
AIRCRAFT CONTROL AND SIMULATION

BRIAN L. STEVENS

Georgia Tech Research Institute

FRANK L. LEWIS

The University of Texas at Arlington

2003

— SET 2003

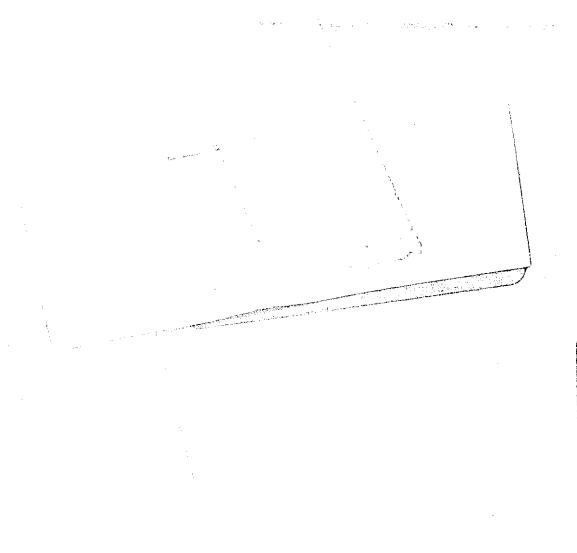
2004



A Wiley-Interscience Publication

JOHN WILEY & SONS, INC.

New York / Chichester / Brisbane / Toronto / Singapore



The cover graphic of the YF-22 was drawn from a photograph provided courtesy of Lockheed Aeronautical Systems Company.

In recognition of the importance of preserving what has been written, it is a policy of John Wiley & Sons, Inc., to have books of enduring value published in the United States printed on acid-free paper, and we exert our best efforts to that end.

Copyright © 1992 by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

Reproduction or translation of any part of this work beyond that permitted by Section 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

Library of Congress Cataloging in Publication Data:

Stevens, Brian L., 1939-

Aircraft control and simulation / Brian L. Stevens, Frank L. Lewis.

p. cm.

Includes bibliographical references.

ISBN 0-471-61397-5

1. Airplanes—Control systems. I. Lewis, Frank L. II. Title.

TL678.S74 1992

629.35—dc20

91-13413

CIP

Printed in the United States of America

10 9 8 7 6 5 4 3 2

AIRCRAFT CONTROL AND SIMULATION

To Richard and Bill
—B.L.S.

To Christopher — who has
brought everything together
—F.L.L.

CONTENTS

LIST OF TABLES	xii
LIST OF EXAMPLES	xiii
PREFACE	xvii
1 EQUATIONS OF MOTION	1
1.1 Introduction / 1	
1.2 Mathematical Preliminaries / 4	
1.3 Dynamic Analysis / 18	
1.4 Kinematic Analysis / 33	
1.5 Assembling the State Equations / 44	
1.6 Summary / 47	
References / 48	
Problems / 49	
2 BUILDING THE AIRCRAFT MODEL	51
2.1 Introduction / 51	
2.2 Basic Aerodynamics / 52	
2.3 Aircraft Forces and Moments / 61	
2.4 The Nonlinear Aircraft Model / 80	
2.5 Linear Models and the Stability Derivatives / 89	
2.6 Summary / 109	
References / 110	
Problems / 111	

3 BASIC ANALYTICAL AND COMPUTATIONAL TOOLS	113
3.1 Introduction / 113	
3.2 Models of Subsystems / 115	
3.3 Aircraft Models for Simulation / 123	
3.4 Steady-State Trimmed Flight / 129	
3.5 Numerical Solution of the State Equations / 139	
3.6 Linearization / 152	
3.7 Computations for Linear Time-Invariant State Equations / 158	
3.8 Feedback Control / 179	
3.9 Summary / 191	
References / 191	
Problems / 193	
4 AIRCRAFT DYNAMICS AND CLASSICAL DESIGN TECHNIQUES	196
4.1 Introduction / 196	
4.2 The Effect of Flight Conditions on the Modes / 204	
4.3 The Flying-Qualities Requirements / 222	
4.4 Stability Augmentation / 236	
4.5 Control Augmentation Systems / 254	
4.6 Autopilots / 285	
4.7 Nonlinear Simulation of Limiters / 325	
4.8 Summary / 332	
References / 333	
Problems / 334	
5 MODERN DESIGN TECHNIQUES	337
5.1 Introduction / 337	
5.2 Assignment of Closed-Loop Dynamics / 342	
5.3 Linear Quadratic Regulator with Output Feedback / 359	
5.4 Tracking a Command / 376	
5.5 Modifying the Performance Index / 391	
5.6 Model-Following Design / 421	
5.7 Linear Quadratic Design with Full State Feedback / 437	
5.8 Summary / 444	
References / 445	
Problems / 447	

