponding q right eigenvectors V_q, and these eigenvectors are proposed to be assigned so that matrix $\overline{C}V_q$ is as well conditioned as possible.

However in practice, many different situations and different requirements may arise that demand the above design procedure be adjusted accordingly.

First, if $q + p \leqslant n$ yet $q \times p > n$, then Algorithm 8.1 may not yield exact solution, yet arbitrarily given poles can be exactly assigned generically [Wang, 1996], although the design procedure of Wang [1996] is very complicated.

Second, even if $q \times p \leqslant n$ and exact assignment of arbitrarily given poles is not even generically possible [Wang, 1996], it is desirable and it should be in many cases possible based on Algorithm 8.1 to assign the second group of q poles approximately to desirable areas. This is achieved while the first group of $n - q$ poles are still exactly assigned by Algorithm 8.1. This level of pole assignment should be good enough in practice.

Third, because the two groups of eigenvalue/vectors are treated very differently in Algorithm 8.1—the first group has much higher priority, it is useful to try different groupings among the assigned eigenvalues and their eigenvectors.

Fourth, unlike the static output feedback case where all m rows of matrix $\overline{C}(\triangleq C)$ are corresponding to direct system output measurements, $q - m$ rows among the q rows of our matrix $\overline{C}$ are corresponding to the converged estimates of the linear combinations of system states. Therefore these $q - m$ rows of $\overline{C}$ can not be treated indifferently from the rest m rows (of matrix C) of matrix $\overline{C}$.

Fifth and finally, in some practical situations it is more desirable to minimize the system zero-input response with some prior knowledge of system initial state, than to make the eigenvectors as linearly independent as possible.

Each of these five different considerations is addressed by each of the following proposed adjustments of Algorithm 8.1. These adjustments are possible because Algorithm 8.1 is uniquely simple, analytical, and explicit.

Adjustment 1: Instead of designing T_{n-q} in Step 1 for the maximized angles between the rows of $[T'_{n-q} : \overline{C}']'$, it will be designed so that the arbitrarily given q eigenvalues of Λ_q are the q transmission zeros of system triple (A, B, T_{n-q}).

Based on the first of the above five considerations, this adjustment should be applied when $p+q \leqslant n$ since otherwise the arbitrary pole assignment is already guaranteed, and should be executable if $q \times p > n$, because arbitrary pole assignment is generically possible if $q \times p > n$ [Wang, 1996]. Comparing the algorithm of Wang [1996], the computation of this adjustment of Algorithm 8.1 is obviously much simpler. Besides, the algorithm of Wang [1996] considered the pole assignment only (not the eigenvector assignment).

This adjustment may not yield result if $q \times p \leqslant n$ because under this condition arbitrary pole assignment is impossible [Wang, 1996].

Example 8.3 below demonstrated this adjustment.

Adjustment 2: Instead of designing T_{n-q} in Step 1 for the maximized angles between the rows of $[T'_{n-q} : \overline{C}']'$, it will be designed so that there are q transmission zeros of system triple (A, B, T_{n-q}) in desirable proximity locations.

This deviation from the priority of exact pole assignment, which is prevalent for forty years until today, is actually quite practical. First, there is no generally optimal and precise pole selection (see Subsection 8.1.1). Secondly, the other parameters of the matrix such as the conditions of the eigenvectors, which determine the sensitivity of the poles, can be as important as the poles themselves.

Because assigning proximity transmission zeros is conceivably easier than assigning precise transmission zeros, this adjustment can be applied for some open-loop systems with $p \times q \leqslant n$, even though exact pole assignment is impossible under these conditions and for arbitrarily given poles [Wang, 1996]. For example, stabilization (assign the q transmission zeros in the open left half plane) should be possible in many cases even if $p \times q \leqslant n$.

Because the requirement of proximity pole assignment is more vague than that of precise pole assignment, the precise sufficient condition in terms of parameters $\{n, p, q\}$ for this assignment may not exist. A practical and high quality feedback control system design that requires the guidance of advanced control theory should consider not just stabilization, but also high performance and robustness.

Adjustment 3: As stated in the beginning of this subsection and in the third of the above five considerations, the two groups of $n - q$ (or $n - p$) and then q (or p) poles are treated really differently by our design algorithm. Therefore, the different grouping of the n poles into these two groups can really make a difference.

Conceivably, One should place the more dominant and more critical poles into the first group. This kind of considerations of groupings of the system poles is similar to all three eigenvector assignment procedures of Sec. 8.2.

This design adjustment is demonstrated by Example 8.4 below, which showed a quite improved design solution which is based on a different pole grouping.

Adjustment 4: As stated in the fourth of the above five considerations or in Sec. 6.3, our matrix $\overline{C}$ has uniquely two components. One component is matrix C which is corresponding to the direct system output $[= C\mathbf{x}(t)]$ measurement. The second component is matrix T (not the same T of this subsection) which is corresponding to a converged estimation of $T\mathbf{x}(t)$. Thus these two component matrices of $\overline{C}$ should be treated differently in the design.

Matrix $\overline{C}$ appeared mainly in Step 2 of the design algorithm where the q right eigenvectors V_q are assigned to make matrix $\overline{C}V_q$ as well conditioned as possible. This assignment should consider the difference between the rows of component matrices C and T in matrix $\overline{C}$. For example, the weighting on the vectors of C may be higher than that on the vectors of T (see Algorithm 8.3). This design adjustment can be applied when there is design freedom for V_q (or when $p + q > n + 1$).

Adjustment 5: Until now the first $n - q$ left eigenvectors are assigned either for maximized angles of the rows of matrix $[T'_{n-q} : \overline{C}']'$ if $q + p > n$, or for pole assignment (if $p + q \leqslant n$), in Design Adjustments 1 and 2 above. However, the following possible different goal of eigenvector assignment can also be considered.

Because the zero-input response of the feedback system state can be stated as $V\,\mathrm{e}^{\Lambda t} T\mathbf{x}(0)$, and because it is often useful to minimize the zero-input response, a useful goal is to assign this T (the same T of this subsection) such that $T\mathbf{x}(0)$ is minimized [if $\mathbf{x}(0)$ is known]. Although this goal of eigenvector assignment was proposed before, that proposition was for state feedback design case only.

This design adjustment is demonstrated by Example 8.5 below.

Example 8.3: Adjustment 1

$$\text{Let}(A, B, \overline{C}) = \left(\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \right)$$

and let the assigned poles be $\{-1, -2, -3, -4\}$.

Because $p + q$ is not greater than n, we will apply Design Adjustment 1 above to assign these four poles precisely.

Step 0: $\Lambda_{n-q} = \text{diag}\{-1, -2\}$, and $\Lambda_q = \text{diag}\{-3, -4\}$

Step 1: The q $(= 2)$ basis vectors for each of the $n - q$ $(= 2)$ rows of matrix T_{n-q} are

$$D_1 = \begin{bmatrix} u & 0 & -1 & 1 \\ v & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad D_2 = \begin{bmatrix} x & 3 & -2 & 1 \\ y & 0 & 0 & 0 \end{bmatrix}$$

where $\{u, v, x, y\}$ can be arbitrary. Any linear combination of the rows of D_1 and D_2 would make the last two columns of the corresponding (4.1) equal 0 for all L. The remaining first two columns of (4.1) can be satisfied by the free matrix L of (4.1).

Because $p + q$ is not greater than n, we will select the linear combinations of D_1 and D_2 so that the remaining two eigenvalues -3 and -4 are the transmission zeros of system (A, B, T_{n-q}). Because $p \times q$ is not greater than n, the solution may not exist. Fortunately, the solution exists in this example as $\mathbf{c}_1 = \mathbf{c}_2 = [1 \quad 0]$ and $u = 60$ and $x = 84$. The corresponding

$$T_{n-q} = \begin{bmatrix} \mathbf{c}_1 D_1 \\ \mathbf{c}_2 D_2 \end{bmatrix} = \begin{bmatrix} 60 & 0 & -1 & 1 \\ 84 & 3 & -2 & 1 \end{bmatrix}$$

and is linearly independent of the rows of matrix $\overline{C}$.

Step 2: The design of Step 1 guarantees the unique solution of this step as

$$V_q = \begin{bmatrix} 1 & 1 \\ -3 & -4 \\ 15 & 12 \\ -45 & -48 \end{bmatrix} \quad \text{and} \quad K_q = \begin{bmatrix} -6 & 4 \\ 119 & 179 \end{bmatrix}$$

Step 3: From Eq. (8.13), $\overline{K} = K_q(\overline{C}V_q)^{-1} = \begin{bmatrix} 36 & 10 \\ 61 & 60 \end{bmatrix}$

The corresponding matrix $A - B\overline{KC}$ has the desired eigenvalues.

Example 8.4: Adjustment 3

Let system $(A, B, \overline{C})$ and the assigned three eigenvalues be the same as that of Examples 8.1 and 8.2.

Step 0: We select $\Lambda_{n-q} = -1$ and $\Lambda_q = \text{diag}\{-2, -3\}$, which is different from the result of Step 0 of Examples 8.1 and 8.2.

Step 1: Equation (4.1) implies that

$$\begin{array}{c} \left[\begin{array}{cccccc} 1 & 0 & 3 & : & -3 & -5 \\ 0 & 1 & 0 & : & 1 & 1 \end{array}\right] \\ \begin{array}{cc} D_1 & \qquad\qquad E_1 \end{array} \end{array} \begin{bmatrix} -3 & 0 & -2 \\ 0 & 1 & 1 \\ 1 & -1 & -1 \\ \hline 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} = 0$$

We arbitrarily select $\mathbf{c}_1 = [1 \ -2]$ so that $T_{n-q} = \mathbf{c}_1 D_1 = [1 \ -2 \ \ 3]$ and matrix $[T'_{n-q} \overline{C}']'$ is rank n. The first column of both sides of (4.1) equals 0 for this T_{n-q} and for all L. The free matrix L can be used to satisfy the remaining two columns of (4.1).

Step 2: Because $p + q = n + 1$, the solution of Step 2 is unique:

$$V_q = \begin{bmatrix} 2 & -9 \\ -5 & 3 \\ -4 & 5 \end{bmatrix} \quad \text{and} \quad K_q = \begin{bmatrix} -5/2 & 13/4 \\ 7 & -7 \end{bmatrix}$$

It can be verified that both (8.11a) and (8.11b) are satisfied.

Step 3: From Eq. (8.13), $\overline{K} = K_q(\overline{C}V_q)^{-1} = \begin{bmatrix} -1/26 & 35/42 \\ -7/13 & -14/13 \end{bmatrix}$

The corresponding matrix $A - B\overline{KC}$ has the desired eigenvalues, as that of Examples 8.1 and 8.2. However, the feedback gain $\overline{K}$ is much smaller (and therefore much more robust and much better) in this example.

Example 8.5: Adjustment 5

Same system and the same assigned eigenvalues as Examples 8.1, 8.2, and 8.4.

If the initial state is known as $\mathbf{x}(0) = [0\ x\ 0]'$ where $x \neq 0$, then in Step 1, $D_1\mathbf{x}(0) = [0\ x]'$. To minimize $T_{n-q}\mathbf{x}(0)$, we will select $\mathbf{c}_1 = [1\ 0]$. The corresponding result of this adjustment is the same as that of Examples 8.1 and 8.2, and has the first row of $T\mathbf{x}(0)$ equal to 0.

On the other hand, if the initial state is known as $\mathbf{x}(0) = [x\ 2x\ x]'$ instead where $x \neq 0$, then in Step 1, $D_1\mathbf{x}(0) = [4x\ 2x]'$. To minimize $T_{n-q}\mathbf{x}(0)$, we will select $\mathbf{c}_1 = [1\ \ -2]$. The corresponding result of this adjustment is the same as that of Example 8.4, and has the first row of $T\mathbf{x}(0)$ equal to 0.

8.1.5 Conclusion

For the problem of assigning exactly n arbitrarily given eigenvalues to the dynamic matrix $A - B\overline{KC}$, the state feedback case [rank $(\overline{C}) = q = n$] has no restriction on system $(A, B, \overline{C})$, while the generalized state feedback case has the restriction of $q + p > n$ on system $(A, B, \overline{C})$.

However, even if $q + p \leqslant n$ but $q \times p > n$, Adjustment 1 of our design algorithm can still make the arbitrary pole assignment generically possible.

Because eigenvectors determine the sensitivity and robustness properties of their corresponding eigenvalues, different eigenvector assignments can make a substantial difference in the condition number of the corresponding eigenvalue assignment problem. As demonstrated by the different eigenvector assignment results of Examples 8.1 and 8.4.

The development of computationally reliable pole assignment algorithms has been substantial for state feedback case [Gopinath, 1971; Miminis and Paige, 1982; Petkov et al., 1986; Duan, 1993a] as well as for the static output feedback case (similar to generalized state feedback case) [Misra and Patel, 1989; Syrms and Lewis, 1993a]. But almost all of these algorithms do not discuss how to assign the eigenvectors (the eigenvalue assignment under more general conditions is already difficult enough [Wang, 1996]). Thus these algorithms cannot prevent a bad eigenvector assignment, which can make the corresponding eigenvalue assignment problem bad conditioned, and thus make the computation of this assignment unreliable in spite of a numerically stable algorithm.

Therefore eigenvector assignment is as important as eigenvalue assignment. All pole assignment algorithms of this book are such that the corresponding eigenvector assignment is in the form of assigning the linear combination coefficients of the basis vectors of these eigenvectors. This is a distinct advantage because the systematic and effective eigenvector assignment algorithms (Algorithms 8.2 and 8.3) of the next section are based entirely on this assignment formulation.

For the problem of eigenvector assignment assuming the eigenvalues are already assigned exactly, there is also much difference between the state feedback case and the generalized state feedback case. There are p basis vectors for each eigenvector in state feedback case. In the generalized state feedback case (Algorithm 8.1), there are q basis vectors for each of the first $n - q$ left eigenvectors, while there are only $q + p - n$ basis vectors for each of the remaining q right eigenvectors.

In addition to assigning the eigenvectors for the best possible condition, this subsection also proposed four different yet practical objectives of eigenvector assignment in Adjustments 1, 2, 3, and 5.

Finally, in addition to these distinct advantages on the generality of eigenvalue assignment (see the dual version and the Adjustments 1 and 2) and on the unique and explicit form of eigenvector assignment, the algorithm of this book is very simple in light of its tasks. This is fully demonstrated by the numerical examples. The result in *every* step of these five examples is explicit and in fraction form. This implies that the algorithm is very explicit, simple, and analytical.

Based on these explicit, simple, and analytical design algorithms, this book has opened several independent research directions on this very challenging and very effective eigenstructure assignment problem, especially in generalized state feedback case, as discussed in the five design adjustments of subsection 8.1.4.

8.2 EIGENVECTOR ASSIGNMENT

Eigenvectors are extremely important not only because they decide the sensitivities of their corresponding eigenvalues, but also because of the following important properties. From (2.2),

$$\mathbf{x}(t) = V\ \mathrm{e}^{\Lambda t} V^{-1}\mathbf{x}(0) + \int_0^t V\ \mathrm{e}^{\Lambda(t-\tau)} V^{-1} B\mathbf{u}(\tau)\, d\tau \tag{8.18}$$

From (8.1b) and (8.6),

$$K = K\,\hat{}\,V^{-1} = [B_1^{-1} : 0](A - V\Lambda V^{-1}) \tag{8.19}$$

Thus if Λ is assigned and $[A, B, \mathbf{x}(0)$, and $\mathbf{u}(\tau)]$ are given, then the dominant factor that finally decides the smoothness of response (8.18) (see also Example 2.1) and the magnitude of the feedback control gain K of (8.19) (see also Examples 8.1 and 8.4), is eigenvector matrix V.

From (8.5) there are $p \times n$ free parameters (in $\mathbf{c}_i$) available for eigenvector assignment after the eigenvalues are assigned. Thus for $p > 1$, the freedom of eigenvector assignment not only exists, but is also very significant.

Research on eigenvector assignment dates from the mid-1970s [Moore, 1976; Klein and Moore, 1977; Fahmy and O'Reilly, 1982; Van Dooren, 1981; Van Loan, 1984]. However, it was only in 1985 that eigenvector assigment freedom began to be expressed in terms of the basis vectors of each eigenvector, such as $\mathbf{c}_i D_i$ of (6.1) for left eigenvectors [Tsui, 1985] and $D_i\mathbf{c}_i$ of (8.5) for right eigenvectors [Kautsky et al., 1985]. Here the D_i matrices are already determined and the $\mathbf{c}_i$ vectors are completely free.

Although this is only a new expression of eigenvecvtor assignment freedom, it finally enabled the full use of this freedom in many important design applications (see Fig. 5.1).

This section discusses how to assign the eigenvectors so that the angles between these vectors are maximized, based on this new expression or formulation. Subsections 8.2.1 and 8.2.2 regard numerical methods [Kautsky et al., 1985] and analytical rules [Tsui, 1986a, 1993a], respectively.

For uniformity, the entire section is formulated as the computation of the p-dimensional column vectors $\mathbf{c}_i$ for the eigenvectors $D_i\mathbf{c}_i$ $(i = 1, \ldots, n)$, even though the D_i matrices computed from different applications [such as (6.1), (6.6), (8.10), (8.11), (8.15), and (8.16)] can have different dimensions.

8.2.1 Numerical Iteration Methods [Kautsky et al., 1985]

The single purpose of numerical eigenvector assignment methods is to maximize the angles between the eigenvectors. This purpose can also be interpreted as minimizing the condition number of eigenvector matrix $V, \kappa(V) \triangleq \|V\| \, \|V^{-1}\|$.

From (2.16), (2.24), (8.19), and (2.2), a smaller $\kappa(V)$ can generally imply higher robust performance, higher robust stability, lower control gain, and smoother response, respectively.

To simplify the computation, the methods of this subsection require that all p vectors $\mathbf{d}_{ij}$ $(j = 1, \ldots, p)$ in each matrix D_i $(i = 1, \ldots, n)$ be orthogonal and normalized, or $\mathbf{d}'_{ij}\mathbf{d}_{ik} = \delta_{jk}$ and $\|\mathbf{d}_{ij}\| = 1$ $\forall i$ and j. This requirement can be met by the following two ways.

The first way is to satisfy this requirement during the computation of D_i itself. For example, in the computation of (8.4), we first make the QR decomposition on the matrix:

$$[A - \lambda_i I : -B] = [R_i : 0]Q'_i, \qquad i = 1, \ldots, n \tag{8.20}$$

where Q_i is an $(n + p)$-dimensional unitary matrix. Then the D_i matrix of (8.4) is

$$D_i = [I_n : 0]Q_i \begin{bmatrix} 0 \\ I_p \end{bmatrix}, \qquad i = 1, \ldots, n \tag{8.21}$$

The second way is to compute matrices D_i first, and then update these matrices to satisfy this requirement. This second step can be accomplished by making the QR decomposition on each D_i:

$$D_i = Q_i R_i, \qquad i = 1, \ldots, n \tag{8.22a}$$

where Q_i is an n-dimensional unitary matrix. The D_i matrix can be updated as

$$D_i = Q_i \begin{bmatrix} I_p \\ 0 \end{bmatrix}, \qquad i = 1, \ldots, n \tag{8.22b}$$

which retains the same properties of the original D_i.

We will study two numerical methods named as Algorithms 8.2 and 8.3, respectively. The first method updates one vector per iteration, to maximize the angle between this vector and other $n - 1$ vectors. The second method updates two among a separate set (say, S) of n-orthonormal vectors at each iteration, to minimize the angles between these two vectors and their corresponding D_i's while maintaining the orthonormality of S. These two methods are named "rank-one" and "rank-two" methods, respectively, and work from quite opposite directions (but for the same ultimate purpose).

Algorithm 8.2

Rank-one method of eigenvector assignment [Kautsky et al., 1985]

Step 1: Let $j = 0$. Set arbitrarily an initial set of n vectors

$$\mathbf{v}_i = D_i\mathbf{c}_i, \qquad i = 1, \ldots, n \; (\|\mathbf{c}_i\| = 1 \; \forall i)$$

Step 2: Let $j = j + 1$. Select a vector $\mathbf{v}_j$ for updating. Then set the $n \times (n-1)$ dimensional corresponding matrix

$$V_j = [\mathbf{v}_1 : \ldots : \mathbf{v}_{j-1} : \mathbf{v}_{j+1} : \ldots : \mathbf{v}_n]$$

Step 3: Make QR upper triangularization of V_j:

$$V_j = Q_j R_j = [\overline{Q}_j : \mathbf{q}_j] \underset{n-1}{\begin{bmatrix} \overline{R}_j \\ 0 \end{bmatrix}}$$

where Q_j and $\overline{R}_j$ are n-dimensional unitary and $(n-1)$-dimensional upper triangular matrices, respectively. Hence $\mathbf{q}_j$ $(\|\mathbf{q}_j\| = 1)$ is orthogonal to $R(V_j)$ because $\mathbf{q}_j' V_j = 0$.

Step 4: Compute the normalized least-square solution $\mathbf{c}_j$ of $D_j\mathbf{c}_j = \mathbf{q}_j$ or the projection of $\mathbf{q}_j$ on D_j: (see Example A.8 or Golub and Van Loan, 1989)

$$\mathbf{c}_j = \frac{D_j'\mathbf{q}_j}{\|D_j'\mathbf{q}_j\|} \tag{8.23}$$

Step 5: Update vector

$$\mathbf{v}_j = D_j\mathbf{c}_j = \frac{D_j D_j'\mathbf{q}_j}{\|D_j'\mathbf{q}_j\|} \tag{8.24}$$

Step 6: Check the condition number of $V = [\mathbf{v}_1 : \ldots : \mathbf{v}_n]$. Stop the iteration if satisfactory. Otherwise go to Step 2 for another iteration.

It is a normal practice to stop when all n vectors are updated, or when index j equals n at Step 6.

At Step 3 the QR decomposition may not be performed from the start to finish on matrix V_j, but may be obtained by updating the previous QR decomposition result (on matrix V_{j-1}). The computation of this update is of order n^2 [Kautsky et al., 1985], which is substantially lower than $2n^3/3$ of the normal QR decomposition (see Appendix A, Sec. A.2). However, such an updating algorithm has not appeared in the literature.

Based on experience, Kautsky et al. [1985] points that the first sweep of n vectors of Algorithm 8.2 is very effective in lowering $\kappa(V)$, but the algorithm cannot guarantee the convergence to the minimal $\kappa(V)$. This is because the maximization of the angle between *one* eigenvector to the others cannot guarantee the maximization of *all* angles between the n eigenvectors.

Tits and Yang [1996] claim that each of the above iterations can increase the determinant of matrix $V, |V|$, and the whole algorithm can converge to a locally maximum $|V|$ depending on the initial value of V at Step 1.

Algorithm 8.2 is also extended to the complex conjugate eigenvalue case by Tits and Yang [1996], using complex arithmetic operations. To use real arithmetic operations, the results corresponding to complex conjugate eigenvalues in Algorithm 5.3 (Step 1) and in (8.3) can be used.

Algorithm 8.3

Rank-two method of eigenvector assignment [Kautsky et al., 1985; Method 3, 4; Chu, 1993b]

Step 1: Select a set of orthonormal vectors $\mathbf{x}_i$, $i = 1, \ldots, n$. An example of such a set is $[\mathbf{x}_1 : \ldots : \mathbf{x}_n] = I$. Compute the basis vector matrix $\overline{D}_i$ which forms the complement space of D_i, $i = 1, \ldots, n$.

$$\overline{D}_i = Q_i \begin{bmatrix} 0 \\ I_{n-p} \end{bmatrix}$$

where

$$D_i = Q_i \begin{bmatrix} R_i \\ 0 \end{bmatrix}$$

for $i = 1, \ldots, n$. Minimizing angles between $\mathbf{x}_i$ and D_i is equivalent to maximizing angles between $\mathbf{x}_i$ and $\overline{D}_i$, $i = 1, \ldots, n$.

Step 2: Select two vectors $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$ among the n vectors. Rotate and update these two vectors by an angle θ such that

$$[\overline{\mathbf{x}}_j : \overline{\mathbf{x}}_{j+1}] = [\mathbf{x}_j : \mathbf{x}_{j+1}] \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{8.24}$$

and such that the angle ϕ_j between $\overline{\mathbf{x}}_j$ and $\overline{D}_j$ and the angle ϕ_{j+1} between $\overline{\mathbf{x}}_{j+1}$ and $\overline{D}_{j+1}$ are maximized. This is expressed in terms of θ as

$$\begin{aligned} & \min_\theta \{ r_j^2 \cos^2 \phi_j + r_{j+1}^2 \cos^2 \phi_{j+1} \} \\ & = \min_\theta \{ r_j^2 \| \overline{D}_j' \overline{\mathbf{x}}_j \|^2 + r_{j+1}^2 \| \overline{D}_{j+1}' \overline{\mathbf{x}}_{j+1} \|^2 \} \\ & = \min_\theta \{ c_1 \sin^2 \theta + c_2 \cos^2 \theta + c_3 \sin\theta \cos\theta \} && (8.25) \\ & = \min_\theta \{ \mathrm{f}(\theta) \} && (8.26) \end{aligned}$$

where in (8.25),

$$\left. \begin{aligned} c_1 &= r_j^2 \mathbf{x}_j' \overline{D}_{j+1} \overline{D}_{j+1}' \mathbf{x}_j + r_{j+1}^2 \mathbf{x}_{j+1}' \overline{D}_j \overline{D}_j' \mathbf{x}_{j+1} \\ c_2 &= r_j^2 \mathbf{x}_j' \overline{D}_j \overline{D}_j' \mathbf{x}_j + r_{j+1}^2 \mathbf{x}_{j+1}' \overline{D}_{j+1} \overline{D}_{j+1}' \mathbf{x}_{j+1} \\ c_3 &= 2\mathbf{x}_j' (r_{j+1}^2 \overline{D}_{j+1} \overline{D}_{j+1}' - r_j^2 \overline{D}_j \overline{D}_j') \mathbf{x}_{j+1} \end{aligned} \right\} \tag{8.27}$$

and r_i are weighting factors to ϕ_i $(i = j,\ j+1)$. For example, the optimization on robust stability measure M_3 of (2.25) requires that

$$r_i = |Re\{\lambda_i)|^{-1}, \qquad i = j,\ j+1$$

The function $f(\theta)$ of (8.26) is positive, continuous and periodic, and has a global minimum. The examination of $f(\theta)$ of (8.25) shows that if $c_3 = 0$ or if $c_1 = c_2$, then (8.25) is at its minimum when $\theta = 0$.

For $c_3 \neq 0$ and $c_1 \neq c_2$, the nonzero values of θ can be determined by setting the derivative of $f(\theta)$ (with respect to θ) to zero:

$$f'(\theta) = (c_1 - c_2)\sin(2\theta) + c_3\cos(2\theta) = 0$$

or

$$\frac{c_3}{c_2 - c_1} = \tan(2\theta) = \frac{2\tan\theta}{1 - \tan^2\theta} \tag{8.28}$$

or

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{c_3}{c_2 - c_1}\right) + k\pi \tag{8.29}$$

where $k = 0, \pm 1, \pm 2, \ldots$
Integer k of (8.29) must also make the corresponding θ satisfy

$$f''(\theta) = 2[(c_1 - c_2)\cos(2\theta) - c_3\sin(2\theta)] > 0$$

or

$$\tan(2\theta) < \frac{c_1 - c_2}{c_3} \tag{8.30}$$

Instead of (8.29), which computes θ from the first equality of (8.28) or from 2θ, there is a more accurate formula for θ, which is derived from the second equality of (8.28) such that

$$\theta = \tan^{-1}\left[\frac{-1 + (1 + c_4)^{1/2}}{c_4}\right] + k\pi \tag{8.31a}$$

where

$$c_4 = \frac{c_3}{c_2 - c_1} \tag{8.31b}$$

After θ is determined from either (8.29) or (8.31) with (8.30) guaranteed, (8.24) is the last computation of Step 2.

Step 3: If the value θ of Step 2 is close to 0 or $k\pi$ (k is an integer), then $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$ are already near the linear combination of D_j and D_{j+1}.

If this is not true for all j, then go back to Step 2 for more iteration. Otherwise, find the projections of all n-updated vectors $\mathbf{x}_i$ on the $\mathbf{R}(D_i)$ $(i = 1, \ldots, n)$, or

$$\mathbf{v}_i = \frac{D_i(D_i'\mathbf{x}_i)}{\|D_i'\mathbf{x}_i\|}, \qquad i = 1, \ldots, n$$

The critical step of Algorithm 8.3 is obviously Step 2, which has not appeared in the literature either. This version is based on and revised from Chu [1993b].

According to Kautsky et al. [1985], the order of computation is similar in each update of Algorithms 8.2 (Step 3) and 8.3 (Step 2). The order of computation of (8.27) is 4pn (four pairs of $\mathbf{x}_i'\overline{D}_k$) which should constitute the main computation of Step 2 of Algorithm 8.3, while the simplified computation of Step 3 (Algorithm 8.2) is of order n^2.