6 ROBUSTNESS AND MULTIVARIABLE FREQUENCY-DOMAIN TECHNIQUES	452
6.1 Introduction / 452	
6.2 Multivariable Frequency-Domain Analysis / 454	
6.3 Robust Output-Feedback Design / 479	
6.4 Observers and the Kalman Filter / 483	
6.5 LQG/Loop-Transfer Recovery / 510	
6.6 Summary / 535	
References / 535	
Problems / 537	
7 DIGITAL CONTROL	542
7.1 Introduction / 542	
7.2 Simulation of Digital Controllers / 543	
7.3 Discretization of Continuous Controllers / 545	
7.4 Modified Continuous Design / 558	
7.5 Implementation Considerations / 572	
7.6 Summary / 581	
References / 581	
Problems / 582	
APPENDIX A F-16 MODEL	584
APPENDIX B SOFTWARE	593
INDEX	609

LIST OF TABLES

CHAPTER 1

- 1.3-1 Variables Used in the Equations of Motion / 21

CHAPTER 2

- 2.2-1 Important Wing-Planform Parameters / 59
2.3-1 Force, Moment, and Velocity Definitions / 64
2.3-2 Typical Aerodynamic Component Buildup Equations / 68
2.4-1 The Flat-Earth, Body-Axes 6-DOF Equations / 81
2.5-1 The Force Dimensional Derivatives (Wind Axes) / 94
2.5-2 The Moment Dimensional Derivatives (Wind Axes) / 98
2.5-3 Longitudinal Dimensional Versus Dimensionless
Derivatives / 105
2.5-4 Lateral–Directional Dimensional Versus Dimensionless
Derivatives / 106

CHAPTER 3

- 3.2-1 Network Transfer Functions and State Equations / 116
3.3-1 Aircraft Control-Surface Sign Conventions / 128
3.3-2 F-16 Model Test Case / 128
3.4-1 Trim Data for the Transport Aircraft Model / 135
3.4-2 Trim Data for the F-16 Model / 136
3.4-3 Trimmed Flight Conditions for the F-16 Model / 139
3.7-1 F-16 Model, Elevator-to-Pitch-Rate Transfer Function / 172

CHAPTER 4

- 4.2-1 Accuracy of Short-Period and Phugoid Formulae / 211
- 4.2-2 Effect of Flight-Path Angle on the F-16 Modes / 221
- 4.2-3 Effect of Speed and Altitude on the F-16 Modes / 221
- 4.3-1 Pilot Opinion Rating and Flying-Qualities Level / 223
- 4.3-2 Definitions: Flying-Qualities Specifications / 233
- 4.3-3a Short-Period Damping Ratio Limits / 234
- 4.3-3b Limits on $\omega_{n_{sp}}^2/(n/\alpha)$ / 234
- 4.3-4 Maximum Roll Mode Time Constant / 235
- 4.3-5 Spiral Mode, Minimum Doubling Time / 235
- 4.3-6 Dutch Roll Mode Specifications / 236
- 4.5-1 Transfer Function Zeros Versus Accelerometer Position / 264
- 4.5-2 Trim Conditions for Determining ARI Gain / 275

CHAPTER 5

- 5.3-1 LQR with Output Feedback / 364
- 5.3-2 Optimal Output Feedback Solution Algorithm / 366
- 5.4-1 LQ Tracker with Output Feedback / 385
- 5.5-1 LQ Tracker with Time-Weighted PI / 398
- 5.7-1 LQR with State Feedback / 440

CHAPTER 6

- 6.4-1 The Kalman Filter / 502

CHAPTER 7

- 7.4-1 Padé Approximants to $e^{-s\Delta}$ for Approximation of Computation Delay / 566
- 7.4-2 Approximants to $(1 - e^{-sT})/sT$ for Approximation of Hold Delay / 567
- 7.5-1 Elements of Second-Order Modules / 580
- 7.5-2 Difference Equation Implementation of Second-Order Modules / 580