Also according to Kautsky et al. [1985], Algorithm 8.3 requires less iteration and is more efficient than Algorithm 8.2, for well-conditioned problems. This is understandable because Algorithm 8.3 starts with an ideal orthonormal solution and then makes it approach the actual solution, while Algorithm 8.2 starts with an arbitrary solution and then makes it approach orthonormal. However, for ill-conditioned problems, both Algorithms 8.2 and 8.3 cannot yield reliable results [Kautsky et al., 1985]. In such a case we may use the analytical rules of Subsection 8.2.2 or the "Method 1" of Kautsky et al. [1985], but the latter can be very complicated.

Although Algorithms 8.2 and 8.3 cannot guarantee convergence for ill-conditioned problems, they are still very popular among researchers because of their relative simplicity as compared to Method 1 of Kautsky et al. [1985], and they have already been made into CAD software [MATLAB, 1990].

An advantage of Algorithm 8.3 over Algorithm 8.2 is that the former can consider the weighting factors r_i, while the latter cannot. Thus a direction of improvement for Algorithm 8.2 is to incorporate weightings into its updating procedure. For example, the eigenvectors corresponding to more dominant eigenvalues (see Subsection 2.2.2) should be updated first and be updated more times, instead of being treated indifferently from less critical eigenvectors as in the current version of Algorithm 8.2.

Algorithm 8.3 could also be improved by the additional arrangement of the combination pairs of $\mathbf{x}_i$ and D_i at Step 1. The current combination pairs between $\mathbf{x}_i$ and D_i $(i = 1, \ldots, n)$ are arbitrarily made. However, in this arbitrary initial combination, the angle between $\mathbf{x}_i$ and D_i may be large, while the angle between $\mathbf{x}_i$ and $D_j(\, j \neq i)$ may be small. Thus a more reasonable initial arrangement should pair $\mathbf{x}_i$ with D_j together instead of with D_i.

Consideration of the analytical information of eigenvalues and controllability indexes in eigenvector assignment, is a feature of analytical eigenvector assignment rules discussed in the next subsection.

8.2.2 Analytical Decoupling Method

Numerical eigenvector assignment methods are aimed at maximizing the angles between the feedback system eigenvectors, or the minimization of the condition number of the eigenvector matrix $\kappa(V)$.

However, $\kappa(V)$ may not be generally accurate in indicating individual eigenvalue sensitivity and system robust stability (see Sec. 2.2). In addition, numerical methods often overlook some critical and analytical system parameters and properties such as eigenvalues, controllability indices, and decoupling. From Examples 2.4 and 2.5 and their analysis, decoupling is very effective in eigenvalue sensitivity and robust stability.

Analytical eigenvector assignment discussed in this subsection is based on decoupling. This assignment is also based substantially on the analytical properties of eigenvalues and controllability indices $(\mu_j, j = 1, \ldots, p)$. However, this assignment cannot claim the sharp numerical property of a minimized $\kappa(V)$.

The analytical eigenvector assignment is also based on the block-controllable Hessenberg form of system matrices (8.6), because this form reveals the information of controllability indices. Three properties should be noticed based on this form.

First, the feedback system eigenvectors (and their basis vectors) are computed from only the lower $n - p$ rows of matrix A and the feedback system poles (see Step 1 of Algorithm 5.3 for the dual case).

Second, the feedback system eigenvectors are determined independent of and prior to the feedback gain K, which can affect only the upper p rows of matrix A and the upper p rows of matrix $A - BK$ (see Step 3 of Algorithm 5.3 for the dual case).

Third and finally, if the basis vectors of the feedback system eigenvectors are computed by back substitution operation, then each of these basis vectors can be identified with one (say the j-th) of the p inputs of the system (see Conclusion 5.1 and Example 5.5 for the dual case).

Let us first analyze some analytical properties of these basis vectors if they are computed by back substitution and based on the form (8.6) of the system matrices. This analysis is also based on the assumption of $\mu = \mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_p$ for simplicity of presentation. This assumption can always be lifted because the sequence of system inputs can always be altered.

Now for a fixed eigenvalue λ_i and its corresponding eigenvector, each of the p corresponding basis vectors, $\mathbf{d}_{ij}, j = 1, \ldots, p$, can be expressed as [Tsui, 1987a,b, 1993a]

$$\mathbf{d}_{ij} = \left[\begin{array}{ccccccc|l}
x & : & \ldots & : & x & : & x & \\
: & : & \ldots & : & : & : & : & \\
: & : & \ldots & : & : & : & x & \\
: & : & \ldots & : & : & : & * & \}p_1 \\
: & : & \ldots & : & : & : & 0 & \\
: & : & \ldots & : & : & : & : & \\
x & : & \ldots & : & x & : & 0 & \\
 & & & & & & & \\
x & : & \ldots & : & x & : & & \\
: & : & \ldots & : & : & : & & \\
: & : & \ldots & : & x & : & & \\
: & : & \ldots & : & * & : & & \}p_2 \\
: & : & \ldots & : & 0 & : & & \\
: & : & \ldots & : & : & : & & \\
x & : & \ldots & : & 0 & : & & \\
 & & & & & & & \\
 & : & & & & & & : \\
 & & & & & & & \\
x & : & & & & & & \\
: & : & & & & & & \\
x & : & & & & & & \\
* & : & & & & & & \}p_{\mu j} \\
0 & : & & & & & & \\
: & : & & & & & & \\
0 & : & & & & & & \\
 & & & & & & & \\
0 & & & & & & & \\
: & & & & & & & \\
0 & & & & & & &
\end{array}\right]
\begin{bmatrix} 1 \\ \lambda_i \\ : \\ : \\ \lambda_i^{\mu j-1} \end{bmatrix} \triangleq U_j \mathbf{v}_{ij} \tag{8.32}$$

In the matrix U_j above, the "x" entries are arbitrary elements, and the "*" entries are nonzero elements and are located at the j-th position from top down of each block, and the block size p_k $(k = 1, \ldots, \mu)$ indicates the number of controllability indices that are no smaller than k (see Definition 5.1 for the dual case). In addition, matrix U_j is determined only by the lower $n - p$ rows of matrix A, and is independent of λ_i. Hence matrix U_j can be considered a "coefficient matrix" of the j-th basis vector $\mathbf{d}_{ij}$ for any λ_i.

The partition of $\mathbf{d}_{ij}$ in (8.32) is for analysis only. In actual design computation the $\mathbf{d}_{ij}$ can be computed directly by back-substitution.

Example 8.6

Let $\mu_1 = 4, \mu_2 = 2, \mu_3 = 2$, and $\mu_4 = 1$. Then from (8.32),

$$\mathbf{d}_{i1} = \begin{bmatrix} x & x & x & * \\ x & x & x & 0 \\ x & x & x & 0 \\ x & x & x & 0 \\ x & x & * & 0 \\ x & x & 0 & 0 \\ x & x & 0 & 0 \\ x & * & 0 & 0 \\ * & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \lambda_i \\ \lambda_i^2 \\ \lambda_i^3 \end{bmatrix} \qquad \mathbf{d}_{i2} = \begin{bmatrix} x & x \\ x & * \\ x & 0 \\ x & 0 \\ x & 0 \\ * & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \lambda_i \end{bmatrix}$$

$$\triangleq U_1 \mathbf{v}_{i1} \qquad\qquad \triangleq U_2 \mathbf{v}_{i2}$$

$$\mathbf{d}_{i3} = \begin{bmatrix} x & x \\ x & x \\ x & * \\ x & 0 \\ \\ x & 0 \\ x & 0 \\ * & 0 \\ \\ 0 & 0 \\ \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \lambda_i \end{bmatrix} \qquad \mathbf{d}_{i4} = \begin{bmatrix} x \\ x \\ x \\ * \\ \\ 0 \\ 0 \\ 0 \\ \\ 0 \\ \\ 0 \end{bmatrix}$$

$$\underset{=}{\triangle} U_3 \mathbf{v}_{i3} \qquad \qquad \underset{=}{\triangle} U_4 \mathbf{v}_{i4}$$

Theorem 8.1

(A): For a fixed value of i (or λ_i), its corresponding p basis vectors $\mathbf{d}_{ij}$ are linearly independent.

(B): For a fixed value of j (or U_j), any set of μ_j basis vectors $\mathbf{d}_{ij}$ (μ_j different values of i) are also linearly independent. Theorem 8.1 is dual to Theorem 5.2.

Proof

Part A can be proved by the form of matrix U_j ($j = 1, \ldots, p$) in (8.32).

Part B can be proved by the fact that any μ_j of $\mathbf{d}_{ij}$ vetors (say, $i = 1, \ldots, \mu_j$) can be written as [from (8.32)]

$$[\mathbf{d}_{1j} | \ldots | \mathbf{d}_{\mu j, j}] = U_j [\mathbf{v}_{1j} | \ldots | \mathbf{v}_{\mu j, j}] \underset{=}{\triangle} U_j V_j \tag{8.33}$$

Here matrix V_j according to (8.32) is a μ_j dimensional Vandermonde matrix, which is nonsingular for different λ_i. Now Part B follows from the form of U_j of (8.32).

Part B can be extended to general eigenvalue cases. This is because in (8.33) the eigenvalues are associated only with matrix V_j which is the right eigenvector matrix of a companion form matrix [the transpose of (1.14) Brand, 1968]. Therefore, in general eigenvalue cases only the matrix V_j varies from the Vandermonde form, but remains a nonsingular right eigenvector matrix.

For example (Example 8.6), if we assign the first $4\ (= u_1)$ $\mathbf{c}_i$'s $(i = 1 - 4) = \mathbf{e}_1$, the next $2\ (= u_2)$ $\mathbf{c}_i$'s $(i = 5, 6) = \mathbf{e}_2$, the next $2\ (= u_3)$ $\mathbf{c}_i$'s $(i = 7, 8) = \mathbf{e}_3$, and the last u_4 $\mathbf{c}_i$ $(i = 9) = \mathbf{e}_4$, $(\mathbf{e}_i, i = 1, \dots, 4$ is the i-th column of a p-dimensional identity matrix), then (8.33) implies that $V = [U_1 : U_2 : U_3 : U_4]\mathrm{diag}\{V_1, V_2, V_3, V_4\}$, where V_j is a Vandermonde matrix of dimension u_j and which is formed by the vectors $\mathbf{v}_{ij}$ $(j = 1 - 4$, values of i are corresponding to the above assignment of $\mathbf{c}_i$).

Theorem 8.2

Let $U \triangleq [U_1|\dots|U_p]$ of (8.32). Let the eigenvalues λ_i $(i = 1, \dots, n)$ of the block-controllable Hessenberg form matrix $A - BK$ be divided into p groups Λ_j $(j = 1, \dots, p)$, and let each Λ_j be a Jordan form matrix and corresponds to the same ordered λ_i's of matrix V_j of (8.33).
Then (A):

$$V = U\ \mathrm{diag}\{V_1, \dots, V_p\} = [U_1 V_1 | \dots | U_p V_p] \tag{8.34}$$

is a right eigenvector matrix of $A - BK$ such that

$$V^{-1}(A - BK)V = \mathrm{diag}\{\Lambda_1, \dots, \Lambda_p\} \tag{8.35}$$

$$\text{and (B)} : U^{-1}(A - BK)U = A_c \triangleq \mathrm{diag}\{A_{c1}, \dots, A_{cp}\} \tag{8.36}$$

where A_{cj} $(j = 1, \dots, p)$ are μ_j dimensional companion form matrices.

Proof

(A): The nonsingularity of matrix V of (8.34) is proved in Theorem 8.1. Because $U_j V_j$ of (8.32–8.34) satisfies (8.1b) to (8.5) for all values of j, (8.35) is proved.

(B): Because V_j is the right eigenvector matrix of A_{cj} $(j = 1, \dots, p)$ such that

$$V_j^{-1} A_{cj} V_j = \Lambda_j, \qquad j = 1, \dots, p \tag{8.37a}$$

or

$$(\mathrm{diag}\{V_1, \dots, V_p\})^{-1} A_c (\mathrm{diag}\{V_1, \dots, V_p\}) = \mathrm{diag}\{\Lambda_1, \dots, \Lambda_p\} \triangleq \Lambda \tag{8.37b}$$

Then equality between the left-hand side of (8.35) and (8.37b) together with the definition of (8.34) proves (8.36).

It should be emphasized that the block-controllable Hessenberg form system matrices (A, B) can be computed from the original system matrices by orthonormal similarity transformation (see Algorithms 5.1 and 5.2). Hence its feedback system dynamic matrix $A - BK$'s corresponding eigenvector matrix V of (8.34) has the same condition number as that of the original system matrix.

On the contrary, the matrix U of (8.34) and (8.36) is not unitary and is often ill conditioned. Hence the eigenvector matrix $\mathrm{diag}\{V_1, \ldots, V_p\}$ of (8.37), which corresponds to the feedback system dynamic matrix A_c of (8.36), does not have the same condition number as that of matrix V of (8.34). Therefore we will work on the assignment of matrix V instead of the matrix $\mathrm{diag}\{V_1, \ldots, V_p\}$, even though the latter matrix is simpler.

Having analyzed the properties of general eigenvector assignment formulation (8.5) and (8.32), we now present the eigenvector assignment rule for decoupling. From (8.5) and (8.32), the eigenvector matrix can always be generally expressed as

$$V = U[\mathrm{diag}\{\mathbf{v}_{11}, \ldots, \mathbf{v}_{1p}\}\mathbf{c}_1 | \ldots | \mathrm{diag}\{\mathbf{v}_{n1}, \ldots, \mathbf{v}_{np}\}\mathbf{c}_n] \tag{8.38}$$

where $\mathbf{c}_i$ $(i = 1, \ldots, n)$ are p-dimensional free column vectors.

General Rule of Eigenvector Assignment for Decoupling

It is clear that the eigenvector matrix V of (8.34) is a special case of that of (8.38) in the sense that

$$\mu_j \text{ of } \mathbf{c}_i\text{'s} = \mathbf{e}_j, j = 1, \ldots, p \text{ (each value of } i \text{ corresponds to a different eigenvalue, and } \mathbf{e}_j \text{ is the } j\text{-th column of a } p\text{-dimensional identity matrix)} \tag{8.39}$$

It is also clear that while (8.38) and (8.34) have the same first (or the left) component matrix U which is fixed by the open-loop system parameters only [the lower $n - p$ rows of (8.6)], the second (or the right) component matrix of (8.38) and (8.34) is different. Specifically, the second component matrix of (8.34) is a special case of the general second component matrix of (8.38), and is decoupled into p diagonal blocks $V_j, j = 1, \ldots, p$. Because decoupling is very effective in system robustness, our analytical eigenvector assignment will be based on the form (8.34) or (8.39).

Under this general eigenvector assignment formulation, the only freedom left is on the distribution of the n eigenvalues into the p diagonal blocks of (8.34), while each block has dimension μ_j and corresponds to one of the inputs, $j = 1, \ldots, p$. The following are three analytical rules which can guide this distribution, and which are closely related to Subsection 8.1.1. Thus this analytical eigenvector assignment not only achieves decoupling, but also fully considers the analytical system parameters such as the controllability indices μ_j and the eigenvalues.

Rule 1

Distribute multiple eigenvalues (say, $\lambda_i = \lambda_{i+1}$) into different input blocks by letting $\mathbf{c}_i \neq \mathbf{c}_{i+1}$ in (8.34) or (8.39). This is because only at the same block or only for a same value of j, can the corresponding eigenvectors of λ_i and λ_{i+1} be generalized or be defective [see the paragraphs before Conclusion 5.2 and (5.15d)].

A basic result in numerical linear algebra is that the defective eigenvectors cause high sensitivity of the corresponding eigenvalues [Golub and Wilkinson, 1976b]. For example, singular value decomposition of any matrix A is always well conditioned because the eigenvectors of matrix A^*A are never defective, even though the singular values or the square roots of eigenvalues of matrix A^*A can be multiple.

Rule 2

Distribute relatively more important eigenvalues (such as the one's closer to imaginary axis, see Chap. 2) into blocks with relatively smaller size μ_j.

This is because the smaller the dimension of a matrix block, usually the smaller the condition number of that matrix block. For example, a matrix with size equal to one (a scalar) always has the smallest possible condition number $(= 1)$.

Rule 3

Distribute the n eigenvalues so that all eigenvalues within each block have as similar magnitude as possible, and have as evenly distributed phase angles (between 90° and $-90°$) as possible.

This is because such eigenvalue pattern is derived from some optimal single-input systems (see Rule (f) of Subsection 8.1.1). From a mathematical point of view, because the eigenvalue magnitudes are related to singular values in equation $\sigma_1 \geqslant |\lambda_1| \geqslant \cdots \geqslant |\lambda_n| \geqslant \sigma_n$, a large difference in eigenvalue magnitude implies a large condition number σ_1/σ_n. From a geometrical point view, less evenly distributed phase angles imply near clustered (or multiple) eigenvalue pattern.

The above three rules may not be exhaustive and cannot claim any numerical optimality. However, the analytical rules are simple, general, and do not require iterative numerical computation. Hence these rules can be applied repeatedly in a trial-and-error and adaptive fashion (adapted using the final results). The analytical results can also provide a more reasonable initial value for the numerical methods.

It should be mentioned again that once the above distribution is made, the eigenvectors can be computed directly from the open-loop system matrices (8.6) and the $\lambda_i{}'s$ without the component matrices U and $\operatorname{diag}\{V_i, i = 1, \ldots, p\}$ (see Step 1 of Algorithm 5.3 for the dual case). Nonetheless, these component matrices are computed in the following example, to demonstrate and analyze more completely our analytical eigenvector assignment rules.

Example 8.7

Let the system matrix be

$$[A : B] = \left[\begin{array}{ccccccccc} -20 & 0 & & 0 & 0 & 0 & : & 20 & 0 \\ 0 & -20 & & 0 & 0 & 0 & : & 0 & 20 \\ \cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots & \cdots\cdots & \cdots & : & \cdots\cdots & \\ -0.08 & -0.59 & : & -0.174 & 1 & 0 & : & 0 & 0 \\ -18.95 & -3.6 & : & -13.41 & -1.99 & 0 & : & 0 & 0 \\ 2.07 & 15.3 & : & 44.79 & 0 & 0 & : & 0 & 0 \end{array}\right]$$

This is the model of a fighter plane at flight condition of 3048 m and Mach 0.77 [Sobel et al., 1984; Spurgeon, 1988]. The five system states are elevator angle, flap angle, incidence angle, pitch rate, and normal acceleration integrated, respectively, while the two inputs are elevator angle demand and flap angle demand, respectively.

The problem is to design a state feedback gain K to assign the eigenstructure of matrix $A - BK$, with eigenvalues $\lambda_1 = -20, \lambda_{2,3} = -5.6 \pm j4.2$, and $\lambda_{4,5} = -10 \pm j10\sqrt{3}$.

We first compute the block-controllable Hessenberg form using the dual of Algorithm 5.2. Because B is already in the desired form, only one triangularization (for $j = 2$) of the lower left corner of matrix A is needed.

Hence the operator matrix H is

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & & & \\ 0 & 0 & & H_2 & \\ 0 & 0 & & & \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -0.0042 & -0.9941 & 0.1086 \\ 0 & 0 & 0.0379 & -0.1086 & 0.9931 \\ 0 & 0 & 0.9991 & -0 & 0.0386 \end{bmatrix}$$

The resulting block-controllable Hessenberg form is

$$[H'AH : H'B]$$
$$= \left[\begin{array}{cccccccccc} -20 & 0 & & 0 & 0 & & 0 & : & 20 & 0 \\ 0 & -20 & & 0 & 0 & & 0 & : & 0 & 20 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 19.0628 & 5.2425 & : & -2.0387 & 0.4751 & & 18.1843 & : & 0 & 0 \\ 0 & -14.8258 & : & -0.0718 & -1.6601 & & -43.0498 & : & 0 & 0 \\ \cdots & \cdots & : & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0.0002 & 0.0005 & : & -0.9931 & -0.109 & : & -0.0114 & : & 0 & 0 \end{array}\right]$$

which is still denoted as $[A{:}B]$ in the rest of this example in order to be compatible with the notation of Sec. 8.2. In this result, the elements [0.0002 0.0005] are computational error, and the controllability indexes are shown as $\mu_1 = 3, \mu_2 = 2$.

We take back substitution operation (A.20) on the last $n - p \; (= 3)$ rows of $A - \lambda_i I$ to derive the analytical expression of basis vectors $\mathbf{d}_{ij}$ of $\mathbf{v}_i$ of (8.4) such that $[0 : I_3][A - \lambda_i I]\mathbf{d}_{ij} = 0$. During each back substitution operation, we first set 1 at the fifth or fourth element of $\mathbf{d}_{ij}$, for $j = 1, 2$, respectively. As a result, the coefficient matrix of (8.32) can be derived as

$$U = [U_1 : U_2]$$
$$= \left[\begin{array}{cccccc} 0.67923 & -0.114 & -0.05395 & : & 0.9648 & 0.012415 \\ -2.889 & 0.015167 & 0 & : & -0.108166 & -0.06743 \\ -0.0043141 & -1.02945 & 0 & : & -0.11683 & 0 \\ 0 & 0 & 0 & : & 1 & 0 \\ 1 & 0 & 0 & : & 0 & 0 \end{array}\right]$$

Because two pairs of the assigned eigenvalues are complex conjugate and there are only two inputs, we have only two possible different distributions:

$$\operatorname{diag}\{V_1, V_2\} = \operatorname{diag}\left\{ \begin{bmatrix} 1 & 1 & 1 \\ \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ \lambda_4 & \lambda_5 \end{bmatrix} \right\}$$

$$\Lambda_1 = \operatorname{diag}\{\operatorname{diag}\{\lambda_1, \lambda_2, \lambda_3\}, \operatorname{diag}\{\lambda_4, \lambda_5\}\} \tag{8.40a}$$

and

$$\operatorname{diag}\{V_1, V_2\} = \operatorname{diag}\left\{ \begin{bmatrix} 1 & 1 & 1 \\ \lambda_1 & \lambda_4 & \lambda_5 \\ \lambda_1^2 & \lambda_4^2 & \lambda_5^2 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ \lambda_2 & \lambda_3 \end{bmatrix} \right\}$$

$$\Lambda_2 = \operatorname{diag}\{\operatorname{diag}\{\lambda_1, \lambda_4, \lambda_5\}, \operatorname{diag}\{\lambda_2, \lambda_3\}\} \tag{8.40b}$$

The eigenvector matrix according to (8.34) is $V = [U_1 V_1 : U_2 V_2]$ and is named V^1 and V^2 for the two assignments of (8.40), respectively. During actual computation, once the distribution of (8.40) is made, the matrix V can be computed *directly* according to the dual of (5.10b) and (5.13c) without explicit U and without complex numbers of (8.40).

To broaden the comparison, we let the third eigenvector matrix be

$$V^3 = QV_1$$
$$= \begin{bmatrix} -0.0528 & 0.0128 & -0.1096 & -0.0060 & -0.1555 \\ 0 & -0.06745 & 0.0049 & -0.1114 & -2.904 \\ 0 & 0 & -1.007 & -0.1098 & -0.01148 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1^4 & \lambda_2^4 & \lambda_3^4 & \lambda_4^4 & \lambda_5^4 \\ \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 \\ \lambda_1^3 & \lambda_2^3 & \lambda_3^3 & \lambda_4^3 & \lambda_5^3 \\ 1 & 1 & 1 & 1 & 1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & \lambda_4^2 & \lambda_5^2 \end{bmatrix}$$

where Q is a nonsingular matrix such that $(Q^{-1}AQ,\ Q^{-1}B)$ is in block-controllable canonical form [the transpose of (1.16)] [Wang and Chen, 1982]. Obviously there is no decoupling in this assignment (only one block V_1) and the Jordan form matrix of $A - BK$ corresponding V^3 is

$$\Lambda_3 = \operatorname{diag}\{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\} \tag{8.41}$$

Matrices V^i $(i = 1, 2, 3)$ are the right eigenvector matrix of $A - BK$ corresponding to eigenvalues in Λ_i $(i = 1, 2, 3)$, where the system matrix $(A,\ B)$ is in the form (8.6). Because this pair of $(A,\ B)$ is computed from the original (A, B) by orthonormal similarity transformation (with unitary matrix H), the condition number of V^i equals that of the right eigenvector matrix of the original $A - BK$.

The state feedback gain can now be determined after the eigenvectors are determined. From (8.19) [or the dual of (5.16)], we first compute the solution $K\hat{}$of (8.1b)

$$K_{\hat{i}} = B_1^{-1}[I_p : 0](AV^i - V^i\Lambda_i), \qquad i = 1, 2, 3 \tag{8.42a}$$

From (8.1b), $K_{\hat{i}}\,(V^i)^{-1} = K_i H'$ is the state feedback gain for the system matrix $H(A - BK_i)H'$ in the form (8.6). The state feedback gains K_i for the original system matrix are (see the end of Subsection 8.1.2)

$$K_i = K_{\hat{i}}(V^i)^{-1}H, \qquad i = 1, 2, 3 \tag{8.42b}$$

and whose numerical values are

$$K_1 = \begin{bmatrix} 0.4511 & 0.7991 & -1.3619 & -0.4877 & 1.0057 \\ 0.0140 & -0.0776 & 2.6043 & 0.2662 & 1.357 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 0.8944 & 0.9611 & -20.1466 & -1.7643 & 0.8923 \\ 0.0140 & -0.5176 & 1.3773 & 0.1494 & 0.1800 \end{bmatrix}$$

and

$$K_3 = \begin{bmatrix} 1 & 5044 & 14{,}565 & 23 & 1033 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

It can be verified that the eigenvalues of matrices $A - BK_i$ $(i = 1, 2)$ are correctly placed, while the eigenvalues of matrix $A - BK_3$ differ a little from the desired λ_i $(i = 1, \ldots, 5)$. This difference is caused by the computational error rather than the method. This difference also shows that having a good eigenvector assignment is important to the numerical accuracy aspect of pole placement.

Table 8.1 provides a comparison of these three results.

Table 8.1 Two Numerical Measures of State Feedback Design for Eigenstructure Assignment

	K_1	K_2	K_3
$\|K_i\|_F$	3.556	20.34	15,448
$\kappa(V^i)$	71.446	344.86	385,320

The zero-input response of state $\mathbf{x}(t)$ of the three feedback systems, corresponding to the initial state $\mathbf{x}(0) = [1 \quad 1 \quad 1 \quad 1 \quad 1]'$, is shown in Fig. 8.1.

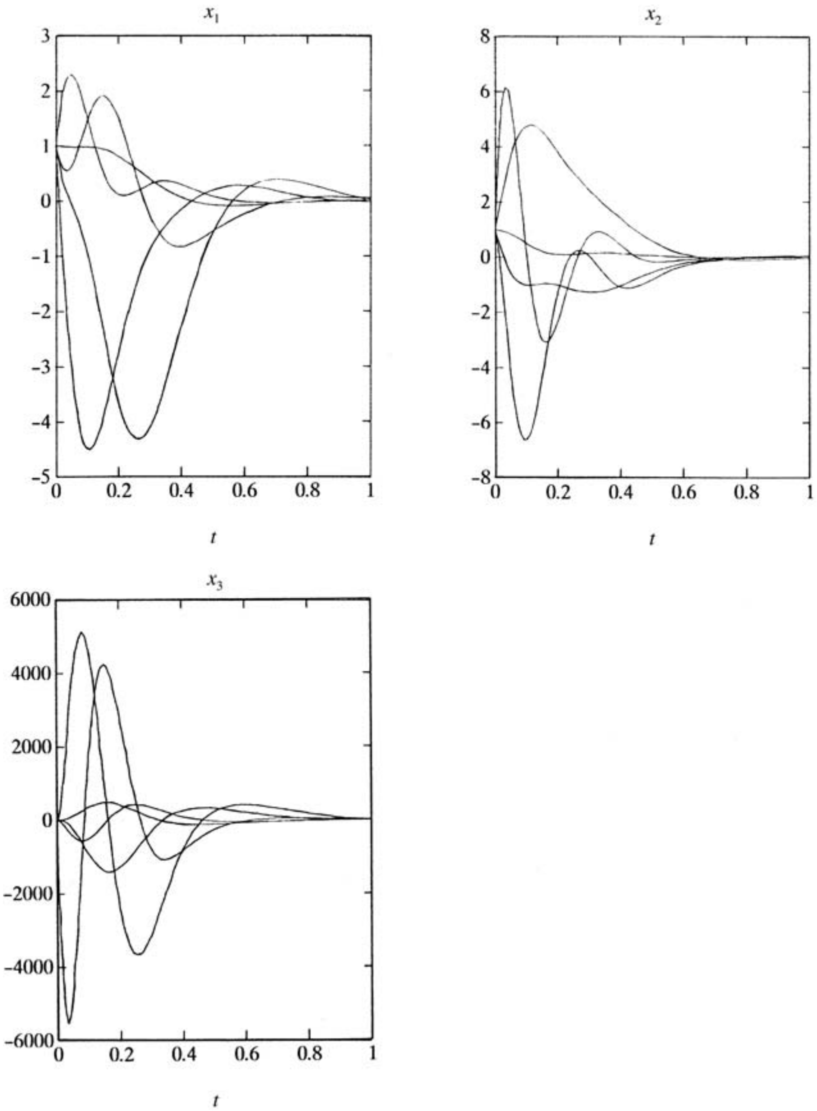


Figure 8.1 Zero-input responses of three systems with same eigenvalues but different eigenvectors.