LIST OF EXAMPLES

CHAPTER 3

- 3.2-1 A Nonlinear Model of a Hydraulic Actuator / 120
- 3.4-1 Longitudinal Trim for the Transport Aircraft Model / 134
- 3.4-2 The F-16 Trimmed Power Curve / 136
- 3.4-3 Trim Conditions for a Coordinated Turn / 137
- 3.5-1 Simulation of a Coordinated Turn / 149
- 3.5-2 Simulated Response to an Elevator Pulse / 150
- 3.5-3 Simulated Response to a Throttle Pulse / 151
- 3.6-1 Linearization of the F-16 Model / 155
- 3.6-2 Comparison of Algebraic and Numerical Linearization / 157
- 3.7-1 Discrete-Time Quaternion Propagation / 159
- 3.7-2 Discrete-Time Inertial Navigation Calculations / 160
- 3.7-3 F-16 Longitudinal Modes / 164
- 3.7-4 F-16 Lateral-Directional Modes / 165
- 3.7-5 F-16 Elevator-to-Pitch-Rate Transfer Function / 172
- 3.7-6 Transport Aircraft Throttle Response / 173
- 3.7-7 F-16 Elevator-to-Pitch-Rate Frequency Response / 177
- 3.8-1 MIMO System with Successive Loop Closures / 187
- 3.8-2 Root-Locus Design Using the State-Space Formulation / 189

CHAPTER 4

- 4.2-1 Lateral Modes of a Business Jet / 218
- 4.2-2 Mode Dependence from the Nonlinear Model / 220

- 4.4-1 The Effects of Pitch-Rate and Alpha Feedback / 239
- 4.4-2 A Pitch SAS with Washed-out Alpha Feedback / 243
- 4.4-3 A Roll Damper/Yaw Damper Design / 248
- 4.5-1 A Pitch-Rate CAS Design / 256
- 4.5-2 Pitch-CAS Nonlinear Simulation / 260
- 4.5-3 A Normal Acceleration CAS Design / 265
- 4.5-4 A Lateral-Directional CAS Design / 276
- 4.6-1 A Simple Pitch-Attitude-Hold Autopilot / 286
- 4.6-2 A Pitch-Attitude Hold with Dynamic Compensation / 288
- 4.6-3 An Altitude-Hold Autopilot Design / 293
- 4.6-4 Longitudinal Control for Automatic Landing / 300
- 4.6-5 Automatic Flare Control / 309
- 4.6-6 A Bank-Angle-Hold Autopilot / 317
- 4.6-7 A Bank-Angle-Steering Control System / 319
- 4.7-1 Simulation of a Controller with Limiters / 326

CHAPTER 5

- 5.2-1 Selecting Eigenvectors for Decoupling / 345
- 5.2-2 Eigenstructure Assignment Using Dynamic Regulator / 353
- 5.2-3 Eigenstructure Design of Longitudinal Pitch Pointing Control / 354
- 5.3-1 LQR Design for F-16 Lateral Regulator / 370
- 5.4-1 Normal Acceleration CAS / 386
- 5.5-1 Constrained Feedback Control for F-16 Lateral Dynamics / 399
- 5.5-2 Time-Dependent Weighting Design of Normal Acceleration CAS / 401
- 5.5-3 Pitch-Rate Control System Using LQ Design / 402
- 5.5-4 Multivariable Wing Leveler / 406
- 5.5-5 Glide-Slope Coupler / 410
- 5.6-1 Automatic Flare Control by Model-Following Design / 429
- 5.7-1 LQR with State Feedback for Systems Obeying Newton's Laws / 442

CHAPTER 6

- 6.2-1 MIMO Bode Magnitude Plots / 461
- 6.2-2 Singular Value Plots for F-16 Lateral Dynamics / 462
- 6.2-3 Precompensator for Balancing and Zero Steady-State Error / 468

- 6.2-4 Model Reduction and Stability Robustness / 472
- 6.3-1 Pitch-Rate Control System Robust to Wind Gusts and Unmodeled Flexible Mode / 480
- 6.4-1 Observer Design for Double Integrator System / 489
- 6.4-2 Kalman Filter Estimation of Angle of Attack in Gust Noise / 502
- 6.5-1 LQG/LTR Design of Aircraft Lateral Control System / 521

CHAPTER 7

- 7.3-1 Discrete PID Controller / 551
- 7.3-2 Digital Pitch Rate Controller via BLT / 553
- 7.4-1 Digital Pitch Rate Controller via Modified Continuous Design / 567
- 7.5-1 Antiwindup Compensation for Digital PI Controller / 576

PREFACE

This book is aimed at a number of different groups of engineers. We hope that it will be useful to practicing aerospace engineers both as a reference book and as an update to their engineering education. Reference material such as equations of motion, the gravity model, differential equation models, controller design algorithms, and so on, are readily accessible and in tabular form wherever appropriate. The design examples can easily be reproduced on a home computer; involvement with these designs will demonstrate many ideas in control theory, computer-aided design techniques, and numerical algorithms.

The book should prove useful for both undergraduate and graduate courses in aerospace engineering. The design examples are quite realistic and offer a great deal of scope to a class of students with access to computers. It has also been our experience that electrical engineering courses in automatic control have a continual need for dynamic models and interesting design examples, and we hope that this material will fulfill some of those needs.

In Chapter 1 we have devoted more than the usual amount of space (for an aircraft book) to the dynamics and kinematics of rigid-body motion, to vector analysis and linear algebra, to coordinate transformations, and to ideas from guidance and navigation. This parallels our experience in the aerospace industry. Very few engineers entering the industry will find themselves designing flight control systems, and those few will take part in the design of only two or three such systems in their careers. Instead, they will find themselves involved in a broad spectrum of projects, where a good grasp of dynamics, coordinate transformations, and related analytical ideas will be invaluable.

Chapter 2 provides the basic aircraft stability and control material, but with a bias toward the controls engineer and toward fully powered irreversible flight control systems. Again this matches the authors' experience in

aircraft and missile control, where the controls designer or analyst will be faced with a large database or complete dynamic model for an existing vehicle. The designer must then deal with it as a "black box," accessible only through the given input and output "terminals." He or she must also work with the group of engineers responsible for this model, speak their language of stability derivatives, and understand the aerodynamic limitations imposed upon them. With respect to the small-perturbation equations and stability derivatives, the engineer with an analytical bias or a pure controls background is often put off by seeing a book with a large number of assumptions and approximations, and lacking a compact mathematical framework. We have attempted to provide compact and theoretically sound derivations of the stability-axes and wind-axes equations, and of the small perturbation equations and the stability derivatives.

We make no apology for devoting space to mathematical modeling, digital simulation, and numerical algorithms in Chapter 3. Large sums of money are spent on mathematical modeling and digital simulation before any hardware is built. Engineers performing these tasks should have an understanding of their tools at least equivalent to that of their predecessors, who designed hardware directly. This chapter also provides the aircraft models that are used in the remainder of the book. We have aimed for greater design realism than in most texts by providing a full nonlinear aircraft model, with look-up-table aerodynamic data. All of our linear models are derived from the nonlinear models by numerical linearization in an appropriate flight condition; this overcomes limitations imposed by the classical stability derivative approach.

There are many excellent mathematical computer packages available for simulation and control system design. However, the algorithms that we give for steady-state trim, numerical linearization, and output-feedback controls design are quite specialized and not readily available. Other software is described because it fits directly into the design framework shown in Fig. 3.1-1, and proved very convenient when the design examples were being developed. Also, we believe that an engineer should be able to construct sound numerical algorithms and write driver programs for the many high-quality scientific subroutines that are readily available. We note that the Laplace transform, which once received so much attention in textbooks, has essentially been pushed out of our toolkit by numerical integration algorithms. Thus one need not distinguish between linear or nonlinear dynamics when a time response is required.

Chapter 4 starts by describing the need for automatic flight controls, the variation of the aircraft dynamics with flight conditions, and the performance criteria for control systems design. The flying-qualities criteria are inadequate for the purposes of control system design, and flight controls design depends heavily on nonlinear simulation and flight tests. For these reasons the design examples in the text cannot be carried through to final designs, nor is there space available to apply the flying-qualities criteria thoroughly to each design.

Nevertheless, we believe that the design examples provide sufficient insight that the approach could be carried through to a final design in the appropriate environment. We learned a great deal from the design examples and the nonlinear simulations that were performed, and if there is merit in the book, a large part of it resides in these design examples.