It is clear that the lower the numerical measures in Table 8.1, the smoother the zero-input response in Fig. 8.1. From Sec. 2.2, lower values of $\kappa(V)$ in Table 8.1 also imply lower eigenvalue sensitivity (or better robust performance) and better robust stability. In addition, lower gain $\|K_i\|_F$ in Table 8.1 also implies lower control energy consumption and lower possibility of disturbance and failure. Hence the numerical measures of Table 8.1 and the response simulation of Fig. 8.1 can both be used to guide design.

The comparison of these three examples also shows that eigenvector assignment makes a dramatic difference in the aspect of technical quality of the feedback system.

Unlike the numerical methods of Subsection 8.2.1 as well as the optimal design methods of Chap. 9, there is a very explicit and analytical understanding of the relation between the above properties of final results and the design process of analytical eigenstructure assignment. Only this understanding can guide the reversed adjustment of design formulation and design procedure, based on the final results.

For example, the final result indicates that decoupling is extremely effective because the third result, which does not have decoupling, is much worse than the first two results, which have decoupling. This understanding supports the basic decoupling formulation of the analytical eigenstructure design.

For another example, a comparison between V^1 and V^2 of (8.40) indicates that the larger block is dominant among the two blocks of V^i. For the larger block V_1 (with dimension $\mu_1 = 3$), $\kappa(V_1)$ equals 653.7 and 896.5 for the two V^i's, respectively, while for the smaller block V_2 (with dimension $\mu_2 = 2$), $\kappa(V_2)$ equals 23.6 and 11.8 for the two V^i's, respectively. Yet the overall $\kappa(V^i = U \text{ diag}\{V_1, V_2\})$ $(i = 1, 2)$ is 71.44 and 344.86, respectively. Thus $\kappa(V_1)$ is dominant over $\kappa(V_2)$ in deciding the overall $\kappa(V^i)$. This understanding reinforces the second rule (somehow over the third rule) of the analytical eigenvector design.

SUMMARY

To summarize, Sec. 8.2 has introduced three relatively simple, systematic, and general eigenvector assignment methods, and showed the dramatic difference in feedback system quality caused by different eigenvector assignment. Eigenvectors determine the robustness properties of their corresponding eigenvalues, and their assignment exists only in multi-input and multi-output (MIMO) system design problems, while eigenvalues can most directly determine system performance. Now the robustness properties of our control are also guaranteed of full realization for most system conditions and for the first time. Thus if the distinct advantage of modern control theory was reflected mainly in the *description* of MIMO systems before and since the 1960s, then this advantage can now be reflected in the *design* of such control systems.

EXERCISES

8.1 Suppose $n = 10$ and all assigned eigenvalues must be complex conjugate.

(a) If $q = 7$ and $p = 4$, can you use Algorithm 8.1 directly? Why? Can you use the dual version of Algorithm 8.1 directly? Why?
(b) If $q = 8$ and $p = 3$, can you use Algorithm 8.1 directly? Why? Can you use the dual version of Algorithm 8.1 directly? Why?
(c) If $q = 7$ and $p = 5$, how can you use Algorithm 8.1 directly?
(d) If $q = 7$ and $p = 5$, how can you use the dual version of Algorithm 8.1 directly?

8.2 Repeat Examples 8.1 and 8.2 by assigning eigenvalue -2 at Step 1 and $\{-1$ and $-3\}$ at Step 2.

8.3 Use Algorithm 8.1 and its dual version to assign poles $\{-1, -2, -3, -4\}$ to the following system [Chu, 1993a]. Notice that the provided answer $\overline{K}$ is not unique:

(a)
$$(A, B, \overline{C}, \overline{K}) = \left(\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -47 & 34 & 10 \\ 49 & -35 & -11 \end{bmatrix} \right)$$

(b)
$$(A, B, \overline{C}, \overline{K}) = \left(\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} -10 & 4.32 \\ 62 & -35 \\ 52.58 & -29/84 \end{bmatrix} \right)$$

8.4 Repeat Example 8.3 to assign eigenvalues $\{-1, -2, -3, -4\}$ to the following system [Chu, 1993a]

$$(A, B, \overline{C}, \overline{K}) = \left(\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -50 & -49.47 \\ 40.49 & 40 \end{bmatrix} \right)$$

8.5 Repeat Example 8.3 and using the dual of (5.15) to assign eigenvalues $\{-1, -1, -2, -2\}$ to the following system [Kwon, 1987]

$$(A, B, \overline{C}, \overline{K}) = \left(\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 14 & 6 \\ 19 & 18 \end{bmatrix} \right)$$

8.6 Let $n = 5, p = m = 2$ and, according to Conclusion 6.3, $q = m + r$, where r is the number of stable transmission zeros of the system. According to Davison and Wang [1974], there are generically $n - m = 3$ transmission zeros of the system. Therefore we assume there are always 3 transmission zeros in this system.

Let the probability of each zero being stable as $P_1 = 1/2$ (if the system model is given arbitrarily) or as $P_2 = 3/4$ (three times better than $1/2$), respectively. Then the probability of minimal-phase (all three zeros are stable) is $(P_1)^3 = 0.125$ or $(P_2)^3 = 0.422$, respectively (see Exercises of Chap. 4).

Answer the following questions based on P_1 and P_2. The probability of r stable zeros is $[r:3](P_i)^r(1 - P_i)^{3-r}$ ($i = 1, 2, [r:3]$ is the combination of r out of 3, also see Exercise 4.2).

(a) The probability of full (arbitrary) state feedback.

Answer: $q = n, r = 3, P_1(r = 3) = 0.125, P_2(r = 3) = 0.422.$

(b) The probability of arbitrary pole placement and partial eigenvector assignment or better.

Answer: $q \geqslant 4$ so that $q + p > n, r \geqslant 2, P_1(r \geqslant 2) = 0.5, P_2(r \geqslant 2) = 0.844.$

(c) The probability of arbitrary pole placement with no eigenvector assignment or better.

Answer: $q \geqslant 3$ so that $q \times p > n, r \geqslant 1, P_1(r \geqslant 1) = 0.875, P_2(r \geqslant 1)$
$= 0.9844$.

(d) The probability of no arbitrary pole or eigenvector assignment (ordinary static output feedback).

Answer: $q = m = 2$ so that $q \times p \not> n, r = 0, P_1(r = 0)$
$= 0.125, P_2(r = 0) = 0.0156$.

This example shows quite convincingly the following three decisive advantages of the new design approach of this book. (1) It is very general even it is required to be good enough [see answer of Part (c)]. (2) It achieves exact LTR far more generally than the existing state feedback control as shown by comparing the probability of Part (c) with the probability of one of the conditions of existing exact LTR—minimum-phase. (3) Its control can achieve arbitrary pole assignment far more generally than the existing static output feedback as shown by comparing the probability of Part (c) with 0%.

8.7 Repeat Problem 8.6 by changing only the parameter n from 5 to 4.

8.8 Assign poles $\{\lambda_1 = -1, \lambda_{2,3} = -2 \pm j, \lambda_{4,5} = -1 \pm j2\}$ by state feedback:

$$A = \begin{bmatrix} 0 & 1 & 0 & : & 0 & 0 \\ 0 & 0 & 1 & : & 0 & 0 \\ 2 & 0 & 0 & : & 1 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & : & 0 & 1 \\ 0 & 0 & 0 & : & -1 & -2 \end{bmatrix} \quad \text{and}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ \cdots & \cdots \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Verify and compare eigenvectors, condition of eigenvector matrix

$\kappa(V)$, norm of feedback gain $\|K\|_F$, robust stability (as in Example 2.5) of the feedback system matrix $A - BK$, and zero-input response for $\mathbf{x}(0) = [2 \quad 1 \quad 0 \quad -1 \quad -2]'$, for the following three eigenstructure assignments (partial answers are provided):

(a) $(\Lambda, K) = \left(\text{diag}\{\lambda_{1,2,3}, \lambda_{4,5}\}, \begin{bmatrix} 7 & 9 & 5 & 1 & 1 \\ 0 & 0 & 0 & 4 & 0 \end{bmatrix}\right)$

(b) $(\Lambda, K) = \left(\text{diag}\{\lambda_{1,4,5}, \lambda_{2,3}\}, \begin{bmatrix} 7 & 7 & 3 & 1 & 1 \\ 0 & 0 & 0 & 4 & 2 \end{bmatrix}\right)$

(c) $(\Lambda, K) = \left(\text{diag}\{\lambda_{1,2,3,4,5}\}, \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 25 & 55 & 48 & 23 & 5 \end{bmatrix}\right)$

Hint:

1. Refer to Example 8.7
2. System (A, B) is a state permutation away from the block-controllable Hessenberg form (and canonical form). To derive the latter, permute the rows of (A, B) and the columns of A for the new sequence $\{3, 5, 2, 4, 1\}$.

8.9 Repeat 8.8 for the new system

$$A = \left[\begin{array}{ccc:cc} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 1 & 2 \\ \hdashline 0 & 0 & 0 & 0 & 1 \\ 4 & 3 & 1 & -1 & -4 \end{array}\right] \quad \text{and}$$

$$B = \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ \hdashline 0 & 0 \\ 0 & 1 \end{array}\right]$$

and for the following three different eigenstructure assignments:

(a) $(\Lambda, K) = \left(\text{diag}\{\lambda_{1,2,3}, \lambda_{4,5}\}, \begin{bmatrix} 8 & 10 & 5 & 1 & 2 \\ 4 & 3 & 1 & 4 & -2 \end{bmatrix}\right)$

(b) $(\Lambda, K) = \left(\text{diag}\{\lambda_{1,4,5}, \lambda_{2,3}\}, \begin{bmatrix} 8 & 8 & 3 & 1 & 2 \\ 4 & 3 & 1 & 4 & 0 \end{bmatrix}\right)$

(c) $(\Lambda, K) = \left(\operatorname{diag}\{\lambda_{1,2,3,4,5}\}, \begin{bmatrix} 3 & 1 & 0 & 0 & 2 \\ 29 & 58 & 49 & 23 & 3 \end{bmatrix} \right)$

8.10 Repeat 8.8 for the new system

$$A = \left[\begin{array}{ccc|cc} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -10 & -16 & -7 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 & 1 \\ 4 & 3 & 1 & -2 & -2 \end{array} \right] \quad \text{and}$$

$$B = \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ \hline 0 & 0 \\ 0 & 1 \end{array} \right]$$

and for the following two different eigenstructure assignments:

(a) $(\Lambda, K) = \left(\operatorname{diag}\{\lambda_{1,2,3}, \lambda_{4,5}\}, \begin{bmatrix} -5 & -7 & -2 & 1 & 2 \\ 4 & 3 & 1 & 3 & 0 \end{bmatrix} \right)$

(b) $(\Lambda, K) = \left(\operatorname{diag}\{\lambda_{1,4,5}, \lambda_{2,3}\}, \begin{bmatrix} -5 & -9 & -4 & 1 & 2 \\ 4 & 3 & 1 & 3 & 2 \end{bmatrix} \right)$

Also see the last three design projects of Appendix B for the exercises of numerical eigenvector assignment algorithms.

9

Design of Feedback Control—Quadratic Optimal Control

In the modern control design literature, besides eigenstructure assignment, there is a main result called "linear quadratic optimal control" (LQ). The two designs are quite opposite in direction. The eigenstructure assignment, especially the analytical eigenvector assignment, is designed mainly from the bottom up, based on the given plant system's structure. On the contrary, the LQ control is designed from top down, based on a given and abstract optimal criterion as

$$J = (1/2)\int_0^\infty [\mathbf{x}(t)'Q\mathbf{x}(t) + \mathbf{u}(t)'R\mathbf{u}(t)]dt \tag{9.1}$$

where Q and R are symmetrical, positive semi-definite and symmetrical, positive definite matrices, respectively. The LQ design is aimed at minimizing J of (9.1) under the constraint (1.1a)

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \qquad \text{and} \qquad \mathbf{x}(0) = \mathbf{x}_0$$

Inspection of (9.1) shows that to minimize or to have a finite value of J, $\mathbf{x}(t \to \infty)$ must be 0. Hence the control system must be stable (see Definition 2.1). In addition, among the two terms of J, the first term reflects the smoothness and quickness of $\mathbf{x}(t)$ before it converges to 0, while the second term reflects the control energy, which is closely related to control gain and system robustness (see Example 8.7). Hence the LQ design can consider both performance and robustness.

Performance and robustness properties in general are contradictory to each other. For example, the faster the $\mathbf{x}(t)$ converges to 0, the higher the control power needed to steer $\mathbf{x}(t)$.

The two weighting matrices Q and R can reflect the relative importance of these two properties. A relatively large Q (compared to R) indicates higher priority for performance over control energy cost. When $R = 0$, the corresponding LQ problem is called "minimal (response) time problem" [Friedland, 1962]. Anti-air missile control problems are such problems. On the other hand, a relatively small Q (compared to R) indicates higher priority on saving control energy over performance. When $Q = 0$, the corresponding LQ problem is called the "minimal fuel problem" [Athanassiades, 1963]. Remote-orbit space craft control can be considered such a problem.

However, the above design consideration is made in terms of *only* the magnitude of matrices Q and R. There are no other general, analytical, and explicit considerations of system performance and robustness made on the n^2 parameters of Q and the p^2 parameters of R (or criterion J). Hence the LQ problem itself is still very abstract and reflects still vaguely the actual system performance and robustness. For example, the problem (J) is set without considering the information of the plant system parameters (A, B). To summarize, matrices Q and R (or J) are not really the direct and accurate reflection of actual system performance and robustness.

This critical problem is further compounded by the fact that unlike the eigenstructure assignment, once the criterion J is set, it is very hard to systematically, automatically, and intelligently adjust it based on the finally computed design solution and its simulation. This is due to the fact that complicated and iterative numerical computation is needed to compute the solution that minimizes J. The comparison of computational difficulty of all design algorithms of Chaps 8 and 9 is made in Sec. 9.3.

It should be noticed that the above two critical drawbacks of LQ design is at least shared by all other optimal design results, if not more severe. For example, the optimal design problems based on the system frequency response are even less direct and less generally accurate in reflecting system performance and robustness (see Chap. 2), and many optimal control problems other than the LQ problem require even much more computation than that of the LQ problem.

Regarding the LQ control problem, it has been extensively studied and covered in the literature. This book intends to introduce only the basic design algorithm and basic physical meanings of this problem. Readers can refer to the ample existing literature for the corresponding theoretical analysis, proofs, and generalizations.

As with the presentation of eigenstructure assignment, the LQ design of this chapter is divided into state feedback control $K\mathbf{x}(t)$ and generalized state feedback control $\overline{K}\overline{C}\mathbf{x}(t)$ [rank$(\overline{C})\leqslant n$] cases, which are treated in Secs 9.1 and 9.2, respectively.

9.1 DESIGN OF DIRECT STATE FEEDBACK CONTROL

The direct state feedback design for LQ optimal control has been extensively covered in literature [Kalman, 1960; Chang, 1961; Pontryagin, 1962; Athans, 1966; Bryson and Ho, 1969; Anderson, 1971; Kwakernaak and Sivan, 1972; Sage and White, 1977]. The following solution can be formulated using calculus of variation with Lagrange multipliers.

Theorem 9.1

The unique solution that minimizes J of (9.1) and that is subject to (1.1a) is

$$\mathbf{u}^*(t) = -K^*\mathbf{x}(t), \qquad \text{where} \qquad K^* = R^{-1}B'P \tag{9.2}$$

and P is the symmetric and positive definite solution matrix of the following algebraic Riccati equation (ARE)

$$PA + A'P + Q - PBR^{-1}B'P = 0 \tag{9.3}$$

Based on the LQ optimal control $\mathbf{u}^*(t)$, there is an optimal state trajectory $\mathbf{x}^*(t)$ which is the solution of

$$\dot{\mathbf{x}}^*(t) = A\mathbf{x}^*(t) + B\mathbf{u}^*(t), \mathbf{x}^*(0) = \mathbf{x}_0 \tag{9.4}$$

and the minimized LQ criterion is

$$J^* = \left(\frac{1}{2}\right)\mathbf{x}_0' P \mathbf{x}_0 \tag{9.5}$$

Theorem 9.1 indicates that the main difficulty of LQ optimal design concerns the solving of ARE (9.3). There are a number of numerical methods available for solving (9.3) such as the method of eigenstructure decomposition of Hamiltonian matrix [Van Loan, 1984; Byers, 1983, 1990; Xu, 1991]

$$H = \begin{bmatrix} A & -BR^{-1}B' \\ -Q & -A' \end{bmatrix} \tag{9.6}$$

and the method of matrix sign function [Byers, 1987], etc. The basic version of the first method with Schur triangularization [Laub, 1979] is described in the following.

Algorithm 9.1. Solving Algebraic Riccati Equation (ARE)

Step 1: Compute the Hamiltonian matrix H of (9.6).

Step 2: Make Schur triangularization [Francis, 1961, 1962] of matrix H such that

$$U'HU = S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}, \qquad U'U = I \tag{9.7}$$

where matrix S is an upper triangular (called Schur triangular form) matrix whose diagonal elements equal the eigenvalues of H (except a 2×2 diagonal block for complex conjugate eigenvalues), and the eigenvalues in matrix S_{11} are stable.

Step 2a: Let $k = 1, H_1 = H$.

Step 2b: Compute the unitary matrix Q_k such that

$$Q'_k H_k = R_k \tag{9.8}$$

where R_k is upper triangular (see Appendix A, Sec. 2).

Step 2c: Compute $H_{k+1} = R_k Q_k$. (9.9)

Step 2d: If H_{k+1} is already in Schur triangular form (9.7), then go to Step 3; otherwise let $k = k + 1$ and go back to Step 2b.

Step 3: Based on (9.8) and (9.9),

$$H_{k+1} = Q'_k H_k Q_k = Q'_k \cdots Q'_1 H_1 Q_1 \cdots Q_k$$

Therefore the solution matrix U of (9.7) is

$$U = Q_1 \cdots Q_k \underset{=}{\triangle} \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \}n \tag{9.10}$$

A comparison of (9.7) and (9.3) shows that

$$P = U_{21} U_{11}^{-1} \tag{9.11}$$

To accelerate the convergence of Step 2 in the actual computation, the Step 2b [or (9.8)] can be adjusted such that it becomes

$$Q'_k (H_k - s_k I) = R_k$$

and Step 2c [or (9.9)] can be adjusted correspondingly such that

$$H_{k+1} = R_k Q_k + s_k I$$

This adjusted version of Step 2 is called the "shifted version," where s_k is determined by the eigenvalues ($a_k \pm jb_k$, or a_k and b_k) of the bottom right 2×2 corner block of H_k:

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

The actual value of s_k is recommended as [Wilkinson, 1965]

$$s_k = \begin{cases} a_k, & \text{if } a_k \pm jb_k \text{ or if } |a_k - h_{2n,2n}| \leqslant |b_k - h_{2n,2n}| \\ b_k, & \text{if } |a_k - h_{2n,2n}| > |b_k - h_{2n,2n}| \end{cases}$$

It is clear that the main computation of Algorithm 9.1 is at Step 2b, which is repeated within Step 2 until matrix H_k converges to the Schur triangular form. From Sec. A.2, the order of computation of Step 2b using the Householder method is about $2(2n)^3/3$ (the dimension of matrix H is $2n$). Hence the computation required by Step 2 can be very complex.

Because of some special properties of Hamiltonian matrix H, it is possible to half the dimension of H during the computation of Step 2. One such algorithm [Xu, 1991] is described briefly in the following.

First compute H^2, which is skew symmetrical ($H^2 = -(H^2)'$). The Schur triangularization will be made on H^2.

Make elementary symplectic transformation on H^2 [Paige and Van Loan, 1981] such that

$$V'H^2V = \begin{bmatrix} H_1 & X \\ 0 & H_1' \end{bmatrix}, (V'V = I) \tag{9.12}$$

where H_1 is in upper Hessenberg form (5.1). This is the key step of the revised version of Step 2.

Make the Schur triangularization on matrix H_1, which has dimension n (instead of $2n$). This is still the main computational step, with the order of computation at each iteration equal to $2n^3/3$.

Finally, compute the square root of the result of the Schur triangularization of H_1 [Bjorck and Hammaling, 1983] in order to recover this result to that of the original Hamiltonian matrix H.

9.2 DESIGN OF GENERALIZED STATE FEEDBACK CONTROL

Generalized state feedback $\overline{KC}\mathbf{x}(t)$ is a state feedback with or without constraint [for rank$(\overline{C}) < n$ or $= n$, respectively]. Therefore its design result is weaker than that of state feedback if rank$(\overline{C}) < n$ but it can also equal that of the state feedback if rank$(\overline{C}) = n$.

Among the existing methods of this design [Levine and Athans, 1970; Cho and Sirisena, 1974; Horisberger and Belanger, 1974; Toivonen, 1985; Zheng, 1989; Yan et al., 1993], that of Yan et al. [1993] is briefly described in the following because this result satisfies the above-stated generalized properties. This method is called the gradient method.

This method is based on the partial derivative of J of (9.1) with respect to $\overline{K}$:

$$\frac{\partial J}{\partial \overline{K}} = [R\overline{KC} - B'P]\, L\overline{C}' \tag{9.13}$$

where P and L are the positive semi-definite solution matrices of the following two Lyapunov equations:

$$P(A - B\overline{KC}) + (A - B\overline{KC})'P = -\overline{C}'\overline{K}'R\overline{KC} - Q \tag{9.14}$$

and

$$L(A - B\overline{KC})' + (A - B\overline{KC})\, L = -P \tag{9.15}$$

Based on this result, the gradient flow of $\overline{K}$ with respect to J is the homogeneous differential equation

$$\dot{\overline{K}} = [B'P - R\overline{KC}]\, L\overline{C}' \tag{9.16}$$

whose solution $\overline{K}$ can be computed by a number of numerical methods. The simplest is the first-order "Euler method":

$$\overline{K}_{i+1} = \overline{K}_i + \Delta\overline{K}_i\Delta t = \overline{K}_i + ([B'P_i - R\overline{K}_i\overline{C}]L_i\overline{C}')\Delta t \tag{9.17}$$

where $\Delta\overline{K}_i$ or P_i and L_i must satisfy (9.14) and (9.15) for the current $\overline{K}_i$, and the initial constant values $\overline{K}_0$ and interval Δt should be set to guarantee the convergence and the speed of convergence of (9.17) [Helmke and Moore, 1992].

Theorem 9.2.

Define **J** to be a set of finite J of $\overline{K}$ (9.1), $(J(\overline{K}))$. In addition, the set **J** includes the global and local minima of $J(\overline{K})$. Then under the assumption that $J(\overline{K}_0)$ is finite, the gradient method (9.14)–(9.16) has the following four properties:

1. The gradient flow (9.16) has a unique solution $\overline{K}$ such that $J(\overline{K}) \in \mathbf{J}$.
2. The index $J(\overline{K}_i)$ is nonincreasing with each increment of i.
3. $\lim_{i\to\infty} \Delta\overline{K}_i = 0$ (9.18)

4. There is a convergent sequence $\overline{K}_i$ to the equilibrium of (9.16) whose corresponding $J(\overline{K}_\infty) \in \mathbf{J}$.

Proof

The proofs can be found in Yan et al. [1993].

In addition, the inspection of (9.13), (9.16), and (9.17) shows that when Rank $(\overline{C}) = n, \overline{K} = R^{-1}B'P\overline{C}^{-1}$, which equals the optimal solution of state feedback case $\overline{K} = R^{-1}B'P$ (9.2) when $\overline{C} = I$. Thus (9.16) and its solution unify the result of the state feedback case of Sec. 9.1 as its special case.

Similar to the state feedback case, the main computation of (9.16)–(9.17) is the solving of Lyapunov equations (9.14)–(9.15). There are a number of such numerical methods available [Rothschild and Jameson, 1970; Davison, 1975]. The following is a method using Schur triangularization [Golub et al., 1979].

Algorithm 9.2 Solving Lyapunov Equation $AP + PA' = -Q$

Step 1: Make a Schur triangularization of matrix A:

$$U'AU = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1r} \\ 0 & A_{22} & \dots & A_{2r} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & A_{rr} \end{bmatrix} \qquad (U'U = I) \qquad (9.19)$$

where A_{ii} $(i = 1, \dots, r)$ are 1×1 or 2×2 real matrix blocks. The actual computation of this step is discussed in Step 2 of Algorithm 9.1.

Step 2: Compute

$$U'QU = \begin{bmatrix} Q_{11} & \dots & Q_{1r} \\ \vdots & & \vdots \\ Q_{r1} & \dots & Q_{rr} \end{bmatrix}$$

where Q_{ij} has the same dimension as that of $A_{ij}, \forall i, j$.

Step 3: Replace A, P, and Q of the Lyapunov equation by $U'AU, U'PU$, and $U'QU$, respectively, to get

$$U'AUU'PU + U'PUU'A'U = -U'QU \tag{9.20}$$

or

$$\begin{bmatrix} A_{11} & \dots & A_{1r} \\ & \ddots & \vdots \\ 0 & & A_{rr} \end{bmatrix} \begin{bmatrix} P_{11} & \dots & P_{1r} \\ \vdots & & \vdots \\ P_{r1} & \dots & P_{rr} \end{bmatrix} + \begin{bmatrix} P_{11} & \dots & P_{1r} \\ \vdots & & \vdots \\ P_{r1} & \dots & P_{rr} \end{bmatrix} \begin{bmatrix} A'_{11} & & 0 \\ \vdots & \ddots & \\ A'_{1r} & \dots & A'_{rr} \end{bmatrix} = -\begin{bmatrix} Q_{11} & \dots & Q_{1r} \\ \vdots & & \vdots \\ Q_{r1} & \dots & Q_{rr} \end{bmatrix}$$

Solving (9.20), we have for $i = r, r-1, \dots, 1$ and $j = r, r-1, \dots, 1$:

$$P_{ij} = \begin{cases} P'_{ji}, & \text{if } i < j \\ \left.\begin{aligned} & -(A_{ii} + A_{jj})^{-1}\left[Q_{ij} + \sum_{k=i+1}^{r} A_{ik}P_{kj} + \sum_{k=j+1}^{r} P_{ik}A'_{jk}\right] \text{(for scalar } A_{jj}) \\ & -\left[Q_{ij} + \sum_{k=i+1}^{r} A_{ik}P_{kj} + \sum_{k=j+1}^{r} P_{ik}A'_{jk}\right](A_{ii} + A_{jj})^{-1} \text{(for scalar} A_{ii}) \end{aligned}\right\} & \text{if } i \geqslant j \end{cases} \tag{9.21}$$

There are two possible formulas for the case $i \geqslant j$ because matrix blocks A_{ii} and A_{jj} can have different dimensions. In this case the scalar block must be multiplied by I_2 before being added to the other 2×2 block. These two formulas are equivalent if both blocks are scalar. However, when both blocks are 2×2, then the corresponding solution P_{ij} will be the solution of a 2×2 Lyapunov equation

$$A_{ii}P_{ij} + P_{ij}A'_{jj} = -\overline{Q}_{ij} \tag{9.22}$$

where $\overline{Q}_{ij}$ equals the matrix inside the square brackets of (9.21). Because (9.22) is a special case of (8.1) [or the dual of (4.1)], we can use the formula (8.3) for solving (8.1) to solve (9.22). To do this, we let $P_{ij} = [\mathbf{p}_1 : \mathbf{p}_2]$ and $\overline{Q}_{ij} = [\mathbf{q}_1 : \mathbf{q}_2]$.

Then

$$[I_2 \otimes A_{ii} + A_{jj} \otimes I_2][\mathbf{p}_1' : \mathbf{p}_2']' = -[\mathbf{q}_1' : \mathbf{q}_2']'$$

can provide all parameters of P_{ij}.

Step 4: Compare (9.20) with the original equation $AP + PA' = -Q$,

$$P = U \begin{bmatrix} P_{11} & \dots & P_{1r} \\ \vdots & & \vdots \\ P_{r1} & \dots & P_{rr} \end{bmatrix} U'$$

The main computation of the above algorithm is still at Step 1 of Schur triangularization. Although the matrix $A - B\overline{KC}$ [of (9.14) and (9.15)] of this step has dimension n while the dimension of a Hamiltonian matrix of Algorithm 9.1 is $2n$, the entire Algorithm 9.2 has to be iteratively used within another iteration loop of (9.17). Hence the generalized state feedback LQ design is much harder than the state feedback LQ design.

9.3 COMPARISON AND CONCLUSION OF FEEDBACK CONTROL DESIGNS

The order of computation of the design methods of Chaps 8 and 9 is summarized in the following Table 9.1. As stated at the beginning of Chap. 8, the designs of these two chapters determine fully the feedback control and the corresponding feedback system loop transfer function. This control and its loop transfer function are guaranteed of full realization by the generalized output feedback compensator of this book.

It should be noted that orthonormal matrix operation is uniformly used in the main step of each design algorithm. Hence the order of computation of Table 9.1 is based on compatible assumptions and can therefore reveal the relative difficulty of each design algorithm.

Although the actual number of iterations needed in each loop/layer differs from problem to problem, it can be very huge (more than n^4) before convergence to a reasonably accurate value (if convergent at all). This is why applied mathematicians make great effort just to half the size of the Hamiltonian matrix before let it go through iteration for Schur triangularization (see the end of Sec. 9.1). Hence the computational difficulty is determined by the number of layers of iteration, as listed in the middle column of Table 9.1.

Table 9.1 Computational Difficulties of Feedback Control Design Methods

Design methods for $A - B\overline{KC}$, where $A \in R^{n\times n}$, $B \in R^{n\times p}$, $\overline{C} \in R^{q\times n}$ are given	Number of layers of iterations for convergence needed in design algorithm	Order of computation in each iteration
Pole assignment:		
Compute (8.6)	0 (Algorithm 5.2)	$4n^3/3$
Compute (8.4)	0 [see (A.20)]	$np(n-q)^2/2$
Compute (8.20) (for Algorithms 8.2 and 8.3)		$2n^4/3$
Eigenvector assignment:		
Analytical methods	0	0
Algorithm 8.2	1	$n^2 2$ to $2n^3/3$
Algorithm 8.3	1	$4pn$
LQ optimal control design:		
State feedback case	1	$2n^3/3$ to $2(2n)^3/3$
Generalized state feedback	2	$2^3/3$

The middle column of Table 9.1 shows clearly that eigenstructure assignment is much easier than LQ optimal design, and the state feedback design is much easier than the generalized state feedback design.

Furthermore, it seems that the addition of each constraint equation to the optimal criterion would result in one more layer of iteration for convergence. For example, because of the addition of a simple constraint equation $K = \overline{KC}$, the generalized state feedback design for LQ has one more layer of iteration than that of the state feedback design for LQ.

It should be noticed that under the new design approach of this book, the dynamic part of the feedback compensator is fully determined in Chap. 6, while Table 9.1 deals only with the design of the compensator's output part $K = \overline{KC}$. Thus the design of Chaps 8 and 9 is already much simplified and much more specific than any design of the whole dynamic feedback compensator.

This simplification should be general for the designs of control objectives other than that of Chaps 8 and 9. For example, H_∞ design is much more simplified and specific in either the state feedback case [Khargoneker, 1988] or the generalized state feedback case [Geromel et al., 1993; Stoustrup and Niemann, 1993]. Other optimal designs such as H_2 [Zhou, 1992; Yeh et al., 1992] and L_1 [Dahleh and Pearson, 1987; Dullerud and Francis, 1992] should have similar simplification and specification, when applied to the design of $\overline{KC}\mathbf{x}(t)$ only.

Out of so many design methods for $\overline{KC}\mathbf{x}(t)$, each has claimed exclusive optimality (the unique solution of one optimal design is not shared by the other optimal designs), we recommend strongly eigenstructure assignment because of the following two distinct advantages.

The first decisive advantage is at design formulation. Eigenvalues determine system performance far more directly and generally accurately (see Sec. 2.1), while the robustness properties of these eigenvalues are determined by their corresponding eigenvectors. Hence their assignment should improve feedback system performance and robustness far more directly and therefore effectively. For example, there is a whole subsection (Subsection 8.1.1) dealing with the translation of system properties to the eigenvalues.

In sharp contrast, there is virtually no general, analytical, and explicit translation from these properties to the weighting matrices (except their magnitude) of any of the optimal design formulations. There is no consideration of open-loop system parameters into the weighting matrices of any of these optimal design formulations either. The frequency response measures of system properties are even less direct and generally accurate. For example the bandwidth is far less generally accurate in reflecting system performance (see Sec. 2.1 especially Example 2.2), while the robust stability measures from the frequency response methods such as gain margins and phase margins are far less generally accurate either (see Subsection 2.2.2). In fact, the existing optimal design result is optimal only to the very abstractly and narrowly defined criterion which does not reflect generally accurately real system performance and robustness. This is further evidenced by the very existence of more than one of these optimal definitions such as H_∞, H_2, and L_1.

The second decisive advantage is at the ability to adjust the design formulation from the final design results and simulations. In practice, there can be no real good design without this feedback and adjustment. In this book, eigenvalue selection (Subsection 8.1.1) and placement (Subsection 8.1.4) are adjustable, as well as the numerical eigenvector assignment (weightings to each eigenvector) and the analytical eigenvector assignment (see Example 8.7). In addition, the feedback compensator order of our design is also fully adjustable for the tradeoff between the control strength and the degree of realization of robustness properties of this control (see Sec. 6.4).

From the middle column of Table 9.1, the computation of the solution of LQ optimal design formulations requires iteration for convergence. Notice that even more layers of iteration for convergence are required by some optimal designs other than the LQ optimal design, such as the state space version of H_∞ design where several Riccati equations are to be satisfied simultaneously [Zho et al., 1995]. This kind of design computation not only is difficult to implement in practice, but also loses the track between the design formulation and the numerical design solution. Thus in optimal designs, the automatic, general, intelligent, and analytical adjustment of the design formulation (from the design solution) is virtually impossible, even though these design formulations are too abstractly and narrowly defined to truly reflect real system performance and robustness.

We believe that these two distinct advantages are also the main reasons that made the state space theory prevalent over the previously prevalent and loop transfer function–based classical control theory in the 1960s and 1970s (eigenvectors can be assigned only based on state space models and only by state/generalized state feedback control). The problem with the state space theory is not at its form of control $\overline{KC}\mathbf{x}(t)$, but at the failure to realize this control especially its robustness properties (generating the signal $\overline{KC}\mathbf{x}(t)$ is not enough). Now this failure is claimed overcome decisively by the design of this book (Chap. 6).

EXERCISES

9.1 Let the system be

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Design state feedback K such that the quadratic criterion (9.1) with

$$Q = \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{and} \qquad R = 1 \text{ is minimized}$$

Answer: $K = [-2 \quad -2]$.

9.2 Repeat 9.1 for a different quadratic criterion (9.1):

$$J = \int_0^\infty \left[2x_1(t)^2 + 2x_1(t)x_2(t) + x_2(t)^2 + \mathbf{u}(t)^2 \right] dt$$

Hint :

$$Q = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \qquad \text{and} \qquad R = 1$$

9.3 Repeat 9.1 for a different quadratic criterion (9.1):

$$J = \int_0^\infty \left[y(t)^2 + 2\mathbf{u}(t)^2 \right] dt \ \text{(Given } \mathbf{y}(t) = C\mathbf{x}(t) = [1 \quad 2]\mathbf{x}(t))$$

$Hint: Q = C'C.$

9.4 Let the system (A, B, C) be the same as that of 9.1 and 9.3. Design a static output feedback gain K for the three criteria of 9.1 to 9.3, respectively.

9.5 (a) Randomly generate five 10×10 matrices. Calculate the Schur triagularization (Step 2 of Algorithm 9.1) and notice the average computational time.

(b) Repeat Part (a) by five 9×9 matrices. Compare the average computational time with that of Part (a) to see the effect of increasing the matrix size from 9 to 10.

(c) Repeat Part (a) by ten 5×5 matrices. Compare the average computational time with that of Part (a) to see the effect of doubling the matrix size from 5 to 10.

10

Design of Failure Detection, Isolation, and Accommodation Compensators

Failure detection, isolation, and accommodation has been an important control system design problem for some years. Whereas the control systems analyzed and designed in the previous chapters deal with minor model uncertainty and disturbance which are less serious and occur often, the control system of this chapter deals with major failure and disturbance which are severe but occur rarely.

Therefore, if the normal control system is designed to have *general* robustness properties regardless of specific model uncertainty and disturbance, then the control system of this chapter is designed to accommodate some *specific* failure situations based on their detection and diagnosis. It is ironic that the strength of failure signals makes their detection and diagnosis relatively easier.

Failure detection, isolation, and accommodation problems are specialized with regard to each specific failure situation. The results are diverse and are summarized in survey papers such as Frank [1990] and Gertler [1991].

This chapter introduces only a specific failure detection, isolation, and accommodation controller and its design algorithm, which have been published in [Tsui, 1993c, 1994b, 1997].

This controller can be designed systematically and generally, can detect and isolate failure very quickly and specifically, can accommodate failure very quickly, generally, and effectively, and can consider minor plant system model uncertainty and output measurement noise. In addition, the failure signal is generally and specifically modeled so that it corresponds to each plant system state, and the controller has very compatible structure with the normal dynamic output feedback compensator of the rest of this book. Therefore, the normal and failure controllers of this book can be designed, connected, and run coordinatively.

There are three sections in this chapter. Section 1 deals with failure detection and isolation, which are essential to the entire controller. The analysis and design formulation of this subject have been made before, but their truly successful design was reported only in Tsui [1989]. Section 2 introduces adaptive failure accommodation, which is uniquely enabled by the failure detection and isolation capability of Sec. 1. Section 3 deals with the effect and corresponding treatment of model uncertainty and measurement noise during failure detection and isolation.

10.1 FAILURE DETECTION AND ISOLATION

10.1.1 Problem Formulation and Analysis

In this book, system failure is modeled as an additional signal $\mathbf{d}(t)$ to the plant system's state space model (1.1a):

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) + \mathbf{d}(t) \tag{10.1}$$

We name $\mathbf{d}(t)$ "failure signal." If failure is free, then $\mathbf{d}(t) = 0$ (or is sufficiently small). If failure occurs, then some among the n elements of $\mathbf{d}(t)$ become nonzero time functions.

Because (1.1a) is a combined description of n system states (or system components), and each system state (or component) is described mainly by its corresponding first-order differential equation in (1.1a) or (10.1), we identify each element of failure signal $\mathbf{d}(t)$ to its corresponding system state component. For example, the nonzero occurrence of the i-th element of $\mathbf{d}(t)$ implies the failure occurrence to the i-th system state component, which will then be called as a "failed state." The nonzero elements of $\mathbf{d}(t)$ are not presumed to have any specific waveform. Hence this failure description is very general.

Failure detection and isolation in this chapter are used to detect the nonzero occurrence of $\mathbf{d}(t)$ and isolate which element of $\mathbf{d}(t)$ is nonzero. In practice, the second purpose is much harder to achieve than the first.

Like observers, the failure detection and isolation are achieved by using the information of plant system inputs and outputs. However, a set of independent but coordinatively designed failure detectors are designed. The number of failure detectors is determined by

$$k = \binom{n}{q} = \frac{n!}{(n-q)!q!} \tag{10.2}$$

where n is the plant system order, and q must be less than the number of plant output measurements m. This requirement of q is drawn from the design feasibility, as will be shown in Step 2 of design algorithm 10.1 in the next subsection.

Parameter q also indicates the maximum number of simultaneous nonzero elements of $\mathbf{d}(t)$ (or the number of simultaneous component failures) this set of failure detectors can isolate. This failure detection and isolation capability is achieved based on the following special properties of the failure detectors.

Each of the k failure detectors has the following structure

$$\dot{\mathbf{z}}_i(t) = F_i\mathbf{z}_i(t) + L_i\mathbf{y}(t) + T_iB\mathbf{u}(t) \tag{10.3a}$$

$$e_i(t) = \mathbf{n}_i\mathbf{z}_i(t) + \mathbf{m}_i\mathbf{y}(t) \qquad i = 1, \ldots, k \tag{10.3b}$$

where the single output $e_i(t)$ is called the "residual signal." It is required that the residual signals all be zero if failure is free ($\mathbf{d}(t) = 0$). It is also required that for each possible set of nonzero elements of $\mathbf{d}(t)$, a *unique and preset* zero/nonzero pattern among the k residual signals be produced. Thus the occurrence of a set of nonzero elements of $\mathbf{d}(t)$ is detected and isolated instantly once its corresponding zero/nonzero pattern of residual signals is formed.

To satisfy the above requirement, it is required by our design that each of the k residual signals must be zero when its corresponding and preset set of q state components has failed, and must be nonzero for any of the other state component failures. We therefore name the failure detectors of (10.3) “robust failure detectors” because each of them is robust (or insensitive) toward its corresponding set of q state component failures. This way, up to q simultaneous state component failures can always be detected and isolated.

Example 10.1

Let a plant system have four states and three output measurements ($n = 4$ and $m = 3$). We will analyze the following two cases for (A) $q = 1$ and (B) $q = 2 (q < m = 3)$, respectively.

(A) $q = 1$: From (10.2), four robust failure detectors are needed. Each failure detector must be robust to q $(= 1)$ state component failure.

In Table 10.1, the symbol “X” represents nonzero and is regarded as “1” (or “TRUE”) in the third column of logic operations, and “$\cap$” stands for the “AND” logic operation. It is clear that if the residual signals behave as desired, each one of the four state component failures can be isolated.

(B) $q = 2$: From (10.2), six robust failure detectors are needed. Each failure detector must be robust to q $(= 2)$ state component failures.

The logic operation of Table 10.2 can isolate not only one failure as listed, but also two simultaneous failures. For example, the failure situation of $d_1(t) \neq 0$ and $d_2(t) \neq 0$ can be isolated by its *unique* residual signal pattern $e_2 \cap e_3 \cap e_4 \cap e_5 \cap e_6$.

The above design idea can easily be extended to the case where among n state components, only p state components can fail. The only adaptation

Table 10.1 Isolation of One State Component Failure for a Four-State Component System

Failure situation	Residual signals				
$\mathbf{d}(t) = [d_1 \; d_2 \; d_3 \; d_4]'$	e_1	e_2	e_3	e_4	Logic policy for failure isolation
$d_1(t) \neq 0$	0	X	X	X	$(d_1(t) \neq 0) = e_2 \cap e_3 \cap e_4$
$d_2(t) \neq 0$	X	0	X	X	$(d_2(t) \neq 0) = e_1 \cap e_3 \cap e_4$
$d_3(t) \neq 0$	X	X	0	X	$(d_3(t) \neq 0) = e_1 \cap e_2 \cap e_4$
$d_4(t) \neq 0$	X	X	X	0	$(d_4(t) \neq 0) = e_1 \cap e_2 \cap e_3$

Table 10.2 Isolation of Up to Two State Component Failures for a Four-State Component System

Failure situation $\mathrm{d}(t) = [d_1\ d_2\ d_3\ d_4]'$	e_1	e_2	e_3	e_4	e_5	e_6	Logic policy for failure isolation
$d_1(t) \neq 0$	0	0	0	X	X	X	$(d_1(t) \neq 0) = e_4 \cap e_5 \cap e_6$
$d_2(t) \neq 0$	0	X	X	0	0	X	$(d_2(t) \neq 0) = e_2 \cap e_3 \cap e_6$
$d_3(t) \neq 0$	X	0	X	0	X	0	$(d_3(t) \neq 0) = e_1 \cap e_3 \cap e_5$
$d_4(t) \neq 0$	X	X	0	X	0	0	$(d_4(t) \neq 0) = e_1 \cap e_2 \cap e_4$

for this case is to design a combination of p over q robust failure detectors [instead of a combination of n over q as in (10.2)].

From the above analysis, the key to the success of this failure detection and isolation scheme is the generation of the desired zero/nonzero residual signal patterns. This difficult yet essential requirement is analyzed by the following theorem.

Theorem 10.1

To achieve the desired properties of robust failure detectors, each detector parameter $(F_i,\ T_i,\ L_i,\ \mathbf{n}_i,\ \mathbf{m}_i,\ i = 1, \ldots, k)$ must satisfy the following four conditions [Ge and Fang, 1988]:

1. $T_iA - F_iT_i = L_iC$ (F_i is stable)
   (so that $\mathbf{z}_i(t) \Rightarrow T_i\mathbf{x}(t)$ if $\mathbf{d}(t) = 0$) (10.4)
2. $0 = \mathbf{n}_iT_i + \mathbf{m}_iC$ (so that $e_i(t) \Rightarrow 0$ if $\mathbf{d}(t) = 0$) (10.5)
3. The q columns of $T_i = 0$ [so that $e_i(t)$ still $= 0$, even if the corresponding q elements of $\mathbf{d}(t) \neq 0$] (10.6)
4. Each of the remaining $n - g$ columns of $T_i \neq 0$ [so that $e_i(t) \neq 0$, if any of the remaining $n - q$ elements of $\mathbf{d}(t) \neq 0$] (10.7)

The statements inside parentheses describe the physical meaning of the corresponding condition.

Proof

Condition (10.4) and its physical meaning have been proved in Theorem 3.2. Although the eigenvalues of F_i can be arbitrarily assigned, they should have negative and sufficiently negative real parts to guarantee convergence and fast enough convergence from $\mathbf{z}_i(t)$ to $T_i\mathbf{x}(t)$.

Condition (10.5) is obviously necessary and sufficient for $e_i(t) \Rightarrow 0$, based on (10.3b) and on the assumption that $\mathbf{z}_i(t) \Rightarrow T_i\mathbf{x}(t)$ and $\mathbf{y}(t) = C\mathbf{x}(t)$.

When $\mathbf{d}(t) \neq 0$, (10.4) implies (see the proof of Theorem 3.2)

$$\dot{\mathbf{z}}_i(t) - T_i\dot{\mathbf{x}}(t) = F_i[\mathbf{z}_i(t) - T_i\mathbf{x}(t)] - T_i\mathbf{d}(t) \tag{10.8}$$

or

$$\mathbf{z}_i(t) - T_i\mathbf{x}(t) = -\int_{t_0}^{t} \mathrm{e}^{Fi(t-\tau)} T_i\mathbf{d}(\tau)\, d\tau \tag{10.9}$$

where t_0 is the failure occurrence time and it is assumed [from (10.4)] that $\mathbf{z}_i(t_0) = T_i\mathbf{x}(t_0)$.

From (10.9), (10.6) [$T_i\mathbf{d}(\tau) = 0\ \forall\tau$ and for the q nonzero elements of $\mathbf{d}(\tau)$] guarantees that $\mathbf{z}_i(t)$ still equals $T_i\mathbf{x}(t)$ at $t > t_0$. Then (10.3b) and (10.5) guarantee that $e_i(t)$ still equals 0 at $t > t_0$.

Equation (10.9) also implies that if (10.7) [or $T_i\mathbf{d}(\tau) \neq 0$] holds, then $\mathbf{z}_i(t) - T_i\mathbf{x}(t) \neq 0$ at $t > t_0$ generally. Consequently, (10.3b) and (10.5) imply $e_i(t) \neq 0$ at $t > t_0$ for most cases.

Together, the physical meanings of the four conditions imply the satisfaction of the required properties of robust failure detectors.

10.1.2 Design Algorithm and Example

The failure detection and isolation problem having been formulated as the four conditions of Theorem 10.1, the real challenge now is *how* to design the robust failure detectors (F_i, T_i, L_i, $\mathbf{n}_i$, $\mathbf{m}_i$, $i = 1, \ldots, k$) which can satisfy (or best satisfy) these four conditions.

The inspection of the four conditions (10.4)–(10.7) shows that parameter T_i is the key parameter which uniquely appears in all four conditions. The solution T_i of (10.4) has already been derived in Algorithm 5.3 of Chap. 5 with many design applications as listed in Fig. 5.1. Hence the remaining three conditions (10.5)–(10.7) can be considered as another application of this solution of (10.4).

Based on the distinct advantages of the solution of (10.4) of Algorithm 5.3, a really systematic and general design algorithm for (10.4)–(10.7) is developed as follows [Tsui, 1989].

Algorithm 10.1

Computation of the solution of (10.4)–(10.7)

Step 1: Set the robust failure detector orders $r_i = n - m + 1$ $(i = 1, \ldots, k)$, and select the eigenvalues of F_i according to the corresponding proof of Theorem 10.1. Then use Step 1 of Algorithm 5.3 to compute the basis vector matrix $D_{ij} \in R^{m \times n}$ for each row $\mathbf{t}_{ij}$ of the solution matrix T_i of (10.4). Thus

$$\mathbf{t}_{ij} = \mathbf{c}_{ij} D_{ij} \quad (j = 1, \ldots, r_i, \; i = 1, \ldots, k) \tag{10.10}$$

where $\mathbf{c}_{ij} \in R^{1 \times m}$ are completely free.

Step 2: Compute $\mathbf{c}_{ij}$ so that

$$\mathbf{c}_{ij}[\text{the } q \text{ columns of } D_{ij}]_{m \times q} = 0 \qquad j = 1, \ldots, r_i, \quad i = 1, \ldots, k \tag{10.11}$$

is satisfied, where the q columns are preset for the i-th failure detector (such as in Table 10.2). The nonzero solution $\mathbf{c}_{ij}$ of (10.11) always exists because q is set to be less than m.

Step 3: Compute (10.10). Then use Step 3 of Algorithm 5.3 [or (5.16)] to compute L_i.

Compute the failure detector parameters $\mathbf{n}_i$ and $\mathbf{m}_i$ to satisfy

$$[\mathbf{n}_i : \mathbf{m}_i] \left[\begin{array}{c} \mathbf{c}_{i1}D_{i1} \\ \vdots \\ \mathbf{c}_{iri}D_{iri} \\ \hdashline C \end{array}\right] \begin{array}{l} \}r_i \\ \\ \\ \}m \end{array} = 0 \qquad (10.12)$$

Because the matrix of (10.12) has n columns, the nonzero solution $[\mathbf{n}_i : \mathbf{m}_i]$ of (10.12) always exists for $r_i = n - m + 1$.

It is obvious that (10.10)–(10.12) of the above algorithm guarantee conditions (10.4), (10.5), and (10.6), respectively. This algorithm is based on Algorithm 5.3 which satisfies (10.4), and then uses the remaining freedom of (10.4) (or the remaining design freedom of the dynamic part of robust failure detector $\mathbf{c}_{ij}$) to satisfy (10.6). The design freedom ($\mathbf{n}_i$ and $\mathbf{m}_i$) of the output part of robust failure detector has been fully used in (10.12) to satisfy (10.5).

However, experience shows that for many plant systems, (10.7) cannot be satisfied for all k failure detectors after (10.4)–(10.6) are satisfied, especially when q is set at its maximum possible value $m - 1$.

The reason for this situation can be explained as follows. First of all, (10.6) is equivalent to equation $T_i\overline{B}_i = 0$, where $\overline{B}_i$ is an $n \times q$ dimensional matrix which picks q columns (out of n columns) of T_i. Secondly, because (10.6) implies that at least q columns of T_i will be zero, (10.7) requires the rows of T_i be linearly independent of each other. Now these two modified requirements of (10.6) and (10.7), together with (10.4), form the design requirement of unknown input observers which do not generally exist (see Sec. 4.3). It is even harder for an exhaustive k combinations of failure detectors (corresponding to k different combinations of $\overline{B}_i$ matrices) to generally exist.

As shown in Sec. 6.2, this book has presented the first general and systematic design solution of (4.1) and (4.3), which are equivalent to (10.4) and (10.6). Hence Algorithm 10.1 also has generally satisfied (10.4) and (10.6) for the *first time* (provided that $m > q$).

Fortunately, (10.4) and (10.6) are, in fact, the most important and most difficult requirements among the four requirements of Theorem 10.1, because (10.5) can always be satisfied by detector parameters $[\mathbf{n}_i : \mathbf{m}_i]$ for sufficiently large r_i [see (10.12)], while at least some nonzero columns of (10.7) always appear automatically. Therefore, even if the k exact solutions of (10.4) to (10.7) (for $i = 1, \ldots, k$) do not all exist, simple adjustment can easily be made, based on the general solution of (10.4) to (10.6) of Algorithm 10.1, to construct a system with partially fulfilled failure detection and isolation capability.

Example 10.2 [Tsui, 1993c]

Let the plant system be

$$(A,\ B,\ C) = \left(\begin{bmatrix} -20.95 & 17.35 & 0 & 0 \\ 66.53 & -65.89 & -3.843 & 0 \\ 0 & 1473 & 0 & -67,420 \\ 0 & 0 & -0.00578 & -0.05484 \end{bmatrix}, \right.$$

$$\left. \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)$$

This is the state space model of an actual automotive powertrain system [Cho and Paolella, 1990]. The four states are engine speed, torque-induced turbine speed, driving axle torque (sum of both sides), and wheel rotation speed. The control input is an engine-indicated torque. This example will be used throughout this chapter.

Because $m = 3$, we let $q = m - 1 = 2$. Thus a total of $k = 6$ robust failure detectors will be designed by Algorithm 10.1 and according to Table 10.2.

In Step 1, we let $r_i = n - m + 1 = 2$ and let the common dynamic matrix of the six failure detectors be randomly chosen as

$$F_i = \begin{bmatrix} -10 & 0 \\ 0 & -20.7322 \end{bmatrix}, \qquad i = 1, \ldots, 6$$

Following Step 1 of Algorithm 5.3 (5.10b), the r_i $(= 2)$ basis vector matrices are common to all six failure detectors as

$$D_{i1} = \begin{bmatrix} 0.3587 & 0.8713 & 0.0002 & 0.3348 \\ -0.0005 & 0.0002 & 1 & -0.0005 \\ -0.9334 & 0.3348 & -0.0005 & 0.1287 \end{bmatrix}, \qquad i = 1, \ldots, 6$$

and

$$D_{i2} = \begin{bmatrix} 0.1741 & 0.9697 & 0 & 0.1715 \\ -0.0003 & 0 & 1 & -0.0003 \\ -0.9847 & 0.1715 & -0.0003 & 0.0303 \end{bmatrix}, \qquad i = 1, \ldots, 6$$

Because the given form of matrix C differs from the C of the block-observable Hessenberg form (5.5) by a column permutation [the last $n-m$ column of C of (5.5) becomes the third column in this C], the D_{ij} matrices are computed based on the third column of matrix $(A-\lambda_j I)$, instead of on the last $n-m$ column according to (5.10b).

In Step 2 of Algorithm 10.1, the solution of (10.11) corresponding to each of the six sets of q zero-columns of Table 10.2 is:

$$\begin{aligned}
(\mathbf{c}_{11},\ \mathbf{c}_{12}) &= ([0\ \ -1\ \ 0.0006],\ [0\ \ -1\ \ 0.0003])\\
(\mathbf{c}_{21},\ \mathbf{c}_{22}) &= ([0.0015\ \ 1\ \ 0],\ [0.0015\ \ 1\ \ 0])\\
(\mathbf{c}_{31},\ \mathbf{c}_{32}) &= ([0.9334\ \ 0\ \ 0.3587],\ [0.9847\ \ 0\ \ 0.1741])\\
(\mathbf{c}_{41},\ \mathbf{c}_{42}) &= ([-0.3587\ \ 0.0005\ \ 0.9334],\ [-0.1741\ \ 0.0003\ \ 0.9847])\\
(\mathbf{c}_{51},\ \mathbf{c}_{52}) &= ([-0.3587\ \ 0.0005\ \ 0.9334],\ [-0.1741\ \ 0.0003\ \ 0.9847])\\
(\mathbf{c}_{61},\ \mathbf{c}_{62}) &= ([-0.3587\ \ 0.0005\ \ 0.9334],\ [-0.1741\ \ 0.0003\ \ 0.9847])
\end{aligned}$$

The inspection of the above result shows that the last three failure detectors are redundant and can therefore be simplified to only one. So only four failure detectors $(i=1,\ldots,4)$ will be computed.

In Step 3, compute T_i according to (10.10) $(i=1,\ldots,4)$. Then the corresponding L_i can be computed based on all columns of Eq. (10.4) except the third column $(i=1,\ldots,4)$ (see the explanation before the result of Step 2). The result is:

$$T_1 = \begin{bmatrix} 0 & 0 & 0.0006 & 1 \\ 0 & 0 & 0.0003 & 1 \end{bmatrix} \qquad L_1 = \begin{bmatrix} 0 & 0.8529 & -29.091 \\ 0 & 0.3924 & 3.715 \end{bmatrix}$$

$$T_2 = [\,0 \quad 0.0015 \quad 0 \quad -1\,] \qquad L_2 = [\,0.1002 \quad -0.0842 \quad -9.9451\,]$$

$$T_3 = \begin{bmatrix} 0 & 0.9334 & 0.3587 & 0 \\ 0 & 0.9847 & 0.1741 & 0 \end{bmatrix} \qquad L_3 = \begin{bmatrix} 62 & 476 & -24,185 \\ 66 & 213 & -11,740 \end{bmatrix}$$

and

$$T_4 = [\,-1 \quad -0 \quad 0 \quad -0\,] \qquad L_4 = [\,10.95 \quad -17.35 \quad 0\,]$$

In the above result, the two rows of matrix T_i $(i=2,4)$ are the same. Hence we have adjusted $r_i=1, F_i=-10$, and parameters $[T_i : L_i]$ as the first row of their original values, for $i=2$ and 4, respectively.