From the beginning the book is based on a "modern" state-space framework for the mathematical models, and this makes the classical controls design techniques in Chapters 3 and 4 very easy to employ. In Chapter 5 modern control design techniques are introduced, followed by robustness theory in Chapter 6. Here we are very conscious of the degree of acceptance and use of these tools in industry. Time response is all-important to the human operator of a control system, yet the modern control thrust of robustness first and foremost has pushed this fact into the background. Thus one can see many examples of robust design that have unacceptable time responses. Furthermore, since flying qualities can never be made satisfactory in the first design iteration, the ease with which the design can be modified is of great importance. Deficiencies in flying qualities must be correlated with properties of the control laws, and the control law design must be revisited many times during the piloted simulation studies and the flight test program.

In practice, robustness is intimately linked to practical engineering considerations that arise in the design and to the complexity of the gain scheduling that is used with the design. Nonlinear effects are often deliberately added to the control systems based on an understanding of the physics of flight (see, e.g., the lateral/directional CAS design in Chapter 4), and it is dangerous to draw design conclusions from linear small-perturbation models. For these reasons we use modern techniques as an extension of the classical techniques. We allow prespecified compensators and filters (e.g., PI controllers and washout filters) to be incorporated into a modern design, we use output feedback where appropriate, and we use modern singular value analysis to assess robustness after the design to meet flying-qualities specifications has been performed. We also show how modern LQ theory can be used to shape control system time response directly, a problem that has always been difficult with classical design techniques.

In the final chapter we introduce digital control. New commercial and military aircraft all have digital flight control systems because of the flexibility offered, and many old designs are being upgraded from analog to digital. Here it is worth noting that the development of new real-time computer code for a flight control computer involves a massive investment of time and money and a large support organization thereafter. Consequently, a design procedure that results in a compensator of arbitrary structure, where a simple phase-lead or PI compensator was used previously, can cause severe problems in developing error-free and transient-free code for real-time computer control.

We realize that several important topics have been omitted from the book. We have mentioned the effects of structural flexibility only in passing, and

disturbance inputs to the aircraft dynamics have not been treated in any detail. Time and size limitations were mainly responsible for these omissions.

The first author wishes to acknowledge many helpful discussions with former Lockheed colleagues, in particular with Dr. P. Vesty, Dr. E. Y. Shapiro, and Mr. H. R. Rooney, and also with Mr. L. Nguyen of NASA-Langley and Dr. R. L. Roach of the Georgia Institute of Technology. He also wishes to acknowledge the past support of Dr. H. E. Plumblee, who provided funding for control systems research in the Lockheed-Georgia Advanced Research Organization. Special thanks are due to Ms. Betty Darden and Deane Stevens for many hours of proofreading. All of the word-processing, scientific computations, graphics, and illustrations were done on the authors' personal computers. The graphs were done with the excellent software developed by M. Baden at Georgia Tech Research Institute, and the forty-plus computer-graphic illustrations for the first four chapters were drawn by Bill Stevens during his high school vacation.