It can be verified that (10.4) and (10.6) are satisfied.

In Step 3 of Algorithm 10.1, we solve (10.12) such that

$$\begin{aligned}
[\mathbf{n}_1 : \mathbf{m}_1] &= [\,0.3753 \quad -0.8156 : 0 \quad 0 \quad 0.4403\,] \\
[\mathbf{n}_2 : \mathbf{m}_2] &= [\,0.7071 : 0 \quad -0.0011 \quad 0.7071\,] \\
[\mathbf{n}_3 : \mathbf{m}_3] &= [\,0.394 \quad -0.8116 : 0 \quad 0.4314 \quad 0\,]
\end{aligned}$$

and

$$[\mathbf{n}_4 : \mathbf{m}_4] = [\,1 : 1 \quad 0 \quad 0\,]$$

It can be verified that (10.5) is also satisfied with this set of parameters.

However, condition (10.7) of Theorem 10.1 is satisfied only for $i = 1, 2, 3$. For $i = 4$, there are three zero columns in matrix T_i, and the requirement (10.7) of Theorem 4.1 that there are $n - q\ (= 2)$ nonzero columns in T_i is not met. As a result, the desired properties of Table 10.2 cannot be fully achieved. Instead, we simply adjust our design and Table 10.2 to have the following partially fulfilled failure isolation capability.

It can be easily verified from Table 10.3 that the three pairs of simultaneous state component failures (d_1 and d_2, d_1 and d_3, and d_1 and d_4) can be isolated by the logic operation of Table 10.3. These three failure situations are isolated by $e_2 \cap e_3 \cap e_4$, $e_1 \cap e_3 \cap e_4$, and $e_1 \cap e_2 \cap e_4$, respectively. However, the other three possible pairs of two simultaneous failures (d_2 and d_3, d_2 and d_4, and d_3 and d_4) cannot be isolated. In these three cases, e_4 is zero, but all other three residual signals are nonzero. Hence one can learn from this residual signal pattern only that all elements of $\mathbf{d}(t)$ except $d_1(t)$ can be nonzero, but one cannot isolate which two state failures among the three are occurring.

Considering the difficulty of instant isolation of all possible pairs of two simultaneous unknown state component failures of this four-state system, the above failure isolation capability of Table 10.3, though not as good as Table 10.2, is still remarkable.

Table 10.3 Failure Isolation of Example 10.2

Failure situation	Residual signals				
$\mathbf{d}(t) = [d_1\ d_2\ d_3\ d_4]'$	e_1	e_2	e_3	e_4	Logic policy for failure isolation
$d_1(t) \neq 0$	0	0	0	X	$(d_1(t) \neq 0) = e_4$
$d_2(t) \neq 0$	0	X	X	0	$(d_2(t) \neq 0) = e_2 \cap e_3$
$d_3(t) \neq 0$	X	0	X	0	$(d_3(t) \neq 0) = e_1 \cap e_3$
$d_4(t) \neq 0$	X	X	0	0	$(d_4(t) \neq 0) = e_1 \cap e_2$

The above failure isolation capability of Table 10.3 is still better than that of Table 10.1, where q is set to be 1, because the system of Table 10.3 can isolate three additional pairs of two simultaneous failures, whereas the system of Table 10.1 cannot. Hence a large q ($<$ m) may be generally recommended even though the isolation of all situations of q simultaneous state component failures may be impossible. The value of q should also be chosen such that k of (10.2) (and the amount of failure detector design freedom) is maximized.

10.2 ADAPTIVE STATE FEEDBACK CONTROL FOR FAILURE ACCOMMODATION [Tsui, 1997]

The purpose of detecting and diagnosing disease is to apply the corresponding and appropriate cure to that disease. Likewise, the purpose of failure detection and isolation is to apply the corresponding and appropriate control to that failure situation. Conversely, a really effective failure accommodation control must be adaptive according to each failure situation.

The failure accommodation control of this chapter is realized by the feedback of two signals—the states $\mathbf{z}_i(t)$ of the robust failure detectors ($i = 1, \ldots, k$) and the plant system output measurements. The feedback gain is adaptive based on the particular and isolated failure situation. We therefore call this control "adaptive." It is obvious that the static feedback gains of this chapter can be most easily and quickly adapted, in either design or implementation.

From Theorem 10.1, the failure detector states $\mathbf{z}_i(t)$ should equal $T_i\mathbf{x}(t)$ ($i = 1, \ldots, k$) before failure occurs. According to the design of robust failure detectors, when q or less than q state components fail, there is at least one robust failure detector whose state [say $\mathbf{z}_i(t)$] still equals $T_i\mathbf{x}(t)$. In addition, some elements of the plant system output measurement $\mathbf{y}(t)$ can also be robust to (or independant of) the failed system states. Both signals are reliable and can be used to control and accommodate the failure.

As discussed in Sec. 4.1, the static gains on $\mathbf{z}_i(t)$ and $\mathbf{y}(t)$ can be considered as state feedbacks (or constrained state feedbacks). We therefore call the failure accommodation control of this chapter "adaptive state feedback control." From the argument of Subsection 2.2.1, the state feedback control is the most powerful among the general forms of control.

This control is uniquely enabled by the information (in both failure isolation decision and plant system state estimation) provided by the failure detection and isolation system of Sec. 10.1.

There are distinct advantages of this failure control scheme.

(a) It is very effective. First of all, the state feedback control is most generally powerful. Secondly, this control is specifically adapted toward each isolated failure situation. Finally, this control is very timely. The failure detection *and* isolation is instant when the zero/nonzero pattern of residual signals is formed upon failure occurrence. The generation of the corresponding control signal, or the switching on of the corresponding static gains on the available signals $\mathbf{z}(t)$ and $\mathbf{y}(t)$, can also be instant.

(b) It can be very easily and simply designed and implemented. The static gains can be designed off-line and can be switched around on-line without worrying about the initial conditions and the transients of the controller.

Finally, it is obvious that this adaptive failure control scheme is *uniquely* enabled by the failure isolation capability of Sec. 10.1.

Example 10.3

Based on the design result of Example 10.1, the ten isolatable failure situations and their respective unfailed plant system states and feedback control signals are listed in Table 10.4.

In Table 10.4, the feedback gain K_i is designed based on the understanding of the corresponding failure situation S_i $(i = 1, \ldots, 10)$. The signal $\bar{\mathbf{y}}_i(t)$, which must be robust to the failed system states of the corresponding failure situation S_i $(i = 1, \ldots, 10)$, can be wholly, partly, or

Table 10.4 Ideal Adaptive Failure Control for a Fourth-Order System with Up to Two Simultaneous State Component Failures

Failure situation	Unfailed states	Control signal
$S_1 : d_1(t) \neq 0$	$x_2(t), x_3(t), x_4(t)$	$K_1[\mathbf{z}_1' : \mathbf{z}_2' : \mathbf{z}_3' : \bar{\mathbf{y}}_1']'$
$S_2 : d_2(t) \neq 0$	$x_1(t), x_3(t), x_4(t)$	$K_2[\mathbf{z}_1' : \mathbf{z}_4' : \mathbf{z}_5' : \bar{\mathbf{y}}_2']'$
$S_3 : d_3(t) \neq 0$	$x_1(t), x_2(t), x_4(t)$	$K_3[\mathbf{z}_2' : \mathbf{z}_4' : \mathbf{z}_6' : \bar{\mathbf{y}}_3']'$
$S_4 : d_4(t) \neq 0$	$x_1(t), x_2(t), x_3(t)$	$K_4[\mathbf{z}_3' : \mathbf{z}_5' : \mathbf{z}_6' : \bar{\mathbf{y}}_4']'$
$S_5 : d_1(t) \neq 0, d_2(t) \neq 0$	$x_3(t), x_4(t)$	$K_5[\mathbf{z}_1' : \bar{\mathbf{y}}_5']'$
$S_6 : d_1(t) \neq 0, d_3(t) \neq 0$	$x_2(t), x_4(t)$	$K_6[\mathbf{z}_2' : \bar{\mathbf{y}}_6']'$
$S_7 : d_1(t) \neq 0, d_4(t) \neq 0$	$x_2(t), x_3(t)$	$K_7[\mathbf{z}_3' : \bar{\mathbf{y}}_7']'$
$S_8 : d_2(t) \neq 0, d_3(t) \neq 0$	$x_1(t), x_4(t)$	$K_8[\mathbf{z}_4' : \bar{\mathbf{y}}_8']'$
$S_9 : d_2(t) \neq 0, d_4(t) \neq 0$	$x_1(t), x_3(t)$	$K_9[\mathbf{z}_5' : \bar{\mathbf{y}}_9']'$
$S_{10} : d_3(t) \neq 0, d_4(t) \neq 0$	$x_1(t), x_2(t)$	$K_{10}[\mathbf{z}_6' : \bar{\mathbf{y}}_{10}']'$

not part of the output $\mathbf{y}(t)$. The actual design of K_i is introduced generally in Chaps 8 and 9.

In actual practice, all six robust failure detectors run simultaneously and all six detector states are available all the time (so are the plant system output measurements). Once a failure situation S_i is detected and isolated (it must be one of the ten in Table 10.4), the corresponding control signal (with gain K_i according to Table 10.4) will automatically be switched on.

Example 10.4 Failure Accommodation for the System of Example 10.2

The failure isolation capability of Table 10.2 is not as fully achievable in Example 10.2 as in Example 10.1. For such cases, Table 10.4 (which corresponds to Example 10.1 and Table 10.2) must be adjusted as follows.

First, all redundant failure detectors and their respective states $\mathbf{z}_4, \mathbf{z}_5$, and $\mathbf{z}_6$ are reduced to $\mathbf{z}_4$ only.

Second, the failure situations S_i $(i = 8, 9, 10)$ cannot be isolated and hence cannot be specifically controlled.

Third, for the specific case of Example 10.2, the $\bar{\mathbf{y}}_i(t)$ signals of Table 10.4 $(i = 1, \ldots, 7)$ can be specified as follows:

$$\begin{aligned}
&\bar{\mathbf{y}}_1(t) = [\,y_2(t) \quad y_3(t)\,]' \qquad \bar{\mathbf{y}}_2(t) = [\,y_1(t) \quad y_3(t)\,]' \\
&\bar{\mathbf{y}}_3(t) = \mathbf{y}(t) \qquad \bar{\mathbf{y}}_4(t) = [\,y_1(t) \quad y_2(t)\,]' \\
&\bar{\mathbf{y}}_5(t) = y_3(t) \qquad \bar{\mathbf{y}}_6(t) = [\,y_2(t) \quad y_3(t)\,]' \qquad \bar{\mathbf{y}}_7(t) = y_2(t)
\end{aligned}$$

In making the actual failure accommodation control signal, we also make sure that the signals used to produce this control are linearly independent (or not redundant) of each other. When two signals are redundant, the output measurement signals $[\bar{\mathbf{y}}_i(t)'\mathrm{s}]$ should be used in priority against the failure detector states $[\mathbf{z}_i(t)'\mathrm{s}]$ because the former are more reliable as linear combinations of plant system states. For example, $\mathbf{z}_4(t) = T_4\mathbf{x}(t)$ is linearly dependent on $y_1(t)$ and will therefore not be used if $y_1(t)$ is used.

Finally, if there are enough unfailed plant system states available for failure control, an additional adjustment can also be made to Table 10.4 as follows. This adjustment can be most important.

Among the unfailed plant system states some may be more strongly influenced by the failed states than others. These unfailed plant system states are therefore less reliable than the others for failure control and should not be used to generate failure accommodation control signals.

This idea can be easily implemented because the coupling between the system states is very clearly revealed by the system's dynamic matrix A. For example, the magnitude of the element a_{ij} of matrix A indicates how strongly state x_i is influenced by state x_j.

Example 10.5

In the completely adjusted failure accommodation control for Example 10.2, where

$$A = \begin{bmatrix} -20.95 & 17.35 & 0 & 0 \\ 66.53 & -65.89 & -3.843 & 0 \\ 0 & 1437 & 0 & -67,420 \\ 0 & 0 & -0.00578 & -0.0548 \end{bmatrix}$$

Matrix A indicates, for example, that state x_3 is strongly influenced by state x_4 because $|a_{34}| = 67,420$ is large, while state x_4 is weakly influenced by state x_3 because $|a_{43}| = 0.00578$ is small (matrix A is not symmetrical).

Based on the above understanding, we list the specific failure control for the plant system of Example 10.2 in the following. We use three different thresholds (10, 100, and 1) to judge whether a state is strongly influenced by another. For example, if the threshold is 10 in Example 10.2, then matrix A indicates that state x_1 is strongly influenced by x_2 ($|a_{12}| = 17.35 > 10$), while x_2 is weakly influenced by x_3 ($|a_{23}| = 3.843 < 10$). Thus for the three different thresholds, there are three corresponding tables (Table 10.5 to 10.7) which are adjusted from Table 10.4.

In Tables 10.5 to 10.7, the most used information for failure control is from $\mathbf{y}(t)$ and one may wonder what is the use of the failure detector. The cause of this fact is that in this particular example m is large compared to n. In more challenging problems where m is small compared to n, the information of $\mathbf{z}(t)$ will be the main source for failure control.

In the failure situation S_7 of Table 10.5, state x_2 is considered weakly influenced by the failed states (x_1 and x_4) even though the actual influence from x_1 is still quite strong ($|a_{21}| = 66.53 > 10$). This is because the only other unfailed state x_3 is even more strongly influenced by the failed state x_4.

The difference between Table 10.6 and Table 10.5 is for failure situations S_1, S_2, and S_6. Table 10.6 adds x_2, x_1, and x_2 as states which are weakly influenced by the failed states for these three failure situations, respectively. This is because the corresponding elements of these states in matrix A (66.53, 17.35, and 66.53, respectively) are less than 100. Thus the

Table 10.5 Failure Accommodation Control for Example 10.2 (With Threshold for State Coupling Strength Set as 10)

Failure situation	Unfailed system states	States weakly Influenced by failed states	Adaptive failure control signal
$S_1 : d_1 \neq 0$	x_2, x_3, x_4	x_3, x_4	$K_1[z_1' : y_3]'$
$S_2 : d_2 \neq 0$	x_1, x_3, x_4	x_4	$K_2 y_3(t)$
$S_3 : d_3 \neq 0$	x_1, x_2, x_4	x_1, x_2, x_4	$K_3 \mathbf{y}(t)'$
$S_4 : d_4 \neq 0$	x_1, x_2, x_3	x_1, x_2	$K_4[y_1 : y_2]'$
$S_5 : d_1 \neq 0, d_2 \neq 0$	x_3, x_4	x_4	$K_5 y_3(t)$
$S_6 : d_1 \neq 0, d_3 \neq 0$	x_2, x_4	x_4	$K_6 y_3(t)$
$S_7 : d_1 \neq 0, d_4 \neq 0$	x_2, x_3	x_2	$K_7 y_2(t)$

control signals for these failure situations are based on more information, although this additional information is less reliable.

The difference between Table 10.7 and Table 10.5 is at failure situation S_3, where state x_2 is no longer considered weakly influenced by the failed state (x_3) in Table 10.7. This is because the corresponding element $|a_{23}| = 3.843 > 1$.

Among the seven isolatable failure situations, S_7 has the least reliable information according to our formulation.

Table 10.6 Failure Accommodation Control for Example 10.2 (With Threshold for State Coupling Strength Set as 100)

Failure situation	Unfailed system states	States weakly influenced by failed states	Adaptive failure control signal
$S_1 : d_1 \neq 0$	x_2, x_3, x_4	x_2, x_3, x_4	$K_1[z_1' : y_2 : y_3]'$
$S_2 : d_2 \neq 0$	x_1, x_3, x_4	x_1, x_4	$K_2[y_1 : y_3]'$
$S_3 : d_3 \neq 0$	x_1, x_2, x_4	x_1, x_2, x_4	$K_3 : \mathbf{y}(t)'$
$S_4 : d_4 \neq 0$	x_1, x_2, x_3	x_1, x_2	$K_4[y_1 : y_2]'$
$S_5 : d_1, d_2 \neq 0$	x_3, x_4	x_4	$K_5 y_3(t)$
$S_6 : d_1, d_3 \neq 0$	x_2, x_4	x_2, x_4	$K_6[y_2 : y_3]'$
$S_7 : d_1, d_4 \neq 0$	x_2, x_3	x_2	$K_7 y_2(t)$

Table 10.7 Failure Accommodation Control for Example 10.2 (With Threshold for State Coupling Strength set as 1)

Failure situation	Unfailed system states	States weakly influenced by failed states	Adaptive failure control signal
$S_1 : d_1 \neq 0$	x_2, x_3, x_4	x_3, x_4	$K_1[\mathbf{z}_1' : y_3]'$
$S_2 : d_2 \neq 0$	x_1, x_3, x_4	x_4	$K_2 y_3(t)$
$S_3 : d_3 \neq 0$	x_1, x_2, x_4	x_1, x_4	$K_3[y_1 : y_3]'$
$S_4 : d_4 \neq 0$	x_1, x_2, x_3	x_1, x_2	$K_4[y_1 : y_2]'$
$S_5 : d_1, d_2 \neq 0$	x_3, x_4	x_4	$K_5 y_3(t)$
$S_6 : d_1, d_3 \neq 0$	x_2, x_4	x_4	$K_6 y_3(t)$
$S_7 : d_1, d_4 \neq 0$	x_2, x_3	x_2	$K_7 y_2(t)$

10.3 THE TREATMENT OF MODEL UNCERTAINTY AND MEASUREMENT NOISE [Tsui, 1994b]

In the previous two sections, a complete failure detection, isolation, and accommodation control system is established. This section discusses the effect and the corresponding treatment of plant system model uncertainty and output measurement noise on the failure detection and isolation part of that system.

To do this, we need to analyze the overall feedback system. It is striking that robust failure detectors $(F_i, T_i, L_i, \ i = 1, \ldots, k)$ and failure accommodation control of Tables 10.4–10.7 are compatible with the feedback compensator (F_0, T_0, L_0) (3.16) in structure. As a result, the normally (assuming failure-free) designed observer feedback compensator (3.16) of Chaps 5 through 9, and the failure detection, isolation, and accommodation system of the previous two sections can be connected in parallel and implemented coordinatively. The combined system can be illustrated in Fig. 10.1, which shows a combined system with a normal feedback compensator and a failure detection, isolation, and accommodation compensator,

where

$\mathbf{r}(t) \triangleq$ external reference input
$\mathbf{d}(t) \triangleq$ failure signal
$\mathbf{n}(t) \triangleq$ output measurement noise signal with $\bar{n}$ as the upper bound of its rms value
$C(sI - A)^{-1}\Delta(s) \triangleq$ plant system model uncertainty with $\overline{\Delta}$ as the upper bound of scalar function $\Delta(s)$

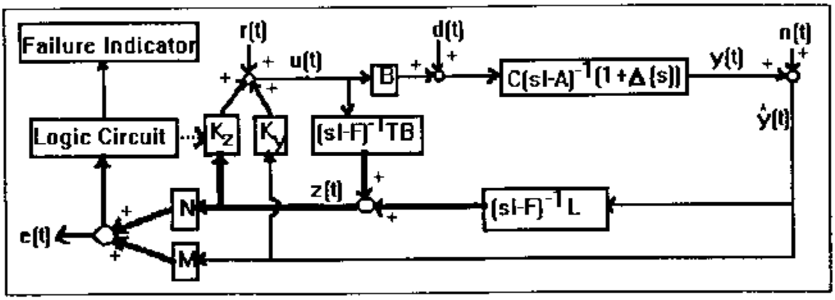


Figure 10.1 Combined system with a normal feedback compensator and a failure detection, isolation, and accommodation compensator.

and for $k = n!/[(n-q)!q!](q = m-1)$ of (10.2),

$$\mathbf{e}(t) \triangleq [e_1(t) \cdots e_k(t)]' \triangleq \text{residual signal vector}$$

$$\mathbf{z}(t) \triangleq [\mathbf{z}_0(t)' : \mathbf{z}_1(t)' : \cdots : \mathbf{z}_k(t)']'$$

$$F \triangleq \operatorname{diag}\{F_0, F_1, \ldots, F_k\}$$

$$N \triangleq \operatorname{diag}\{\mathbf{n}_1, \ldots, \mathbf{n}_k\}(\mathbf{n}_0 \triangleq 0)$$

and

$$T \triangleq \begin{bmatrix} T_0 \\ T_1 \\ \vdots \\ T_k \end{bmatrix}, L \triangleq \begin{bmatrix} L_0 \\ L_1 \\ \vdots \\ L_k \end{bmatrix}, M \triangleq \begin{bmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_k \end{bmatrix} (\mathbf{m}_0 \triangleq 0)$$

The feedback gain $[K_Z : K_y]$ is normally applied to $\mathbf{z}_0(t)$ and $\mathbf{y}(t)$ only, but will be adapted when failure is detected and isolated (see Tables 10.4–10.7 for example). Thus $[K_Z : K_y]$ is modeled as the gain to the entire $\mathbf{z}(t)$ and $\check{\mathbf{y}}(t)$ signals.

Because failure detection and isolation is achieved based on the zero/nonzero pattern of the residual signals (see Tables 10.1–10.3), the effect of model uncertainty $\Delta(s)$ and measurement noise $\mathbf{n}(t)$ is reflected in the zero/nonzero determination of these residual signals.

To analyze this effect, we must first derive the transfer function relationship between $\Delta(s), N(s), R(s)$, and $D(s)$ to $E(s)$, where $N(s)$,

$R(s), D(s)$, and $E(s)$ are the Laplace transforms of $\mathbf{n}(t), \mathbf{r}(t), \mathbf{d}(t)$, and $\mathbf{e}(t)$, respectively. In addition, we also let $X(s), U(s), Y(s)$, and $Z(s)$ be the Laplace transforms of their respective time signals $\mathbf{x}(t), \mathbf{u}(t), \mathbf{y}(t)$, and $\mathbf{z}(t)$.

Theorem 10.2 [Tsui, 1993c]

For small enough $\Delta(s)$, the transfer functions from $\Delta(s), N(s), R(s)$, and $D(s)$ to $E(s)$ are:

$$E(s) = H_{er}(s)\Delta(s)R(s) + H_{ed}(s)[1 + \Delta(s)]D(s) + H_{en}(s)N(s) \qquad (10.13)$$

$$\triangleq E_r(s) + E_d(s) + E_n(s) \qquad (10.14)$$

where the transfer functions $H_{er}(s), H_{ed}(s)$, and $H_{en}(s)$ are fully determined by the parameters of Fig. 10.1.

Proof

Let $G_o(s) \triangleq C(sI - A)^{-1}$ and $G_c(s) \triangleq (sI - F)^{-1}$. Then from Fig. 10.1,

$$\check{Y}(s) = G_o(s)[1 + \Delta(s)][BU(s) + D(s)] + N(s) \qquad (10.15)$$

$$U(s) = K_Z Z(s) + K_y \check{Y}(s) + R(s) \qquad (10.16)$$

$$Z(s) = G_c(s)[TBU(s) + L\check{Y}(s)] \qquad (10.17a)$$

$$\triangleq G_u(s)U(s) + G_y(s)\check{Y}(s) \qquad (10.17b)$$

and

$$E(s) = NZ(s) + M\check{Y}(s) \qquad (10.18)$$

Substituting (10.16) into (10.17b), then

$$Z(s) = [I - G_u(s)K_Z]^{-1}\{[G_u(s)K_y + G_y(s)]\check{Y}(s) + G_u(s)R(s)\} \qquad (10.19a)$$

$$\triangleq H_{zy}(s)\check{Y}(s) + H_{zr}(s)R(s) \qquad (10.19b)$$

Now substituting (10.19b) into (10.16) and then into (10.15),

$$\begin{aligned}\check{Y}(s) = {} & \{I - G_o(s)[1 + \Delta(s)]B[K_Z H_{zy}(s) + K_y]\}^{-1} \\ & \{G_o(s)[1 + \Delta(s)][B(K_Z H_{zr}(s) + I)R(s) + D(s)] + N(s)\}\end{aligned}$$

for small enough $\overline{\Delta}$ [Emami-Naeini and Rock, 1988],

$$\begin{aligned} & \approx \{I - G_o(s)B[K_Z H_{zy}(s) + K_y]\}^{-1} \\ & \quad \{G_o(s)[1 + \Delta(s)][B(K_Z H_{zr}(s) + I)R(s) + D(s)] + N(s)\} && (10.20a) \\ & \underset{=}{\triangle}\ H_{yr}(s)[1 + \Delta(s)]R(s) + H_{yd}(s)[1 + \Delta(s)]D(s) + H_{yn}(s)N(s) && (10.20b)\end{aligned}$$

Finally, substituting (10.19b) into (10.18),

$$\begin{aligned} E(s) & = [NH_{zy}(s) + M]\check{Y}(s) + NH_{zr}(s)R(s) \\ \text{(By 10.20b):} \quad & = [NH_{zy}(s) + M]H_{yr}(s)[1 + \Delta(s)]R(s) + NH_{zr}(s)R(s) \\ & \quad + [NH_{zy}(s) + M]H_{yd}(s)[1 + \Delta(s)]D(s) \\ & \quad + [NH_{zy}(s) + M]H_{yn}(s)N(s) \\ \text{(By 10.5):} \quad & = [NH_{zy}(s) + M]H_{yr}(s)\Delta(s)R(s) \\ & \quad + [NH_{zy}(s) + M]H_{yd}(s)[1 + \Delta(s)]D(s) \\ & \quad + [NH_{zy}(s) + M]H_{yn}(s)N(s) && (10.21) \\ & \underset{=}{\triangle}\ H_{er}(s)\Delta(s)R(s) + H_{ed}(s)[1 + \Delta(s)]D(s) \\ & \quad + H_{en}(s)N(s) && (10.13)\end{aligned}$$

It is useful to notice that a moment before the failure occurrence,

$$\begin{aligned} U(s) & = K_0 Z_0(s) + K_y \check{Y}(s) + R(s) \\ & = [K_0 : 0 \ldots 0]Z(s) + K_y \check{Y}(s) + R(s)\end{aligned} \qquad (10.22)$$

Let us partition $G_u(s)$ and $G_y(s)$ of (10.17) such that

$$G_u(s) \underset{=}{\triangle} [G_{u0}(s)' : \ldots : G_{uk}(s)']'$$

and

$$G_y(s) \underset{=}{\triangle} [G_{y0}(s)' : \ldots : G_{yk}(s)']' \qquad (10.23a)$$

then

$$G_{ui}(s) = (sI_{ri} - F_i)^{-1}T_iB, \qquad i = 0, 1, \ldots, k \tag{10.23b}$$

$$G_{yi}(s) = (sI_{ri} - F_i)^{-1}L_i, \qquad i = 0, 1, \ldots, k \tag{10.23c}$$

Based on (10.22) and (10.23), in (10.19), the term

$$[I - G_u(s)K_Z]^{-1}G_u(s) = \begin{bmatrix} I_{r0} - G_{u0}(s)K_0 & 0 & \cdots & 0 \\ -G_{u1}(s)K_0 & I_{r1} & \ddots & : \\ : & 0 & \ddots & 0 \\ -G_{uk}(s)K_0 & & & I_{rk} \end{bmatrix}^{-1} \begin{bmatrix} G_{u0}(s) \\ G_{u1}(s) \\ \vdots \\ G_{uk}(s) \end{bmatrix}$$

Now each block of $H_{zy}(s)$ of (10.19b)

$$H_{zy}(s) \overset{\triangle}{=} [H_{zy0}(s)' : H_{zy1}(s)' : \ldots : H_{zyk}(s)']' \tag{10.24a}$$

can be *separately* expressed as

$$H_{zyi}(s)$$
$$= \begin{cases} [I_{r0} - G_{u0}(s)K_0]^{-1}[G_{u0}(s)K_y + G_{y0}(s)], & if\, i = 0 \qquad (10.24b) \\ G_{ui}(s)K_0[I_{r0} - G_{u0}(s)K_0]^{-1}[G_{u0}(s)K_y + G_{y0}(s)] & \\ +G_{ui}(s)K_y + G_{yi}(s) & if\, i \neq 0 \qquad (10.24c) \end{cases}$$

Equation (10.24) implies that each of the k rows of the term $NH_{zy}(s) + M$ of $H_{er}(s), H_{ed}(s)$, and $H_{en}(s)$ of (10.21) and (10.13) can be separately and explicitly expressed. Thus each of the k residual signals has its own explicit transfer function from $R(s), D(s)$, and $N(s)$.

A distinct and important feature of the normal feedback compensator of this book is $T_0B = 0$. Thus from (10.23b) $G_{u0}(s) = 0$, and the expression of $H_{zy}(s)$ of (10.24) can be further greatly simplified as

$$H_{zyi}(s)$$
$$= \begin{cases} G_{y0}(s) = (sI_{r0} - F_0)^{-1}L_0 & if\, i = 0 \qquad (10.25a) \\ G_{ui}(s)K_0G_{y0}(s) + G_{ui}(s)K_y + G_{yi}(s) & if\, i \neq 0 \qquad (10.25b) \end{cases}$$

Conversely, this simplification adds another significant advantage to the new design approach of this book.

After the transfer function relationships between $E(s)$ and its source signals $R(s), D(s)$, and $N(s)$ are established, we can now have the explicit effect of model uncertainty $\Delta(s)$ and measurement noise $N(s)$ on $E(s)$, and can devise the corresponding treatment of $\Delta(s)$ and $N(s)$.

Theorem 10.2 shows that the effect of model uncertainty and of measurement noise, and the effect of failure can be separated in different terms. Thus in a failure-free situation [$D(s) = 0$ or is very minor] the effect of model uncertainty and measurement noise can be explicitly expressed as

$$E(s) = E_r(s) + E_n(s) \tag{10.26}$$

We therefore set the threshold of nonzero $\mathbf{e}(t)$ as the largest possible value of (10.26):

$$\begin{aligned} J_{\text{th}} \overset{\triangle}{=} \max \|E_r(s) + E_n(s)\| &\leqslant \max_{\omega}\|E_r(j\omega)\| + \max_{\omega}\|E_n(j\omega)\| \\ &\leqslant \max_{\omega}\{\overline{\sigma}[H_{er}(j\omega)]\|R(j\omega)\|\}\overline{\Delta} \\ &\quad + \max_{\omega}\{\overline{\sigma}[H_{en}(j\omega)]\}\overline{n} \end{aligned} \tag{10.27}$$

where $\overline{\sigma}$ stands for the largest singular value.

Although the J_{th} is a threshold on $E(j\omega)$ in the frequency domain, it is directly related to $\mathbf{e}(t)$ by the Parseval theorem. For example, according to Emami–Naeini and Rock [1988], J_{th} of (10.27) can be applied to the rms value of $\mathbf{e}(t)$ with “window” length τ:

$$\|\mathbf{e}\|_{\tau} = \left[\left(\frac{1}{\tau}\right) \int_{t_0}^{t_0+\tau} \mathbf{e}(t)'\mathbf{e}(t)\ dt \right]^{1/2} \tag{10.28}$$

If $\|\mathbf{e}\|_{\tau} < J_{th}$, then the nonzero $\mathbf{e}(t)$ is considered to be caused by the model uncertainty $\Delta(s)$ [with input $R(s)$] and noise $N(s)$ only, but not by failure $D(s)$. Only when $\|\mathbf{e}\|_{\tau} > J_{th}$ can we consider that the nonzero $\mathbf{e}(t)$ is caused by failure $D(s)$. It is reasonable from (10.27) that the more severe the model uncertainty $\overline{\Delta}$ and measurement noise $\overline{n}$, the higher the threshold J_{th}. The actual value of τ should be adjusted in practice [Emami–Naeini and Rock, 1988].

Another important technical adjustment is to test *each* residual signal $e_i(t)$ $(i = 1, \ldots, k)$. This will greatly improve the test resolution because both J_{thi} and $\|e_i\|$ should be much lower than J_{th} and $\|\mathbf{e}\|$, respectively. Fortunately, based on (10.24), the test operation (10.27) and (10.28) can be directly adjusted to

$$J_{thi} = \max_{\omega}\{\|H_{eri}(j\omega)\|\|R(j\omega)\|\}\overline{\Delta} + \max_{\omega}\{\|H_{eni}(j\omega)\|\}\overline{n}, \quad i = 1, \ldots, k \tag{10.29}$$

and

$$\|e_i\|_\tau = \left[\left(\frac{1}{\tau}\right)\int_{t_0}^{t_0+\tau} e_i^2(t)\; dt\right]^{1/2}, \qquad i = 1, \ldots, k \tag{10.30}$$

To summarize, this treatment of model uncertainty and measurement noise is very general and simple. This treatment is again uniquely enabled by the failure detection and isolation scheme of this book, which needs to check only the zero/nonzero pattern of the residual signals.

After the threshold J_{th} for the residual signal $\mathbf{e}(t)$ is established, it is useful to establish the theoretical lower bound of the failure signal strength for guaranteed detection. For simplicity of presentation, this bound will be established on J_{th} instead of the k individual J_{thi}'s. Obviously, this lower bound must cause $\|E(s)\|$ to exceed the threshold J_{th}, or

$$\begin{aligned} &\min\|E(s) = E_d(s) + E_r(s) + E_n(s)\| \\ &> J_{th} \triangleq \max\|E_r(s) + E_n(s)\| \end{aligned} \tag{10.31}$$

Theorem 10.3

For sufficiently strong failure such that Emami–Naeini and Rock [1988]

$$\min\|E_d(s)\| > \max\|E_r(s) + E_n(s)\| \tag{10.32}$$

the detectable failure $D(s)$ must satisfy

$$\|H_{ed}(s)D(s)\| > \frac{2J_{th}}{(1 - \overline{\Delta})} \tag{10.33}$$

Proof

From (10.32),

$$\begin{aligned}&\min \|E(s) = E_d(s) + E_r(s) + E_n(s)\| \\ &> \min \|E_d(s)\| - \max \|E_r(s) + E_n(s)\|\end{aligned}$$

Hence the inequality that

$$\min \|E_d(s)\| - \max \|E_r(s) + E_n(s)\| > J_{th}(\triangleq \max \|E_r(s) + E_n(s)\|)$$

or

$$\min \|E_d(s)\| > 2 \max \|E_r(s) + E_n(s)\| \triangleq 2J_{th} \tag{10.34}$$

can guarantee the detectable requirement (10.31).

From (10.21)

$$\|E_d(s)\| = \|H_{ed}(s)D(s) + \Delta(s)H_{ed}(s)D(s)\|$$

Hence from $\Delta(s) \leqslant 1$,

$$\|E_d(s)\| \geqslant (1 - \overline{\Delta})\|H_{ed}(s)D(s)\| \tag{10.35}$$

The inequalities (10.35) and (10.34) together prove (10.33).

Guaranteeing failure detection while also guaranteeing that all effects of plant system model uncertainty and measurement noise are not misinterpreted as the effects of failure, is almost impossible in practice. Reflected in Theorem 10.3, the requirement (10.33) is almost impossible to satisfy generally. For example, $H_{edi}(s)$ is most often a row vector. Thus the theoretical minimal value of the left-hand side of (10.33) is 0 and hence cannot satisfy (10.33).

Nonetheless, Theorem 10.3 is still a simple and general theoretical result. Its requirement (10.33) indicates that the larger the model uncertainty $\overline{\Delta}$ and the measurement noise $\overline{n}$, and the higher the threshold J_{th}, the more difficult for $D(s)$ to satisfy (10.33) (or to be guaranteed detectable). This interpretation is certainly reasonable.

Example 10.6

A complete normal feedback control and failure accommodation control design (with treatment of model uncertainty and measurement noise) follows from Example 10.2.

Let us first design a normal feedback compensator (4.10) for the plant system of Example 10.2, with order $r_0 = n - m = 1$:

$$\begin{aligned}(F_0,\, T_0,\, L_0,\, K_0,\, K_y) = {} & (-21.732, [0 \quad 5.656 \quad -0.0003 \quad 1],\\ & [376.3 \quad -250.2 \quad 41.9], 10{,}000,\\ & [70.9 \quad -56{,}552.3 \quad -13{,}393])\end{aligned}$$

where F_0 is arbitrarily assigned with sufficiently negative real part (see the proof of Theorem 3.2), T_0 and L_0 are designed to satisfy (4.1) and (4.3) $(T_0 B = 0)$, and $[K_0 : K_y]$ is designed such that the corresponding control

$$\begin{aligned}\mathbf{u}(t) &= K_0 \mathbf{z}_0(t) + K_y \mathbf{y}(t)\\ &= [K_0 : K_y][T_0' : C']'\mathbf{x}(t) \ \text{(at steady state)}\\ &= [\,70.9 \quad 7.7 \quad -3 \quad -3393\,]\mathbf{x}(t)\end{aligned}$$

can assign eigenvalues $-2.778 \pm j14.19$ and $-5.222 \pm j4.533$ to the feedback system dynamic matrix $A + B[K_0 : K_y][T_0' : C']'$. This set of eigenvalues guarantee the fastest settling time for step response of the corresponding system [D'Azzo and Houpis, 1988].

Because $T_0 B = 0$, we use (10.25a) to compute

$$\begin{aligned}H_{zy0}(s) &= (sI_{r0} - F_0)^{-1} L_0\\ &= \frac{[\,376.3 \quad -250.2 \quad 41.9\,]}{(s + 21.732)}\end{aligned} \tag{10.36}$$

Because the two terms of (10.29) are mathematically quite compatible, we let the first term be zero [or let the model uncertainty $\Delta(s)$ be 0] to simplify the presentation. Then (10.29) only has its second term left

$$J_{thi} = \max_{\omega}\{\|H_{eni}(j\omega)\|\}\bar{n}, \qquad i = 1, \ldots, 4 \tag{10.37}$$

which will be computed in the following, based on the result of Example 10.2.

Because $T_0 B = 0$, we use (10.25b) to compute

$$H_{zy1}(s) = \begin{bmatrix} 0 & 0.8529/(s+10) & -29.091/(s+10) \\ 0 & 0.3924/(s+21.732) & 3.715/(s+21.732) \end{bmatrix}$$

$$H_{zy2}(s) = [0.1002 \quad -0.0842 \quad -9.9451]/(s+10)$$

$$H_{zy3}(s) = \begin{bmatrix} 62/(s+10) & 476/(s+10) - 24{,}185/(s+10) \\ 66/(s+21.732) & 213/(s+21.732) - 11{,}740/(s+21.732) \end{bmatrix}$$

$$H_{zy4}(s) = [-59.96 \quad 56{,}535 \quad 13{,}393]/(s+10)$$

Substituting $H_{zy}(s)$ into the first part of $H_{en}(s)$ of (10.13), we have

$$NH_{zy}(s) + M = \begin{bmatrix} 0 & \frac{3.754}{(s+10)(s+21.732)} & \frac{0.44s^2+0.0217s-171.88}{(s+10)(s+21.732)} \\ \frac{0.07085}{(s+10)} & \frac{-0.0011s-0.07054}{(s+10)} & \frac{0.7071s+0.039}{(s+10)} \\ \frac{-29.14s-4.782}{(s+10)(s+21.732)} & \frac{0.4314s^2+28.36s+2441}{(s+10)(s+21.732)} & \frac{-0.706s-111{,}801.9}{(s+10)(s+21.732)} \\ \frac{s-49.95}{(s+10)} & \frac{56{,}535}{(s+10)} & \frac{13{,}393}{(s+10)} \end{bmatrix}$$

The second part of $H_{en}(s)$ of (10.13), which equals $H_{yn}(s)$ of (10.20), is computed in the following. From (10.36),

$$\begin{aligned} K_Z H_{zy}(s) + K_y &= K_0 H_{zy0}(s) + K_y \\ &= \frac{70.9s + 3{,}764{,}541 \quad -56{,}552s - 3{,}731{,}006 \quad -13{,}393s + 127{,}941}{s+21.732} \\ &\triangleq [b_1(s) \quad b_2(s) \quad b_3(s)] \end{aligned}$$

Let

$$\begin{aligned} C(sI - A)^{-1}B &\triangleq \begin{bmatrix} a_1(s) \\ a_2(s) \\ a_3(s) \end{bmatrix} \\ &= \begin{bmatrix} s^3 + 65.945s^2 + 5274s - 25{,}411 \\ 66.53s^2 + 3.6485s - 25{,}971 \\ 567.4 \end{bmatrix} \Big/ d(s) \end{aligned}$$

where

$$\begin{aligned} d(s) &= \det(sI - A) \\ &= (s^2 + 66.07s + 4145)(s - 0.9076)(s + 21.732) \end{aligned}$$

Now

$$\begin{aligned} H_{yn}(s) &= \{I - C(sI - A)^{-1}B[K_z H_{zy}(s) + K_y]\}^{-1} \\ &= \left\{ 1 - \begin{bmatrix} a_1(s) \\ a_2(s) \\ a_3(s) \end{bmatrix} [\, b_1(s) \quad b_2(s) \quad b_3(s) \,] \right\}^{-1} \\ &= \frac{\begin{bmatrix} 1 - a_2(s)b_2(s) - a_3(s)b_3(s) & a_1(s)b_2(s) & a_1(s)b_3(s) \\ a_2(s)b_1(s) & 1 - a_1(s)b_1(s) - a_3(s)b_3(s) & a_2(s)b_3(s) \\ a_3(s)b_1(s) & a_3(s)b_2(s) & 1 - a_1(s)b_1(s) - a_2(s)b_2(s) \end{bmatrix}}{1 - a_1(s)b_1(s) - a_2(s)b_2(s) - a_3(s)b_3(s)} \end{aligned}$$

Finally, the J_{thi} of (10.37) are computed for the four failure detectors, $i = 1, \ldots, 4$ (Table 10.8).

We have completely designed a normal (failure-free) feedback compensator and a failure detection, isolation, and accommodation system for the plant system of Example 10.2. The second system can treat plant system output measurement noise with given upper bound $\overline{n}$. The treatment of plant system model uncertainty is very similar to that of the plant output measurement noise.

This example also shows that the design results of this book—the normal (failure-free) feedback compensator of Chaps 5 through 9—and the failure detection, isolation, and accommodation system of this chapter can be coordinatively designed and implemented.

Table 10.8 Threshold Treatment of Measurement Noise of the Four Robust Failure Detectors of Example 10.2 and Fig. 10.1

For	$e_1(t)$	$e_2(t)$	$e_3(t)$	$e_4(t)$
$\omega \approx$	2.7	11.6	2.6	0
$J_{th} \approx$	$7.87 \times 10^3 \overline{n}$	$4.66 \times 10^3 \overline{n}$	$50.3 \times 10^6 \overline{n}$	$43.7 \times 10^6 \overline{n}$

EXERCISES

10.1 Consider an observable system with $n = 5$ and $m = 4$.

(a) Construct Tables 10.1 and 10.2 for the failure detection and isolation of this system, and for $q = 1$ and 2, respectively.

(b) Construct a similar table for $q = 3$. Compare the results of $q = 2$ and 3.

Answer: Although the number of robust failure detectors is the same for $q = 2$ and 3, the failure isolation capability is different for $q = 2$ and 3 (the latter is better but is more difficult to design).

(c) Construct Table 10.4 for $q = 1, 2,$ and 3, respectively.

(b) If one plant system state is known to be failure free, then how many robust failure detectors are needed to isolate q simultaneous failures ($q = 1, 2, 3$, respectively)?

10.2 Under the condition that F_i is in diagonal form, how can (10.7) be the sufficient condition of the physical requirement which is expressed inside the parentheses attached to (10.7)?

10.3 Repeat the design of Examples 10.2 and 10.6 with a new set of robust failure detector poles: $\{-1, -2\}$ and a new $F_0 = -10$. Other parameters remain unchanged.

Partial answer:

$$T_1 = \begin{bmatrix} 0 & 0 & 0.0058 & 1 \\ 0 & 0 & 0.0029 & 1 \end{bmatrix}$$

$$T_2 = [0 \quad 0.0015 \quad 0 \quad -1]$$

$$T_3 = \begin{bmatrix} 0 & 0.2518 & 0.9678 & 0 \\ 0 & 0.4617 & 0.8871 & 0 \end{bmatrix}$$

and

$$T_4 = [-1 \quad 0 \quad 0 \quad 0]$$

Appendix A

Relevant Linear Algebra and Numerical Linear Algebra

This appendix introduces the relevant mathematical background to this book. In addition, an attempt is made to use the simplest possible language, even though such a presentation may sacrifice certain degree of mathematical rigor.

The appendix is divided into three sections.

Section 1 introduces some basic results of linear algebra, especially the geometrical meanings and numerical importance of orthogonal linear transformation.

Section 2 describes and analyzes some basic matrix operations which transform a given matrix into echelon form. A special case of the echelon form is triangular form. This operation is the one used most often in this book.

Section 3 introduces a basic result of numerical linear algebra—the singular value decomposition (SVD). Several applications of SVD are also introduced.

A.1 LINEAR SPACE AND LINEAR OPERATORS

A.1.1 Linear Dependence, Linear Independence, and Linear Space

Definition A.1

A set of n vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is linearly dependent if there exists an n-dimensional nonzero vector $\mathbf{c} \triangleq [c_1, \ldots, c_n]' \neq 0$ such that

$$[\mathbf{x}_1 : \ldots : \mathbf{x}_n]\mathbf{c} = \mathbf{x}_1 c_1 + \cdots + \mathbf{x}_n c_n = 0 \tag{A.1}$$

Otherwise, this set of vectors is linearly independent. At least one vector of a set of linear dependent vectors is a linear combination of other vectors in that set. For example, if a set of vectors satisfies (A.1), then

$$\mathbf{x}_i = -\frac{\left(\sum_{i \neq j} \mathbf{x}_j c_j\right)}{c_i}, \qquad \text{if } c_i \neq 0 \tag{A.2}$$

Example A.1

Let a set of two vectors be

$$[\mathbf{x}_1 : \mathbf{x}_2] = \begin{bmatrix} 1 & -2 \\ -1 & 2 \end{bmatrix}$$

Because there exist a vector $\mathbf{c} = [2 \;\; 1]' \neq 0$ such that $[\mathbf{x}_1 : \mathbf{x}_2]\mathbf{c} = 0$, $\mathbf{x}_1$ and $\mathbf{x}_2$ are linearly dependent of each other.

Example A.2

Let another set of two vectors be

$$[\mathbf{x}_3 : \mathbf{x}_4] = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$$

Because only a zero vector $\mathbf{c} = 0$ can make $[\mathbf{x}_3 : \mathbf{x}_4]\mathbf{c} = 0$, therefore $\mathbf{x}_3$ and $\mathbf{x}_4$ are linearly independent of each other.

Similarly, any combination of two vectors, with one from the set $\{\mathbf{x}_3, \mathbf{x}_4\}$ and another from the set $\{\mathbf{x}_1, \mathbf{x}_2\}$ of Example A.1, is linearly independent. However, any set of three vectors out of $\mathbf{x}_i$ $(i = 1, \ldots, 4)$ is linearly dependent.

Examples A.1 and A.2 can be interpreted geometrically from Fig. A.1, which can be interpreted to have the following three points.

1. Because $\mathbf{x}_1$ and $\mathbf{x}_2$ vectors are parallel in Fig. A.1, or because the angle between them is 180° (or 0°), $\mathbf{x}_1$ and $\mathbf{x}_2$ are linearly dependent, or $\mathbf{x}_1$ differs from $\mathbf{x}_2$ only by a scalar factor. Because the angles between all other vector pairs in Fig. A.1 are not equal to 0° or 180°, all other pairs of vectors are linearly independent.
2. From analytical geometry, the angle θ between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ satisfies the relation

$$\mathbf{x}_i'\mathbf{x}_j = \mathbf{x}_j'\mathbf{x}_i = \|\mathbf{x}_i\|\|\mathbf{x}_j\| \cos\theta, \tag{A.3}$$

where the vector norm $\|\mathbf{x}\|$ is defined in Definition 2.3. For example,

$$\begin{aligned} \mathbf{x}_1'\mathbf{x}_2 &= [\,1 \quad -1\,][-2 \quad 2\,]' = -4 \\ &= \|[\,1 \quad -1\,]\|\|[-2 \quad 2\,]\| \cos 180^\circ = (\sqrt{2})(2\sqrt{2})(-1) \\ \mathbf{x}_1'\mathbf{x}_3 &= [\,1 \quad -1\,][\,1 \quad 0\,]' = 1 \\ &= \|[\,1 \quad -1\,]\|\|[\,1 \quad 0\,]\| \cos\ 45^\circ = (\sqrt{2})(1)(1/\sqrt{2}) \end{aligned}$$

and

$$\begin{aligned} \mathbf{x}_1'\mathbf{x}_4 &= [\,1 \quad -1\,][\,1 \quad 1\,]' = 0 \\ &= \|\mathbf{x}_1\|\|\mathbf{x}_4\| \cos 90^\circ = \|\mathbf{x}_1\|\|\mathbf{x}_4\|(0) \end{aligned}$$

We define two vectors as "orthogonal" if the angle between them is ±90°. For example, $\{\mathbf{x}_1, \mathbf{x}_4\}$ and $\{\mathbf{x}_2, \mathbf{x}_4\}$ are orthogonal pairs, while other vector pairs of Fig. A.1 are not. If cos 0° and cos 180° have the largest magnitude (1) among cosine functions, $\cos(\pm 90^\circ) = 0$ has the smallest. Hence orthogonal vectors are considered "most linearly independent."

We also define $\|\mathbf{x}_i\| \cos\theta$ as the "projection" of $\mathbf{x}_i$ on $\mathbf{x}_j$, if θ

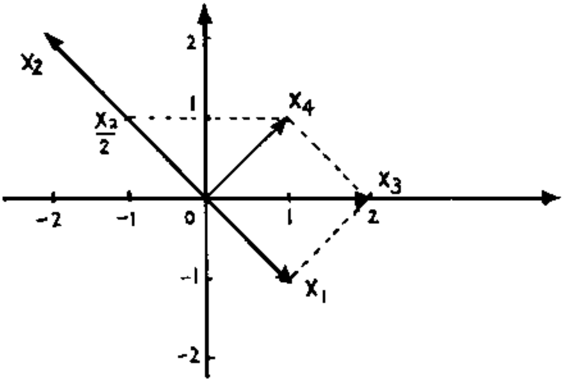


Figure A.1 Four two-dimensional vectors.

is the angle between $\mathbf{x}_i$ and $\mathbf{x}_j$. Obviously, a projection of $\mathbf{x}_i$ is always less than or equal to $\|\mathbf{x}_i\|$ and is equal to 0 if $\theta = \pm 90°$.

3. Any two-dimensional vector is a linear combination of any two linearly independent vectors on the same plane. For example, $\mathbf{x}_3 = \mathbf{x}_1 + \mathbf{x}_4$ and $\mathbf{x}_4 = (1/2)\mathbf{x}_2 + \mathbf{x}_3$. These two relations are shown in Fig. A.1 by the dotted lines. Therefore, the vectors in any set of three two-dimensional vectors are linearly dependent of each other. For example, if $\mathbf{x} = [\mathbf{y} \colon \mathbf{z}]\mathbf{c}$, then $[\mathbf{x} \colon \mathbf{y} \colon \mathbf{z}][1 \colon -\mathbf{c}']' = 0$.

   If two vectors $(\mathbf{y}, \mathbf{z})$ are orthogonal to each other, and if $[\mathbf{y} \colon \mathbf{z}]\mathbf{c}$ equals a third vector $\mathbf{x}$, then the two coefficients of $\mathbf{c}$ equal the projections of $\mathbf{x}$ on $\mathbf{y}$ and $\mathbf{z}$ respectively, after dividing these two projections by their respective $\|\mathbf{y}\|$ and $\|\mathbf{z}\|$. For example, for $\mathbf{x}_3$ of Fig. A.1, the linear combination coefficients (1 and 1) of the orthogonal vectors $\mathbf{x}_1$ and $\mathbf{x}_4$ equal the projections ($\sqrt{2}$ and $\sqrt{2}$) of $\mathbf{x}_3$ on $\mathbf{x}_1$ and $\mathbf{x}_4$, divided by the norms ($\sqrt{2}$, $\sqrt{2}$) of $\mathbf{x}_1$ and $\mathbf{x}_4$.

Definition A.2

A linear space $\mathbf{S}$ can be formed by a set of vectors such that any vector within this set (defined as $\in \mathbf{S}$) can be represented as a linear combination of some other vectors $X = [\mathbf{x}_i \colon \ldots \colon \mathbf{x}_n]$ within this set (defined as the span of X). The largest number of linearly independent vectors needed to represent the vectors in this space is defined as the dimension $\dim(\mathbf{S})$ of that space.

For example, vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ of Example A.1 can span only a straight line space which is parallel to $\mathbf{x}_1$ and $\mathbf{x}_2$. Any of these parallel vectors is a linear combination of another parallel vector only. Hence the dimension of this straight line space is 1.

In Examples A.1 and A.2, each of the vector pair $\{\mathbf{x}_1, \mathbf{x}_3\}, \{\mathbf{x}_1, \mathbf{x}_4\}$, and $\{\mathbf{x}_3, \mathbf{x}_4\}$ can span a plane space, because any vector on this plane can be represented as a linear combination of one of these three vector pairs. In fact, because any one vector in this plane is a linear combination of two linearly independent vectors on this plane, the dimension of this plane space is 2.

Example A.3

The above result can be extended to higher dimensional vectors. Let a set of three-dimensional vectors be

$$[\mathbf{y}_1 \vdots \mathbf{y}_2 \vdots \mathbf{y}_3 \vdots \mathbf{y}_4 \vdots \mathbf{y}_5 \vdots \mathbf{y}_6 \vdots \mathbf{y}_7] = \begin{bmatrix} 2 & 1 & 1 & 0 & 0 & -1 & -2 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 2 & 1 & 0 \end{bmatrix}$$

which are plotted in Fig. A.2.

From Fig. A.2, vectors $\mathbf{y}_1$ and $\mathbf{y}_2$ span a horizontal two-dimensional plane space. Any three-dimensional vector with form $[x\ x\ 0]'$ ("x" stands for an arbitrary entry) or with 0 at the third (vertical) direction equals a linear combination of $\mathbf{y}_1$ and $\mathbf{y}_2$, and therefore lies within this horizontal plane space. For example, $\mathbf{y}_7 = [-2\ \ 1\ \ 0]' = [\mathbf{y}_1 \vdots \mathbf{y}_2][-3/2\ \ 1]'$ belongs to this space. However, all other vectors $\mathbf{y}_3$ to $\mathbf{y}_6$ which stretch on the vertical

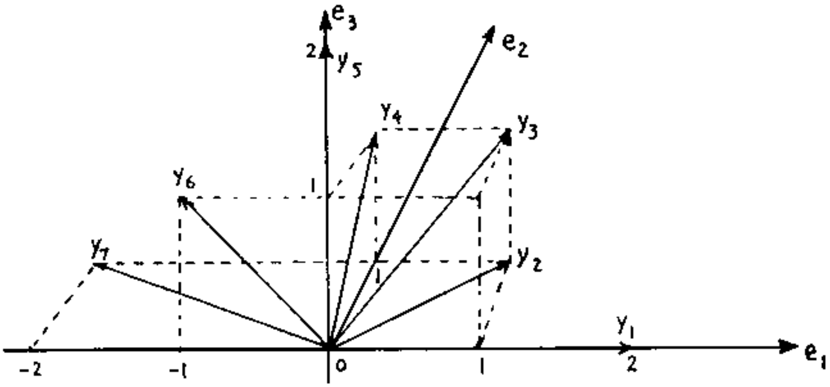


Figure A.2 Seven three-dimensional vectors.

direction are linearly independent of the vectors $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_7\}$ of this horizontal plane space, and hence do not belong to this horizontal plane space.

Although $\mathbf{y}_3$ to $\mathbf{y}_6$ are linearly independent of the vectors $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_7\}$ on the horizontal plane space, only $\mathbf{y}_5$ $(= [0\ 0\ x])$ is orthogonal to all vectors of this horizontal space (also called orthogonal to that space). Finally, any one of the vectors $\mathbf{y}_3$ to $\mathbf{y}_6$, together with two linearly independent vectors of this horizontal plane space, form a three-dimensional cubic space.

Similarly, vectors $\mathbf{y}_1$ and $\mathbf{y}_6$ span a two-dimensional plane space which is parallel to this page flat. Any three-dimensional vector with form $[x\ 0\ x]'$ or with 0 at the second (depth) direction equals a linear combination of $\mathbf{y}_1$ and $\mathbf{y}_6$. For example, $\mathbf{y}_5 = [0\ 0\ 2]' = [\mathbf{y}_1 : \mathbf{y}_6][1\ 2]'$ belongs to this space. However, all other vectors of Fig. A.2 have nonzero projection on the depth direction. Therefore these vectors are linearly independent of the vectors $\{\mathbf{y}_1, \mathbf{y}_5, \mathbf{y}_6\}$ and do not belong to this space. Among these vectors, none is orthogonal to this two-dimensional space because none has the form $[0\ x\ 0]'$, even though within each pair of $\{\mathbf{y}_4, \mathbf{y}_1\}$, $\{\mathbf{y}_2, \mathbf{y}_5\}$, and $\{\mathbf{y}_3, \mathbf{y}_6\}$, the two vectors are orthogonal to each other.