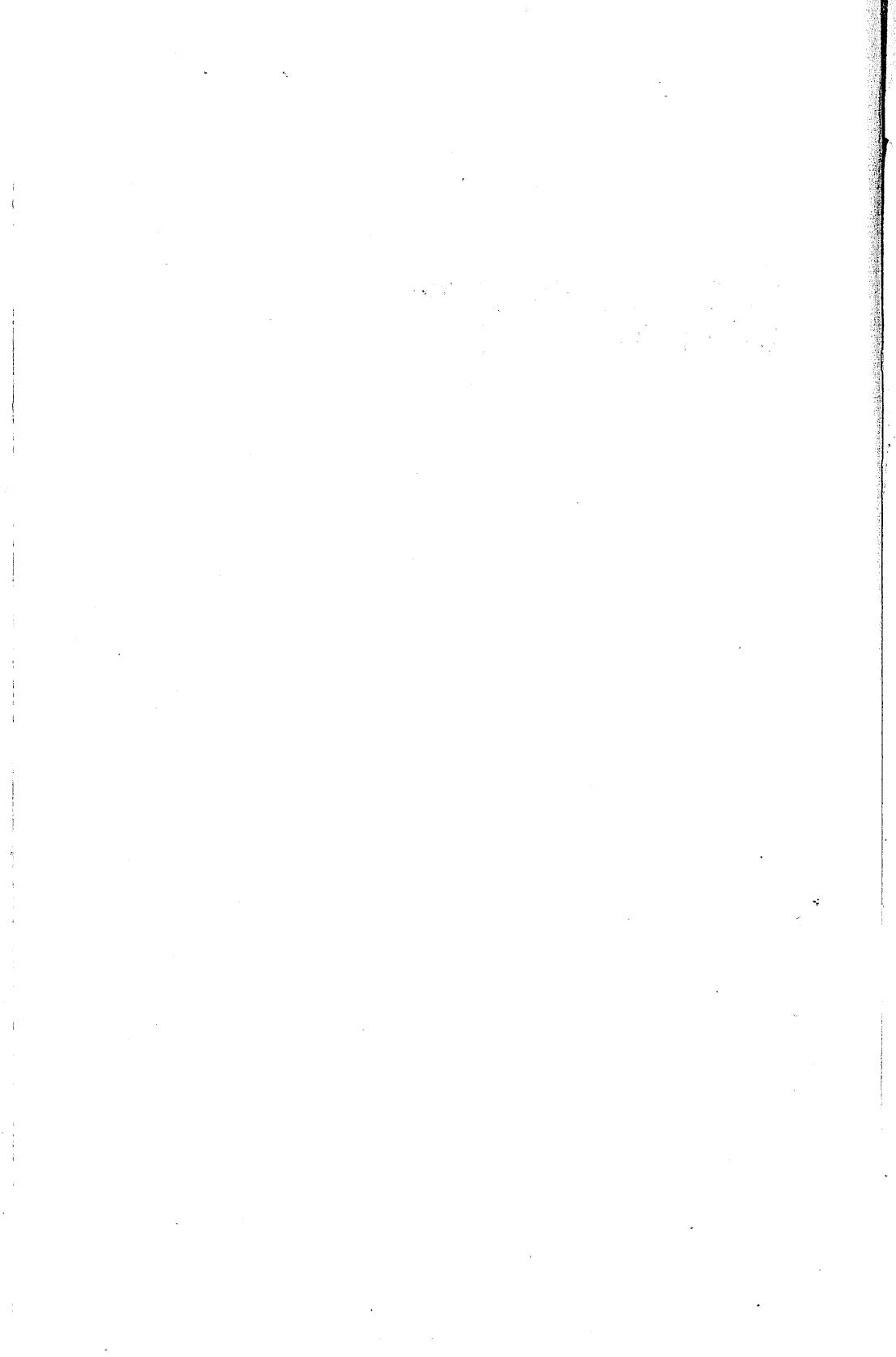
BRIAN L. STEVENS

Atlanta, Georgia

FRANK L. LEWIS

Arlington, Texas

AIRCRAFT CONTROL AND SIMULATION



CHAPTER 1

EQUATIONS OF MOTION

1.1 INTRODUCTION

The State-Space Formulation

Our purpose in this chapter is to derive the equations of motion of a *rigid* aircraft and express them in the state-space form:

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}, \mathbf{U}), \quad (1.1-1)$$

where \mathbf{X} is the $(n \times 1)$ *state vector*, \mathbf{U} is the $(m \times 1)$ *control vector*, and \mathbf{f} is a vector-valued nonlinear (in general) function of the individual states and controls. This vector equation symbolizes the n first-order, coupled ordinary differential equations (ODEs)

$$\dot{X}_1 = f_1(X_1, X_2, \dots, X_n, U_1, \dots, U_m)$$

⋮

$$\dot{X}_n = f_n(X_1, X_2, \dots, X_n, U_1, \dots, U_m),$$

where the f_i represent different nonlinear functions of the n state variables, X_i , and m inputs, U_i . In addition to the *state equation* (1.1-1) we shall also require an *output equation* of the general form

$$\mathbf{Y} = \mathbf{g}(\mathbf{X}, \mathbf{U}), \quad (1.1-2)$$

where \mathbf{Y} is a $(p \times 1)$ *output vector* and \mathbf{g} represents a set of nonlinear

equations similar to f. The state variables, X_i , will be any set of variables that completely define the *state*. The state is an indication of the stored energy of a system (i.e., potential and kinetic of the aircraft) and its distribution. We will normally choose a minimal set of state variables (i.e., independent state variables) and use the obvious orthogonal components of position and velocity (translational and angular) for this purpose. The output variables, Y_i , will usually correspond to physical quantities for which measurement sensors have been supplied.

Capital (uppercase) symbols will be used for the state, control, and output vectors and their elements because at this stage they will represent the actual values of aircraft variables. Later we shall derive linear, constant-coefficient state and output equations of the form

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \quad \mathbf{x} \in R^n, \quad \mathbf{u} \in R^m \quad (1.1-3a)$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}, \quad \mathbf{y} \in R^p \quad (1.1-3b)$$

where the lowercase symbols for the state, control, and output vectors indicate that they are deviations from some nominal values (the steady-state flight conditions).