In the literature, there is a more rigorous definition than Definition A.2 for the linear space $\mathbf{S}$ [Gan, 1959]. For example, if we generalize the vectors of a linear space $\mathbf{S}$ as "elements" of that space, then $\mathbf{S}$ must also have "0" and "1" elements [Gan, 1959].

Example A.4

We define the space formed by all n-dimensional vectors $\mathbf{b}$ satisfying the equation $\mathbf{b} = A\mathbf{x}$ (matrix A is given and $\mathbf{x}$ is arbitrary) as $\mathbf{R}(A)$, or as the "range space of A." We also define the number of linearly independent columns/rows of A as the "column rank/row rank" of A. It is clear that the necessary and sufficient condition for $\dim[\mathbf{R}(A)] = n$ [or for $\mathbf{R}(A)$ to include any possible nonzero $\mathbf{b}$] is that the column rank of A equals n.

If the column/row rank of a matrix equals the number of columns/rows of that matrix, then we call this matrix "full-column rank"/"full-row rank."

We also define the space formed by all vectors $\mathbf{x}$ satisfying $A\mathbf{x} = 0$ as $\mathbf{N}(A)$ or the "null space of A." It is clear that if matrix A is full-column rank, then the only vector in $\mathbf{N}(A)$ is $\mathbf{x} = 0$.

However, the set of all vectors $\mathbf{x}$ satisfying $A\mathbf{x} = \mathbf{b}$ ($\mathbf{b} \neq 0$ is given) cannot form a linear space, because this set lacks a "0" element (or $\mathbf{0}$ vector) such that $A\mathbf{0} = \mathbf{b} \neq 0$.

A.1.2 Basis, Linear Transformation, and Orthogonal Linear Transformation

Definition A.3

If any vector $\mathbf{x}$ of a linear space $\mathbf{S}$ is a linear combination of a set of linearly independent vectors of $\mathbf{S}$, then this set of linear independent vectors is defined as a “basis” of $\mathbf{S}$. The linear combination coefficient is defined as the “representation” of $\mathbf{x}$ with respect to this set of basis vectors.

Because any set of n linearly independent n-dimensional vectors can span an n-dimensional linear space $\mathbf{S}$, by Definition A.3 any of these sets can be considered as a basis of $\mathbf{S}$.

Definition A.4

An n-dimensional linear space can have many different sets of basis vectors. The operation which transforms the representation of a vector from one basis to another basis is called a “linear transformation.”

For example, the simplest and most commonly used basis is a set of orthogonal unit coordinate vectors

$$I \underset{=}{\triangle} [\mathbf{e}_1 : \ldots : \mathbf{e}_n] \underset{=}{\triangle} \begin{bmatrix} 1 & 0 & \ldots & & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \\ & & & & 0 \\ 0 & \ldots & & 0 & 1 \end{bmatrix}$$

Because any n-dimensional vector $\mathbf{b} = [b_1, \ldots, b_n]'$ is a linear combination of the vectors $[\mathbf{e}_1 : \ldots : \mathbf{e}_n]$ such that

$$\mathbf{b} = I\mathbf{b} \tag{A.4}$$

and because the representation of $\mathbf{b}$ on I is $\mathbf{b}$ itself, I is a basis and is called an “identity matrix.”

For another example, if we let the vectors of $A = [\mathbf{a}_1 : \ldots : \mathbf{a}_n]$ be the basis for a vector $\mathbf{b}$, then $A\mathbf{x} = \mathbf{b}$ implies that $\mathbf{x} = A^{-1}\mathbf{b}$ is the representation of $\mathbf{b}$ on A.

Now let another set of vectors $V = [\mathbf{v}_1 : \ldots : \mathbf{v}_n]$ be the basis for the same $\mathbf{b}$. Then

$$V\overline{\mathbf{x}} = \mathbf{b} = A\mathbf{x} \tag{A.5a}$$

implies

$$\overline{\mathbf{x}} = V^{-1}\mathbf{b} = V^{-1}A\mathbf{x} \tag{A.5b}$$

is the representation of $\mathbf{b}$ on V.

Definition A.5

A set of orthogonal basis vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_n, (\mathbf{u}_i'\mathbf{u}_j = x\delta_{ij})\}$ is called an "orthogonal basis." The linear transformation which transforms to an orthogonal basis is called "orthogonal linear transformation."

Furthermore, if all vectors of this orthogonal basis are "normalized" $(\|\mathbf{u}_i\| = 1, \forall i)$, then the basis is called "orthonormal" and the corresponding orthogonal linear transformation becomes an "orthonormal linear transformation." A matrix U, which is formed by a set of orthonormal basis vectors, satisfies $U'U = I$ and is called a "unitary matrix."

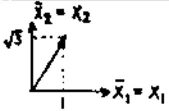
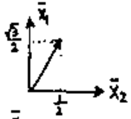
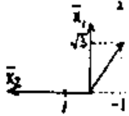
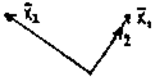
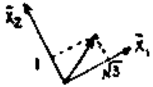
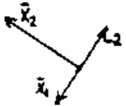
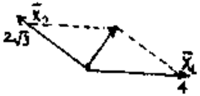
Example A.5

Let a vector $\mathbf{x} = [1\ \sqrt{3}]'$. Table A.1 shows some two-dimensional linear transformation examples in which the orthonormal linear transformation can preserve the norms of any vector $\mathbf{x}$ and its representation $\overline{\mathbf{x}}$ on the new orthonormal basis. This property can be interpreted geometrically from the fourth column of Table A.1, which shows that every element of $\overline{\mathbf{x}}$ equals the projection of $\mathbf{x}$ on the corresponding axis [see interpretation (3) of Fig. A.1]. This property can be proved mathematically that

$$\|\overline{\mathbf{x}}\| = (\overline{\mathbf{x}}'\overline{\mathbf{x}})^{1/2} = (\mathbf{x}'(V^{-1})'(V^{-1})\mathbf{x})^{1/2} = (\mathbf{x}'\mathbf{x})^{1/2} \tag{A.6}$$

if V (or V^{-1}) is a unitary matrix. This property implies that the orthonormal matrix operation is numerically stable [Wilkinson, 1965].

Table A.1 Some Examples of Two-Dimensional Linear Transformation

Basis vectors	Representation of $\mathbf{x}$, $\bar{\mathbf{x}}$	$\lVert\bar{\mathbf{x}}\rVert$	$\mathbf{x}$ and its new basis	Transformation form
$E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ \sqrt{3} \end{bmatrix}$	2		Identity
$O = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}$	$\begin{bmatrix} \sqrt{3}/2 \\ 1/2 \end{bmatrix}$	1	 $\bar{x}_1$ $\frac{\sqrt{3}}{2}$ $\frac{1}{2}$ $\bar{x}_2$	Orthogonal
$U = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} \sqrt{3} \\ -1 \end{bmatrix}$	2		Orthonormal (Givens 90°)
$G_1 = \begin{bmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 0 \end{bmatrix}$	2		Orthonormal (Givens 60°)
$G_2 = \begin{bmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{bmatrix}$	$\begin{bmatrix} \sqrt{3} \\ 1 \end{bmatrix}$	2		Orthonormal (Givens 30°)
$H = \begin{bmatrix} -1/2 & -\sqrt{3}/2 \\ -\sqrt{3}/2 & 1/2 \end{bmatrix}$	$\begin{bmatrix} -2 \\ 0 \end{bmatrix}$	2		Orthonormal (Householder)
$P = \begin{bmatrix} 1 & -\sqrt{3}/2 \\ 0 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 4 \\ 2\sqrt{3} \end{bmatrix}$	$2\sqrt{7}$		Ordinary

A.2 COMPUTATION OF MATRIX DECOMPOSITION

In solving a set of linear equations

$$A\mathbf{x} = \mathbf{b} \tag{A.7a}$$

or in computing the representation $\mathbf{x}$ of $\mathbf{b}$ on the column vectors of A, a nonsingular matrix V^{-1} can be multiplied on the left side of A and $\mathbf{b}$ to make matrix $\overline{A} = V^{-1}A$ in echelon form. Then based on the equation

$$\overline{A}\mathbf{x} = V^{-1}\mathbf{b} \underset{=}{\triangle} \overline{\mathbf{b}} \tag{A.7b}$$

$\mathbf{x}$ can be computed. In other words, the representation $\overline{\mathbf{b}}$ of $\mathbf{b}$ can be computed on the new basis vectors of V such that $\overline{A}$ of (A.7b) is in a decomposed form, and then $\mathbf{x}$ can be computed based on (A.7b).

We will study three different matrices of V^{-1}. All three matrices can be computed from the following unified algorithm.

Algorithm A.1 QR Decomposition [Dongarra et al., 1979]

Let $A = [\mathbf{a}_1 \;\; : \;\; \ldots \;\; : \;\; \mathbf{a}_n]$ be an $n \times n$ dimensional square matrix.

Step 1: Compute $n \times n$ dimensional matrix V_1^{-1} such that

$$V_1^{-1}\mathbf{a}_1 = [x, \; 0 \ldots 0]' \tag{A.8a}$$

Step 2: Let

$$V_1^{-1}A = A_1 = \left[\begin{array}{c:c:c:c:c} x & \multicolumn{4}{c}{\mathbf{a}'_{11}} \\ \hdashline 0 & & & & \\ \vdots & \mathbf{a}_{12} & & \cdots & \mathbf{a}_{1n} \\ 0 & & & & \end{array}\right]$$

Step 3: Compute $(n-1) \times (n-1)$ dimensional matrix $\overline{V}_2^{-1}$ such that

$$\overline{V}_2^{-1}\mathbf{a}_{12} = [x, \; 0 \ldots 0]' \tag{A.8b}$$

Step 4: Let

$$\begin{bmatrix} 1: & \dots & 0 \\ 0: & & \\ \vdots & \overline{V}_2^{-1} & \\ 0: & & \end{bmatrix}(V_1^{-1}A) \triangleq V_2^{-1}A_1$$

$$= \begin{bmatrix} x & : & & & & & \mathbf{a}'_{11} & & \\ .. & .. & .. & .. & \dots & .. & \dots & .. & \dots \\ 0 & : & x & : & & & \mathbf{a}'_{22} & & \\ & : & .. & .. & \dots & .. & \dots & .. & \dots \\ : & : & 0 & : & & : & & : & \\ & : & : & : & \mathbf{a}_{23} & : & \dots & : & \mathbf{a}_{2\mathrm{n}} \\ 0 & : & 0 & : & & : & & : & \end{bmatrix}$$

Continuing in this fashion, at most $n-1$ times we will have

$$(V_{n-1}^{-1} \dots V_2^{-1} V_1^{-1} A \triangleq \ V^{-1}A)$$

$$= \begin{bmatrix} x & \text{-}\mathbf{a}'_{11}\text{-} & & \\ 0 & x & \text{-}\mathbf{a}'_{22}\text{-} & \\ \vdots & & \ddots & \\ & \ddots & & \\ 0 & & \dots & x \end{bmatrix} \tag{A.9}$$

During this basic procedure, if $\mathbf{a}_{i,i+1} = 0$ is encountered $(i = 1, 2, \dots)$, or if

$$V_i^{-1} \dots V_2^{-1} V_1^{-1} A = \begin{bmatrix} x & X & & : & & & & & & & \\ & \ddots & & : & & & & X & & & \\ & & x & : & & & & & & & \\ .. & \dots & \dots & \dots & \dots & \dots & \dots & \dots & & & \\ & & & : & 0 & : & & : & & : & \\ & 0 & & : & : & : & \mathbf{a}_{i,i+2} & : & .. & : & \mathbf{a}_{in} \\ & & & : & 0 & : & & : & & : & \end{bmatrix}$$

then the matrix V_{i+1}^{-1} will be computed based on the next nonzero vector positioned on the right side of $\mathbf{a}_{i,i+1}$ (for example, if $\mathbf{a}_{i,i+2} \neq 0$) such that $\overline{V}_{i+1}^{-1}\mathbf{a}_{i,i+2} = [x,\ 0 \dots 0]'$. The algorithm will then proceed normally.

The above situation can happen more than once. However, as long as this situation happens at least once, the corresponding result of (A.9) will become a so-called upper-echelon form, such as

$$V^{-1}A = \left[\begin{array}{cccccccccccccccccc}
x & R_1 & & : & & & & & & & & & & & & & & \\
 & \ddots & & : & & & & & & & X & & & & & & & \\
0 & & x & : & & & & & & & & & & & & & & \\
\cdots & \cdots & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdots & \cdots & \cdot\cdot & \cdots\cdots & \cdot\cdot & \cdot\cdot & \cdots & \cdots & \cdot\cdot & \cdots & \cdot\cdot \\
0 & \cdots & 0 & : & 0 & : & x & & R_2 & : & & & & & & & & \\
 & & & : & : & : & & \ddots & & : & & & & X & & & & \\
 & & & : & 0 & : & 0 & & x & : & & & & & & & & \\
\cdots & \cdots & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdots & \cdots & \cdot\cdot & \cdots\cdots & \cdot\cdot & \cdot\cdot & \cdots & \cdots & \cdot\cdot & \cdots & \cdot\cdot \\
0 & \cdots & 0 & : & 0 & & 0 & \cdots & 0 & : & 0\ldots0 & : & x & & R_3 & : & & \\
 & & & : & & & & & & : & & : & & \ddots & & : & X & \\
 & & & : & & & & & & : & 0\ldots0 & : & 0 & & x & : & & \\
 & & & : & 0 & & & & & : & \cdots\cdots & : & \cdot\cdot & \cdots & \cdots & \cdot\cdot & \cdots & \cdot\cdot \\
 & & & : & & & & & & : & & : & 0 & \cdots & \cdots & \cdot\cdot & \cdots & 0 \\
 & & & : & & & & & & : & & : & : & & & & & : \\
 & & & : & & & & & & : & & : & 0 & \cdots & \cdots & \cdot\cdot & \cdots & 0 \\
 & & & : & & & & & & : & & : & & & & & & \\
 & \underbrace{\hphantom{000}}_{p} & & & & & & & & & \underbrace{\hphantom{000}}_{q} & & & & & & &
\end{array}\right] \triangleq R \tag{A.10}$$

where "x"'s are nonzero elements.

In the upper-echelon form, the nonzero elements appear only at the upper right-hand side of the upper triangular blocks [such as R_1, R_2, and R_3 in (A.10)]. These upper triangular blocks appear one after another after shifting one or more columns to the right. For example, in (A.10), R_2 follows R_1 after shifting one column to the right, and R_3 follows R_2 after shifting q columns to the right.

If two upper triangular blocks appear one next to the other without column shifting, then the two blocks can be combined as one upper triangular block. If there is no column shifting at all, then the entire matrix is an upper triangular matrix as in (A.9). Hence the upper triangular form is a special case of the upper-echelon form.

The main feature of an upper-echelon-form matrix is that it reveals clearly the linear dependency among its columns. More explicitly, all columns corresponding to the upper triangular blocks are linearly independent of each other, while all other columns are linear combinations of their respective linearly independent columns at their left.

For example in matrix R of (A.10), the $(p+1)$-th column is linearly dependent on the columns corresponding to R_1, while the q columns between R_2 and R_3 are linearly dependent on the columns corresponding to R_1 and R_2.

The above property of an upper-echelon-form matrix enables the solving of Eq. (A.7a). We first let matrix

$$\tilde{A} = [A : \mathbf{b}] \tag{A.11}$$

Then apply Algorithm A.1 to matrix $\tilde{A}$. If after V_r^{-1} is applied,

$$V_r^{-1} \dots V_1^{-1} \tilde{A} = \begin{bmatrix} A_{11} & : A_{12} & : \mathbf{b}_1 \\ 0 & : A_{22} & : 0 \end{bmatrix} \begin{array}{l} \}r \\ \}n-r \end{array} \tag{A.12a}$$

then $\mathbf{b}_1$ is already a linear combination of the columns of A_{11}, and the coefficients of this linear combination form the solution $\mathbf{x}$ of (A.7a). In other words, if $A_{11}\mathbf{x}_1 = \mathbf{b}_1$, then the solution of (A.7a) is $\mathbf{x} = [\mathbf{x}_1' : \mathbf{0}']'$ with $n-r$ 0's in vector $\mathbf{0}$.

In general, we cannot expect the form of (A.12a) for all A and $\mathbf{b}$. Instead, we should expect

$$V_{n-1}^{-1} \dots V_1^{-1} \mathbf{b} = [x \dots x]' \tag{A.12b}$$

For (A.12b) to be represented as a linear combination of the columns of $\overline{A} \underset{=}{\triangle} V_{n-1}^{-1} \dots V_1^{-1} A$, $\overline{A}$ must be in upper triangular form or must have all n columns linearly independent of each other. This is the proof that to have $A\mathbf{x} = \mathbf{b}$ solvable for all $\mathbf{b}$, matrix A must have full-column rank (see Example A.4).

In the basic procedure of Algorithm A.1, only matrix $V_i(V_i^{-1}\mathbf{a}_i = [x,\ 0 \dots 0]')$ can be nonunique. We will introduce three kinds of such matrices in the following. The last two matrices among the three are unitary. We call Algorithm A.1 "QR decomposition" when matrix V is unitary.

For simplicity of presentation, let us express

$$\mathbf{a}_i \underset{=}{\triangle} \mathbf{a} = [a_1, \dots, a_n]'$$

A. Gaussian Elimination with Partial Pivoting

$$
E = E_2E_1 = \begin{bmatrix}
1 & 0 & & \cdots & & & 0 \\
-a_2/a_j & 1 & \ddots & & & & \vdots \\
\vdots & & & \ddots & & & \vdots \\
-a_{j-1}/a_j & & & & \ddots & & \vdots \\
-a_1/a_j & & & & & \ddots & \vdots \\
-a_{j+1}/a_j & & & & & \ddots & \vdots \\
\vdots & & & & & & 0 \\
-a_n/a_j & & & & & & 1
\end{bmatrix}
$$

$$
\begin{array}{c l}
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & & \vdots \\
\vdots & & \ddots & & & & & \\
0 & 0 & 0 & 1 & 0 & 0 & & \\
1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & & \vdots \\
\vdots & & & 0 & & & \ddots & 0 \\
0 & & \cdots & & & & 0 & 1
\end{bmatrix} & \begin{array}{l} \\ \\ \\ \\ \leftarrow j\text{-th row} \\ \\ \\ \\ \end{array} \\
\begin{array}{cccccccc} & & & & \uparrow & & & \end{array} & \\
\text{the } j\text{-th column} &
\end{array}
\tag{A.13}
$$

where $|a_j| = \max_i \{|a_i|\}$ is called the "pivotal element."

Because $E_1\mathbf{a} \triangleq \bar{\mathbf{a}} = [a_j, a_2, \ldots, a_{j-1}, a_1, a_{j+1}, \ldots, a_n]'$, it can be easily verified that $E_2E_1\mathbf{a} = E_2\bar{\mathbf{a}} = [a_j, 0 \ldots 0]'$.

Because of (A.13), all unknown parameters of E_2

$$
|-a_i/a_j| \leqslant 1, \qquad \forall\ i \text{ and } j \tag{A.14}
$$

Therefore the Gaussian elimination with partial pivoting is fairly numerically stable [Wilkinson, 1965].

The order of the computation (multiplications only) of $E\mathbf{x}$ ($\mathbf{x} \neq \mathbf{a}$) is n, excluding the computation of matrix E itself. Hence the order of computation for Algorithm A.1 using Gaussian elimination method is $\Sigma_{i=2 \text{ to } n}\ i^2 \approx n^3/3$.

B. Householder Method [Householder, 1958]

$$H = I - 2\bar{\mathbf{a}}\bar{\mathbf{a}}' \tag{A.15a}$$

where

$$\bar{\mathbf{a}} = \frac{\mathbf{b}}{\|\mathbf{b}\|} \tag{A.15b}$$

and

$$\mathbf{b} = \begin{cases} \mathbf{a} + \|\mathbf{a}\|\mathbf{e}_1, & \text{if } a_1 \geqslant 0 \\ \mathbf{a} - \|\mathbf{a}\|\mathbf{e}_1, & \text{if } a_1 < 0 \end{cases} \tag{A.15c}$$

Because

$$\|\mathbf{b}\| = (\mathbf{b}'\mathbf{b})^{1/2} = (2\|\mathbf{a}\|^2 \pm 2a_1\|\mathbf{a}\|)^{1/2} \tag{A.16}$$

$$\begin{aligned} H\mathbf{a} &= (I - 2\mathbf{b}\mathbf{b}'/\|\mathbf{b}\|^2)\mathbf{a} \\ (\text{A.15}) : &= \mathbf{a} - 2\mathbf{b}(\|\mathbf{a}\|^2 \pm a_1\|\mathbf{a}\|)/\|\mathbf{b}\|^2 \\ (\text{A.16}) : &= \mathbf{a} - 2\mathbf{b}/2 \\ (\text{A.15}) : &= \mathbf{a} - (\mathbf{a} \pm \|\mathbf{a}\|[1, 0 \ldots 0]') \\ &= \mp [\|\mathbf{a}\|, 0 \ldots 0]' \end{aligned} \tag{A.17}$$

In addition, because

$$\begin{aligned} H'H &= (I - 2\bar{\mathbf{a}}\bar{\mathbf{a}}')(I - 2\bar{\mathbf{a}}\bar{\mathbf{a}}') \\ &= I - 4\bar{\mathbf{a}}\bar{\mathbf{a}}' + 4\bar{\mathbf{a}}\bar{\mathbf{a}}'\bar{\mathbf{a}}\bar{\mathbf{a}}' \\ (\text{A.15b}) : &= I - 4\bar{\mathbf{a}}\bar{\mathbf{a}}' + 4\bar{\mathbf{a}}(\mathbf{b}'\mathbf{b}/\|\mathbf{b}\|^2)\bar{\mathbf{a}}' \\ &= I - 4\bar{\mathbf{a}}\bar{\mathbf{a}}' + 4\bar{\mathbf{a}}\bar{\mathbf{a}}' \\ &= I \end{aligned}$$

matrix H is unitary. Hence this computation is numerically stable (see Example A.5).

The actual computation of $H\mathbf{x}$ ($\mathbf{x} \neq \mathbf{a}$) does not need to compute the matrix H itself but can follow the following steps:

Step 1: Compute $2\|\mathbf{b}\|^{-2} = (\mathbf{a}'\mathbf{a} \pm a_1(\mathbf{a}'\mathbf{a})^{1/2})^{-1}$ (computation order: n)
Step 2: Compute scalar $c = 2\|\mathbf{b}\|^{-2}(\mathbf{b}'\mathbf{x})$ (computation order: n)

Step 3: Compute $H\mathbf{x} = \mathbf{x} - c\mathbf{b}$ (computation order: n)

Because the result of Step 1 remains the same for different vectors $\mathbf{x}$, the computation of Step 1 will not be counted. Hence the computation of Algorithm A.1 using Householder method is $\Sigma_{i=2 \text{ to } n}\, 2i^2 \approx 2n^3/3$.

Because computational reliability is more important than computational efficiency, the Householder method is very commonly used in practice and is most commonly used in this book, even though it requires twice as much computation as the Gaussian elimination method.

C. Givens' Rotational Method [Givens, 1958]

$$G = G_1 G_2, \ldots, G_{n-2} G_{n-1} \tag{A.18a}$$

where

$$G_i = \left[\begin{array}{cccccccc} 1 & & \vdots & & \vdots & & & \\ & \ddots & \vdots & & \vdots & & & \\ & & 1\vdots & & \vdots & & & \\ \cdots & \cdots & \cdots\cdot & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & \vdots & R_i & \vdots & & & \\ \cdots & \cdots & \cdots\cdot & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & \vdots & & \vdots & 1 & & \\ & & \vdots & & \vdots & & \ddots & \\ & & \vdots & & \vdots & & & 1 \end{array}\right] \begin{array}{l} \\ \}i-1 \\ \\ \\ \}2 \\ \\ \\ \}n-i-1 \\ \\ \end{array} \tag{A.18b}$$

and

$$R_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \tag{A.18c}$$

Equation (A.18) shows that the component matrices G_i (or R_i) of matrix G are decided by their respective parameter θ_i, $(i = n-1, n-2, \ldots, 1)$. The parameter θ_i is determined by the two-dimensional vector operated by R_i. Let this vector be $\mathbf{b}_i = [x\ y]'$. Then

$$\theta_i = \tan^{-1}(y/x)\ (= 90^\circ \text{ if } x = 0)$$

or

$$\cos\theta_i = x/\|\mathbf{b}_i\| \qquad \text{and} \qquad \sin\theta_i = y/\|\mathbf{b}_i\|$$

It is easy to verify that

$$R_i\mathbf{b}_i = [\|\mathbf{b}_i\|,\ 0\ldots 0]'$$

The geometrical meaning of $R_i\mathbf{b}_i$ can be interpreted as the rotation of the original cartesian coordinates counterclockwise θ_i degrees so that the x-axis now coincides with $\mathbf{b}_i$. This operation is depicted in the Fig. A.3.

The reader can refer to Example A.5 for three numerical examples of Givens' method.

It is easy to verify that according to (A.18a,b,c),

$$G\mathbf{a} = [\|\mathbf{a}\|,\ 0\ldots 0] \tag{A.18d}$$

Because $R_i'R_i = I\ \forall i$, the matrix G of (A.18a,b,c) is a unitary matrix. Therefore like the Householder method, the Givens' rotational method is numerically stable.

It is easy to verify that the order of computation for $G\mathbf{x}$ $(\mathbf{x} \neq \mathbf{a})$ is $4n$, excluding the computation of G itself. Hence the order of computation of Algorithm A.1 is $\Sigma_{i=2 \text{ to } n}\ 4i^2 \approx 4n^3/3$.

Although Givens' method is only half as efficient as Householder's method, it has very simple and explicit geometrical meanings. Therefore it is still commonly used in practice and is used in Algorithm 8.3 of this book.

Finally, after Algorithm A.1 is applied and the echelon-form matrix $V^{-1}A = \overline{A}$ is obtained, we still need to compute $\mathbf{x}$ from $\overline{A}$ and $V^{-1}\mathbf{b} = \overline{\mathbf{b}}$. Eliminating the linearly dependent columns of $\overline{A}$ [see description of the echelon form of (A.10)], we have

$$\overline{A}\mathbf{x} \triangleq \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1r} \\ 0 & a_{22} & \ldots & a_{2r} \\ \vdots & & \ddots & \vdots \\ 0 & \ldots & 0 & a_{rr} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \end{bmatrix} = \overline{\mathbf{b}} \triangleq \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_r \end{bmatrix} \tag{A.19}$$

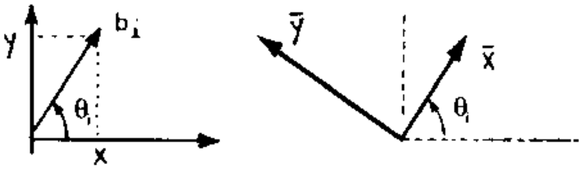


Figure A.3 Geometrical meaning of Givens' rotational method.

where the diagonal elements of the matrix are nonzero. It is obvious that the solution of (A.19) is

$$x_r = \frac{b_r}{a_{rr}} \qquad x_i = \frac{\left(b_i - \sum_{j=i+1}^{r} a_{ij}x_j\right)}{a_{ii}}, \quad i = r-1, r-2, \ldots, 1 \quad \text{(A.20)}$$

The computation of (A.20) is called “back substitution,” whose order of computation at $r = n$ is $n^2/2$. This computation is numerically stable with respect to the problem (A.19) itself [Wilkinson, 1965].

However, because this operation requires consecutive divisions by a_{ii} $(i = r, r-1, \ldots, 1)$, the problem (A.19) can be ill conditioned when these elements have small magnitudes. This understanding conforms with the theoretical result about the condition number $\|\overline{A}\|\|\overline{A}^{-1}\|$ of matrix $\overline{A}$ (2.13). In the next section (A.28)–(A.29), we will show that $\|\overline{A}^{-1}\| = \sigma_r^{-1} \geqslant |\lambda_r|^{-1}$, where σ_r and

$$|\lambda_r| = \min_i\{|a_{ii}|\}$$

are the smallest singular value and the smallest eigenvalue of $\overline{A}$, respectively. Thus small elements a_{ii} imply large and bad condition of matrix $\overline{A}$ as well as problem (A.19).

Comparing the resulting vector $[a_{ii}, 0 \ldots 0]'$ of the three matrix decomposition methods, both orthogonal methods (Householder and Givens) have $a_{ii} = \|\mathbf{a}_i\|_2$ [see (A.17) and (A.18d)], while the Gaussian elimination method has $a_{ii} = \|\mathbf{a}_i\|_\infty$ [see (A.13) and Definition 2.1]. Because $\|\mathbf{a}_i\|_2 \geqslant \|\mathbf{a}_i\|_\infty$, the orthogonal methods not only are computationally more reliable than the Gaussian elimination method, but also make their subsequent computation better conditioned.

A.3 SINGULAR VALUE DECOMPOSITION (SVD)

Matrix singular value decomposition was proposed as early as in 1870 by Betram and Jordan. It became one of the most important mathematical tools in numerical linear algebra and linear control systems theory only in the 1970s [Klema and Laub, 1980], about a hundred years later. This is because SVD is a well-conditioned problem and because of the development of a systematic and numerically stable computational algorithm of SVD [Golub and Reinsch, 1970].

A.3.1 Definition and Existence

Theorem A.1

For any $m \times n$ dimensional matrix A, there exists an $m \times m$ and $n \times n$ dimensional unitary matrix U and V such that

$$A = U \sum V^* = U_1 \Sigma_r V_1^* \tag{A.21}$$

where

$$\sum = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}, \qquad \Sigma_r = \text{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_r\}$$

$$U = [\underset{r}{U_1} \;\;\vdots\;\; \underset{m-r}{U_2}] \qquad \text{and} \qquad V = [\underset{r}{V_1} \;\;\vdots\;\; \underset{n-r}{V_2}]$$

and

$$\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > 0$$

Here σ_i $(i = 1, \ldots, r)$ is the positive square root of the i-th largest eigenvalue of matrix A^*A, and is defined as the i-th nonzero singular value of matrix A. Matrices U and V are the orthonormal right eigenvector matrices of AA^* and A^*A, respectively. In addition, there are $\min\{m, n\} - r \triangleq n - r$ (if $n \leqslant m$) zero singular values $(\sigma_{r+1} = \cdots = \sigma_n = 0)$ of matrix A. Equation (A.21) is defined as the singular value decomposition of matrix A.

Proof

See Stewart [1976].

A.3.2 Properties

Theorem A.2 Minimax Theorem

Let the singular values of an $m \times n$ dimensional matrix A be $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n > 0$. Then

$$\sigma_k = \min_{\dim(S) = n-k+1} \; \max_{\substack{\mathbf{x} \in S \\ \mathbf{x} \neq 0}} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} \qquad k = 1, 2, \ldots, n \tag{A.22}$$

where the linear space $\mathbf{S}$ is spanned by the $n-k+1$ basis vectors $\{\mathbf{v}_k, \mathbf{v}_{k+1}, \ldots, \mathbf{v}_n\}$ which are the last $n-k+1$ vectors of matrix V of (A.21).

Proof

From Definition A.2, let the unitary matrix

$$V \overset{\triangle}{=} [V_1 : V_2] \overset{\triangle}{=} [\mathbf{v}_1 \quad : \quad \ldots \quad : \quad \mathbf{v}_{k-1} | \mathbf{v}_k \quad : \quad \ldots \quad : \quad \mathbf{v}_n]$$

Then the vectors of V_1 will span the "orthogonal complement space" $\overline{\mathbf{S}}$ of $\mathbf{S}$ such that $\overline{\mathbf{S}}'\mathbf{S} = 0$ and $\overline{\mathbf{S}} \cup \mathbf{S} = n$-dimensional space.

Because $\mathbf{x} \in \mathbf{S}$ implies

$$\mathbf{x} = [\mathbf{v}_1 \quad : \quad \ldots \quad : \quad \mathbf{v}_{k-1} | \mathbf{v}_k \quad : \quad \ldots \quad : \quad \mathbf{v}_n] \begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_k \\ \vdots \\ a_n \end{bmatrix} \begin{array}{l} \left.\vphantom{\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}}\right\} k-1 \\ \left.\vphantom{\begin{matrix} a_k \\ \vdots \\ a_n \end{matrix}}\right\} n-k+1 \end{array} \overset{\triangle}{=} V\mathbf{a} \qquad \text{(A.23)}$$

Hence

$$\begin{aligned} \|A\mathbf{x}\|/\|\mathbf{x}\| &= (\mathbf{x}^* A^* Ax/\mathbf{x}^*\mathbf{x})^{1/2} \\ &= (\mathbf{a}^* V^* A^* AV\mathbf{a}/\mathbf{a}^* V^* V\mathbf{a})^{1/2} \\ &= (\mathbf{a}^* \Sigma^2 \mathbf{a}/\mathbf{a}^*\mathbf{a})^{1/2} \\ &= [(a_k^2\sigma_k^2 + a_{k+1}^2\sigma_{k+1}^2 + \cdots + a_n^2\sigma_n^2)/(a_k^2 + a_{k+1}^2 + \cdots + a_n^2)]^{1/2} \\ &\leqslant \sigma_k \end{aligned} \qquad \text{(A.24)}$$

Thus the maximum part of the theorem is proved. On the other hand,

$$\mathbf{x} = [\, \mathbf{v}_1 \;\; \vdots \;\; \ldots \;\; \vdots \;\; \mathbf{v}_k \;\; \vdots \;\; \mathbf{v}_{k+1} \;\; \vdots \;\; \ldots \;\; \vdots \;\; \mathbf{v}_n \,] \begin{bmatrix} a_1 \\ \vdots \\ a_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{A.25}$$

similarly implies that $\|A\mathbf{x}\|/\|\mathbf{x}\| \geqslant \sigma_k$.

The combined (A.24) and (A.25) prove Theorem A.2.

Corollary A.1

$$\|A\| \triangleq \max_{\mathbf{x}\neq 0} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} = \sigma_1 \tag{A.26}$$

Corollary A.2

$$\min_{\mathbf{x}\neq 0} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} = \sigma_n \tag{A.27}$$

Corollary A.3

$$\sigma_1 \geqslant |\lambda_1| \geqslant \cdots \geqslant |\lambda_n| \geqslant \sigma_n \tag{A.28}$$

where λ_i $(i = 1, \ldots, n)$ are the eigenvalues of matrix A (if $m = n$)

Corollary A.4

If A^{-1} exists, then

$$\|A^{-1}\| \triangleq \max_{\mathbf{x}\neq 0} \frac{\|A^{-1}\mathbf{x}\|}{\|\mathbf{x}\|} = \max_{\mathbf{x}\neq 0} \frac{\|V_1 \Sigma_r^{-1} U_1^{*} \mathbf{x}\|}{\|\mathbf{x}\|} = \sigma_n^{-1} \tag{A.29}$$

Corollary A.5

If A^{-1} exists, then

$$\min_{\mathbf{x}\neq 0}\frac{\|A^{-1}\mathbf{x}\|}{\|\mathbf{x}\|}=\sigma_1^{-1} \tag{A.30}$$

Corollary A.6

Let the singular values of two $n\times n$ matrices A and B be $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n$ and $s_1 \geqslant s_2 \geqslant \cdots \geqslant s_{\mathrm{n}}$, respectively, then

$$|\sigma_k - s_k| \leqslant \|A-B\| \overset{\triangle}{=} \|\triangle A\|, \qquad (k=1,\ldots,n) \tag{A.31}$$

Proof

From (A.22),

$$\begin{aligned}\sigma_k = \min_{\substack{\dim(\mathbf{S})=n-k+1\\ \mathbf{x}\in\mathbf{S}.\mathbf{x}\neq 0}} \max\frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} &= \max_{\mathbf{x}\in\mathbf{S},\mathbf{x}\neq 0}\frac{\|(B+\triangle A)\mathbf{x}\|}{\|\mathbf{x}\|}\\ &\leqslant \frac{\|B\mathbf{x}\|_{\mathbf{x}\in\mathbf{S},\mathbf{x}\neq 0}}{\|\mathbf{x}\|}+\frac{\|\triangle A\mathbf{x}\|_{\mathbf{x}\in\mathbf{S},\mathbf{x}\neq 0}}{\|\mathbf{x}\|}\\ &\leqslant s_k + \|\triangle A\|\end{aligned} \tag{A.32}$$

Similarly,

$$s_k \leqslant \sigma_k + \|\triangle A\| \tag{A.33}$$

Hence the theorem.

Corollary A.6 implies that SVD problem is well conditioned, or is insensitive to the original data variation $\triangle A$.

A.3.3 Applications

For simplicity of presentation, we let all matrices of this section be real, and we present all theorems without proof.

A. Solving of a Set of Linear Equations

$$A\mathbf{x} = \mathbf{b}, (\mathbf{b} \neq 0) \tag{A.34}$$

From (A.21):

$$\mathbf{x} = V_1 \Sigma_r^{-1} U_1' \mathbf{b} \tag{A.35}$$

Theorem A.3

If $\mathbf{b}$ is a linear combination of the columns of U_1, then (A.35) is an exact solution of (A.34).

This theorem proves that the columns of U_1 span the range space of A $\mathbf{R}(A)$ (see Example A.4).

Theorem A.4

If $\mathbf{b}$ is not a linear combination of the columns of U_1, then (A.35) is the least-square solution of (A.34). In other words, for all $\triangle \mathbf{x} \neq 0$,

$$\|A\mathbf{x} - \mathbf{b}\| \leqslant \|A(\mathbf{x} + \triangle \mathbf{x}) - \mathbf{b}\| \tag{A.36}$$

if $\mathbf{x}$ is computed from (A.35).

Theorem A.5

If the rank of matrix A is n, then U_1 has n linearly independent columns. Thus the necessary and sufficient condition for (A.34) to have exact solution (A.35) for all $\mathbf{b}$ is that matrix A be full rank.

Theorem A.6

The nonzero solution $\mathbf{x}$ of linear equations

$$A\mathbf{x} = 0 \tag{A.37}$$

is a linear combination of the columns of V_2. In other words, the columns of V_2 span the null space of $A, \mathbf{N}(A)$.

The above result can be generalized to its dual case.

Example A.6

[See Step 2(a), Algorithm 6.1.]

Let the $m \times p$ $(m \leqslant p)$ dimensional matrix DB be full-row rank. Then in its SVD of (A.21), $U_1 = U$ and $U_2 = 0$. Thus based on the duality (or transpose) of Theorem A.6, there is no nonzero solution $\mathbf{c}$ such that $\mathbf{c}DB = 0$.

Based on the duality (or transpose) of Corollary A.2,

$$\min \|\mathbf{c}DB\| = \sigma_m, \quad \text{when } \mathbf{c} = \mathbf{u}'_m = (\text{the } m\text{-th column of } U)'$$

Example A.7

(See Conclusion 6.4 and its proof.)

Let the $n \times p$ $(n > p)$ dimensional matrix B be full-column rank. Then in its SVD of (A.21), U_1 and U_2 have dimensions $n \times p$ and $n \times (n-p)$, respectively. Based on the transpose of Theorem A.6, all rows of $(n-m) \times n$ dimensional matrix T such that $TB = 0$ are linear combinations of the rows of U'_2.

Now because $\mathbf{R}(U_1) \cup \mathbf{R}(U_2) = n$-dimensional space R^n and $U'_1 U_2 = 0$, the rows of any $m \times n$ matrix C such that $[T' : C']'$ is full rank must be linear combinations of the rows of U'_1. Consequently, CB must be full-column rank.

Example A.8

(See Steps 4 of Algorithm 8.2 and Step 3 of Algorithm 8.3.)

Let the columns of an $n \times p$ $(n > p)$ dimensional matrix D be orthonormal. Then in its SVD of (A.21), $U = D$ and $\Sigma_r = V = I_p$. Thus the least-square solution (A.35) of $D\mathbf{c} = \mathbf{b}$ is

$$\mathbf{c} = D'\mathbf{b}$$

Theorem A.7

Let us define A^+ as the pseudo-inverse of matrix A such that $A^+AA^+ = A^+$, $AA^+A = A$, $(AA^+)' = AA^+$, and $(A^+A)' = A^+A$. Then

$$A^+ = V_1 \Sigma_r^{-1} U'_1$$

Thus from Theorems A.3 and A.4, $\mathbf{x} = A^{+}\mathbf{b}$ is the least-square solution of $A\mathbf{x} = \mathbf{b}$.

B. Rank Determination

From Theorems A.3 to A.6, the rank r of an $n \times n$ dimensional matrix A determines whether the Eqs. (A.34) and (A.37) are solvable. If $r = n$, then (A.34) is solvable for all $\mathbf{b} \neq 0$ while (A.37) is unsolvable. If $r < n$, then (A.34) may not be solvable while (A.37) has $n - r$ linearly independent solutions $\mathbf{x}$.

There are several numerical methods for rank determination. For example, Algorithm A.1 can be used to determine the number $(= r)$ of linearly independent columns/rows of a matrix. The rank of a square matrix also equals the number of nonzero eigenvalues of that matrix. However, both numbers are very sensitive to the variation and uncertainty of matrix $A, \Delta A$. Therefore these two methods are not very reliable in rank determination.

On the other hand, the rank of matrix A also equals the number of nonzero singular values of A, and the singular values are insensitive to ΔA. Therefore, this is, so far, the most reliable method of rank determination.

Theorem A.8

If the singular values computed from a given matrix $A + \Delta A$ are $s_1 \geqslant s_2 \geqslant \cdots \geqslant s_n > 0$ $(r = n)$, then the necessary condition for the rank of the original matrix A to be less than n (or σ_n of $A = 0$) is $\|\Delta A\| \geqslant s_n$, and the necessary condition for the rank of A to be less than r (or σ_r of $A = 0$) is $\|\Delta A\| \geqslant s_r$ $(r = 1, \ldots, n)$.

Proof

Let σ_r be zero in Corollary A.6 for $r = 1, \ldots, n$, respectively.

Theorem A.8 implies that the determination of rank $= r$ (or r nonzero singular values) has an accuracy margin which is equivalent of $\|\Delta A\| < \sigma_r$.

In solving the set of linear equations (A.34), the higher the determined r, the more accurate the least-square solution (A.35), and the greater the norm of the corresponding solution because of the greater corresponding σ_r^{-1} (see the end of Sec. A.2). This tradeoff of accuracy and solution magnitude is studied in depth in Lawson and Hanson [1974] and Golub et al. [1976a].

From this perspective, not only the absolute magnitude of the singular values, but also the relative magnitude among the singular values should be considered in rank determination. For example, the r is determined so that there is a greater gap between singular values s_r and s_{r+1} than other singular value gaps.

This tradeoff between accuracy and solution magnitude (or the condition of subsequent computation) also surfaced in control systems problems. For example, such a tradeoff is involved between the condition of Eq. (4.1) and the amount of system order (or system information), as discussed at the end of Sec. 5.2. Such a tradeoff also appears at the Hankow matrix-based model reduction problem [Kung and Lin, 1981] and minimal order realization problem [Tsui, 1983b].

Finally, although singular values are most reliable in revealing the *total* number of linearly independent columns/rows of a matrix, they cannot reveal *which* columns/rows of that matrix are linearly independent of each other. On the other hand, *each* system matrix column or row corresponds to a certain state, a certain input, or a certain output. Hence linear dependency of *each* system matrix column/row is essential in many control problems such as controllability/observability index computation or analytical eigenvector assignment. Because the orthogonal QR matrix decomposition operation (Algorithm A.1) can reveal such linear dependency, and because this method is still quite reliable in computation [DeJong, 1975; Golub et al., 1976a; Tsui, 1983b], it is most widely used in this book.

Appendix B

Design Projects and Problems

There are eight design projects listed with partial answers, in this appendix. Its purpose is twofold. First, these design projects show the usefulness of the theoretical design methods of this book. Second, these design projects are the synthesized and practical exercises of the theoretical design methods of this book.

Because of the limitations on the scope of this book and of control theory itself, only the mathematical models and mathematical design requirements, and not the physical meanings of each project, are described in this appendix. Readers are referred to the original papers for the detailed physical meanings of each project, because such understanding of the actual physical project is essential to any good design.

System 1 Airplane system [Choi and Sirisena, 1974]

$$A = \begin{bmatrix} -0.037 & 0.0123 & 0.00055 & -1 \\ 0 & 0 & 1 & 0 \\ -6.37 & 0 & -0.23 & 0.0618 \\ 1.25 & 0 & 0.016 & -0.0457 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.00084 & 0.000236 \\ 0 & 0 \\ 0.08 & 0.804 \\ -0.0862 & -0.0665 \end{bmatrix}$$

and

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(a) Using Algorithms 5.3 to 6.1, design the dynamic part of the dynamic output feedback compensator of this system, with $F = -2$.

(b) Using Algorithm 9.1, design the LQ optimal state feedback control K for $Q = I, R = I$. Compute the eigenvalues of the corresponding feedback system dynamic matrix $A - BK$.

(c) Using the result K of part (b), design the output part of the dynamic output $\overline{K} \triangleq [K_Z : K_y]$ of the feedback compensator of part (a) such that $K = \overline{K}[T' : C']' \triangleq \overline{KC}$ is best satisfied.

(d) Using Algorithm 8.1 (dual version), design K_y such that the matrix $A - BK_yC$ has the same eigenvalues of part (b).

(e) Using Algorithm 9.2, design the LQ static output feedback control $K_yC\mathbf{x}(t)$ for $Q = I, R = I$. The answer is:

$$K_y = \begin{bmatrix} -0.397 & -1.591 & -7.847 \\ 1.255 & 3.476 & 4.98 \end{bmatrix}$$

(f) Repeat part (e) for the generalized state feedback control $\overline{KC}\mathbf{x}(t)$.

(g) Compare the control systems of part (c) to part (f) in terms of poles, eigenvector matrix condition number, feedback gain, and zero-input response.

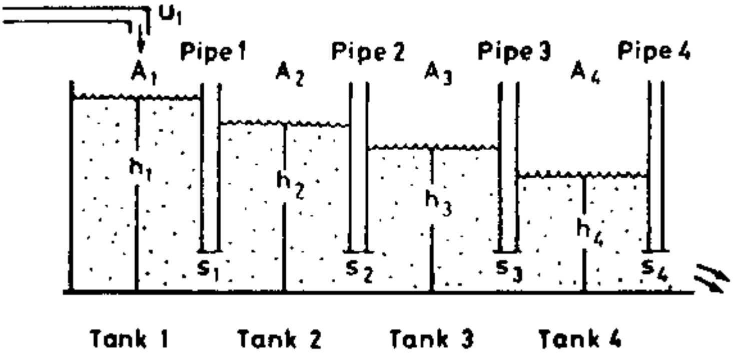


Figure B.1 A four-tank system.

(h) Design a complete failure detection/isolation/accommodation system, with poles equal −1 and −2, and with any result selected from part (c) to part (f) as a normal (failure-free) compensator.

System 2 Four-Water-Tank System [Ge and Fang, 1988]

Figure B.1 shows a four-tank system. On the condition that $A_i = 500\ \text{cm}^2, s_i = 2.54469\ \text{cm}^2\ (i = 1, \ldots, 4)$, and $u(t) = 1\ \text{cm}^3/\text{sec}$, the state space model with the water levels h_i (cm) $(i = 1, \ldots, 4)$ as the four-system states is

$$A = 21.886^{-1}\begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix} \qquad B = \begin{bmatrix} 0.002 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(a) Using Algorithms 5.3 to 6.1 (especially Step 2(b) of Algorithm 6.1), design the dynamic part of a dynamic output feedback compensator of this system, with $F = -7$.

(b) Using Algorithm 10.1, design the failure detection and isolation system with $q = 2$ and with robust failure detector pole (or double pole) equal -7. The partial answer is:

$$T_1 = [0 \quad 0 \quad 0 \quad -130.613]$$

$$T_3 = \begin{bmatrix} 0 & 16.702 & -110.655 & 0 \\ 0 & -27.144 & 135.721 & 0 \end{bmatrix}$$

$$T_5 = [84.582 \quad 0 \quad -84.582 \quad 0]$$

$$T_6 = \begin{bmatrix} 116.517 & -18.713 & 0 & 0 \\ -114.759 & 22.952 & 0 & 0 \end{bmatrix}$$

(c) Using the duality of Algorithm 5.3, design a unique state feedback gain K to place the eigenvalues $-2.778 \pm j14.19$ and $-5.222 \pm j4.533$ in matrix $A - BK$.

(d) Compare the parameter T_0 of part (a) and T_i $(i = 1, 3, 5, 6)$ of part (b) in generating the normal-state feedback K of part (c): Let $K = K_i C_i$, where $C_i \triangleq [T_i' : C']'$ $(i = 0, 1, 3, 5, 6)$, and compare the accuracy of $K_i C_i$ and the magnitude of gain K_i.

(e) Repeat Examples 10.5 and 10.6 for the design of failure accommodation control and of threshold treatment of model uncertainty and measurement noise.

System 3 A Corvette 5.7 L, Multi-port, Fuel-Injected Engine [Min, 1990]

At the operating point that manifold pressure = 14.4 In-Hg, throttle position at 17.9% of maximum, engine speed = 1730 RPM, and load torque = 56.3 ft-lb., the linearized state space model is

$$A = \begin{bmatrix} 0.779 & 0.0632 & -0.149 & -0.635 & -0.211 \\ 1 & 0 & 0 & 0 & 0 \\ 0.271 & -0.253 & 0.999 & 0 & 0.845 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.579 & 0.22598 \\ 0 & 0 \\ 0 & -0.9054 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The five system states are: change in manifold pressure, change in manifold pressure (last rotation), change in engine RPM, change in throttle position, and change in external load, respectively. The two inputs are the next rotation throttle angle change and the change of external load during the next rotation, respectively.

(a) Using Algorithms 5.3 to 6.1, design the dynamic part of an output feedback compensator for this system, with poles $-1 \pm j$.

(b) Determine the rank of matrix $[T' : C']'$. It should be $5 = n$.

(c) Design the failure detection, isolation, and accommodation system, with $q = 2$, and poles $= -2, -4$, and -6.

System 4 Booster Rockets Ascending Through Earth's Atmosphere [Enns, 1990]

$$A = \begin{bmatrix} -0.0878 & 1 & 0 & 0 \\ 1.09 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -37.6 & -0.123 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 4.2 \times 10^{-10} \\ 0 & 0 & 1.27 \times 10^{-8} \\ 0 & 0 & 0 \\ 1 & 0 & -1.2 \times 10^{-6} \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & -0.00606 \\ 0 & 0 & -37.6 & -0.123 \\ 0 & 0 & 0 & -0.00606 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1.2 \times 10^{-6} \\ 0 & 0 & 0 \end{bmatrix}$$

The four system states are angle of attack (rad.), pitch rate q (rad/sec), lowest frequency elastic model deflection η, and $\dot{\eta}$, respectively. The three inputs are the error of elastic model poles v_{POLE}, error of elastic model zeros v_{ZERO}, and the thrust vectoring control u_{TVC}(lb.), respectively. The three outputs are the gyro output measurement y_{GYRO}(rad/sec), $\eta - v_{\text{POLE}}$, and $y_{\text{GYRO}} - q - v_{\text{ZERO}}$, respectively.

From a control theory point of view, a difficulty involved with this problem is that the third column of B is too small, while its corresponding input is the only real control input u_{TVC} (the other two inputs are artificially added to account for the errors associated with the elasticity model). In addition, adjustment has to be made to consider the nonzero D matrix, which is assumed to be zero in this book. Nonetheless, without matrix D and by eliminating the second column of matrix B, the example becomes similar to that of System 1.

System 5 Bank-to-Turn Missile [Wise, 1990]

At the flight conditions of 16° of angle of attack, Mach 0.8 (velocity of 886.78 ft/sec), and attitude of 4000 ft, the linearized missile rigid body airframe state space model is

$$A = \begin{bmatrix} -1.3046 & 0 & -0.2142 & 0 \\ 47.7109 & 0 & -104.8346 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -12{,}769 & -135.6 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 12{,}769 \end{bmatrix}$$

and

$$C = \begin{bmatrix} -1156.893 & 0 & 189.948 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The four system states are angle of attack, pitch rate, fin deflection, and fin rate, respectively. The control input is fin deflection command (rad.), and the two outputs are normal acceleration (ft/s^2) and pitch rate (rad/s), respectively.

(a) Using Algorithms 5.3 to 6.1, design the dynamic part of the dynamic output feedback compensator of this system. Because $CB = 0$, we let the compensator order $r = 1$ and dynamic matrix $F = -10$.

(b) Using the duality of Algorithm 5.3, design state feedback gain K which can place each of the following four sets of eigenvalues in matrix $A - BK$ [Wilson et al., 1992]: $\{-5.12,\ -14.54,\ -24.03 \pm j18.48\}$, $\{-10 \pm j10,\ -24 \pm j18\}$, $\{-9.676 \pm j8.175,\ -23.91 \pm j17.65\}$, and $\{-4.7 \pm j2.416,\ 23.96 \pm j17.65\}$.

(c) Design the respective output part $\overline{K}$ of the dynamic output feedback compensator of part (a), for the four sets of eigenvalues of part (b).

(d) Compare the controls of parts (b) and (c), for each of the four sets of part (b). The comparison can be made in the practical aspects such as the control gain (K vs. $\overline{K}$) and the zero-input response.

(e) Design a complete failure detection, isolation, and accommodation system, with $q = 1$ and poles $= -14, \; -10 \pm j10$. The normal feedback compensator can be chosen from any of the four compensators of parts (a) and (c).

System 6 Extended Medium-Range Air-to-Air Missile [Wilson et al., 1992]

At the flight condition of 10° of angle of attack, Mach 2.5 (velocity of 2420 ft/s), and dynamic pressure of $1720\,\text{lb/ft}^2$, the normal roll-yaw missile airframe model is

$$A = \begin{bmatrix} -0.501 & -0.985 & 0.174 & 0 \\ 16.83 & -0.575 & 0.0123 & 0 \\ -3227 & 0.321 & -2.1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0.109 & 0.007 \\ -132.8 & 27.19 \\ -1620 & -1240 \\ 0 & 0 \end{bmatrix}$$

The four system states are sideslip, yaw rate, roll rate, and roll angle, respectively. The two control inputs are rudder position and aileron position, respectively.

(a) For each of the four sets of feedback system eigenvalues of System 5, use Algorithms 8.2 and 8.3 and the analytic decoupling rules to design the eigenvectors and the corresponding state feedback gains.

(b) Compare each of the four sets of results of part (a) with the following corresponding result of Wilson et al. [1992]:

$$K = \begin{bmatrix} 1.83 & -0.154 & 0.00492 & -0.0778 \\ -2.35 & 0.287 & -0.03555 & 0.0203 \end{bmatrix}$$

$$K = \left[\begin{matrix} 5.6 & -0.275 & -0.00481 & -0.989 \\ -4.71 & 0.359 & -0.00815 & 1.1312 \end{matrix}\right.$$

$$K = \begin{bmatrix} 3.19 & -0.232 & 0.10718 & 0.1777 \\ -1.63 & 0.299 & -0.15998 & -0.4656 \end{bmatrix}$$

$$K = \begin{bmatrix} 1.277 & -0.172 & 0.10453 & 0.1223 \\ 0.925 & 0.2147 & -0.15696 & -0.2743 \end{bmatrix}$$

The comparison can be made in practical aspects such as feedback gain, robust stability (2.23 – 2.25), and zero-input response.

System 7 Chemical Reactor [Munro, 1979]

$$A = \begin{bmatrix} 1.38 & -0.2077 & 6.715 & -5.676 \\ -0.5814 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{bmatrix}$$

(a) Repeat part (a) of System 6, but for a new eigenvalue set: $\{-0.2, -0.5, -5.0566, -8.6659\}$.

(b) Repeat part (b) of System 6, but compare the following two possible results [Kautsky et al., 1985]:

$$K = \begin{bmatrix} 0.23416 & -0.11423 & 0.31574 & -0.26872 \\ 1.1673 & -0.28830 & 0.68632 & -0.24241 \end{bmatrix}$$

$$K = \begin{bmatrix} 0.10277 & -0.63333 & -0.11872 & 0.14632 \\ 0.83615 & 0.52704 & -0.25775 & 0.54269 \end{bmatrix}$$

System 8 Distillation Column [Kle, 1977]

$$A = \begin{bmatrix} -0.1094 & 0.0628 & 0 & 0 & 0 \\ 1.306 & -2.132 & 0.9807 & 0 & 0 \\ 0 & 1.595 & -3.149 & 1.547 & 0 \\ 0 & 0.0355 & 2.632 & -4.257 & 1.855 \\ 0 & 0.00227 & 0 & 0.1636 & -0.1625 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 & 0 \\ 0.0638 & 0 \\ 0.0838 & -0.1396 \\ 0.1004 & -0.206 \\ 0.0063 & -0.0128 \end{bmatrix}$$

(a) Repeat part (a) of System 6, but for a new set of eigenvalues $\{-0.2, -0.5, -1, -1 \pm j\}$.

(b) Repeat part (b) of System 6, but compare the following result of Kautsky et al. [1985]:

$$K = \begin{bmatrix} -159.68 & 69.844 & -165.24 & 125.23 & -45.748 \\ -99.348 & 7.9892 & -14.158 & -5.9382 & -1.2542 \end{bmatrix}$$