As an alternative to the state equations, the aircraft mathematical model could be built from a set of simultaneous ODEs of various orders. In this case, if we wished to examine the dynamic behavior starting from some point on a trajectory, we would need to know the initial values of the derivatives of various orders. Because of the “canonical” first-order form of the state equations, all of the necessary initial-condition information is carried by the state vector. In fact, the state variables can be defined as a set of variables such that knowledge of the state vector at a particular time, and the control vector after this time, completely defines the motion (state trajectory) from that time on [Kalman, 1963]. This concept of state implies our earlier definition—that the state vector defines the energy stored in the dynamics at each instant of time; this property will be useful when we come to choose a set of state variables. Obviously, reduction of equations of motion to state-space form may include simply redefining the derivatives of some state variables as state variables in their own right.

The usefulness of the state-space description will become increasingly evident as we proceed. At this stage we simply note that the functions f_i cannot in general be represented analytically because they encompass a number of nonlinear effects, in the form of a large body of tabular aerodynamic data. Despite this there is no difficulty involved in computing a trajectory. This is because the expression for the derivative vector [equation (1.1-1)], and a starting value for the state vector, constitute an initial value problem in the mathematical theory of ordinary differential equations. Mathematicians long ago developed the theory and algorithms for numerical

integration of this type of problem. One class of numerical integration algorithms (Runge-Kutta) allows the state to be computed successively at discrete time instants, given only the previous state and the expression needed to evaluate the corresponding derivatives (i.e., the state equations). Another class (multistep methods) uses more than one previous state. Reliable computer programs are readily available to perform the numerical integration, and a simulated trajectory can be "flown" by supplying the control input U . The simulation can be stopped, and restarted at any time, by storing the value of the state vector.

Degrees of Freedom

The term *rigid aircraft* used earlier implies that all points in the aircraft structure maintain fixed relative positions in space at all time. However, it is a common (and disturbing) experience to observe flexing of the wings of a large passenger aircraft during flight. The deflection of the wingtips, corresponding to the difference between sitting on the runway and fully loaded flight, may be several feet. Even a fighter aircraft with a sturdy fuselage and short stubby wings will exhibit flexibility effects. The natural mathematical description of a flexible aircraft is in terms of partial differential equations, and a good deal of preliminary analysis would be needed to arrive at the ODEs required for a state model. The interaction of flexibility effects with the aerodynamics greatly complicates this model and is the subject of ongoing research. Nevertheless, the rigid model is of fundamental importance and will be the focus of our attention.

The equations of motion of a rigid body can be separated (decoupled) into rotational equations and translational equations if the coordinate origin is chosen to be at the *center of mass* [Wells, 1967]. The rotational motion of the aircraft will then be equivalent to yawing, pitching, and rolling motions about the center of mass as if it were a fixed point in space [note that we shall use the common term *center of gravity* (cg) synonymously with center of mass]. The remaining components of the motion will be three components of translation of the cg. Therefore, the state model derived here will be a six-degrees-of-freedom (6-DOF) model. We shall not consider degrees of freedom associated with flexible modes such as body bending and wing flexure.

Our choice of state variables will be a natural one. Three components of position are needed to specify potential energy in the gravitational field. Three components of velocity are required to specify translational kinetic energy, and three components of angular velocity to specify rotational kinetic energy. We shall find that three additional state equations are needed for attitude states. The attitude variables are used to specify orientation relative to the gravity vector. Therefore, the state vector \mathbf{X} of the basic model will contain 12 components (state variables).

1.2 MATHEMATICAL PRELIMINARIES

Vector Formulation

The starting point for the derivation of the aircraft equations of motion will be the vector form of Newton's second law of motion. We shall use a mixture of vector analysis and linear algebra, and present a summary of the relevant theory. More details are available in texts that cover classical mechanics and analytical dynamics [Goldstein, 1981; Meirovitch, 1970; Greenwood, 1965; Becker, 1954], and to a lesser extent in books on flight dynamics and control [Etkin, 1972; McRuer et al., 1973; Perkins and Hage, 1949].

The magnitudes and directions of the three-dimensional vectors used in the dynamic analysis are in general independent of the coordinate system in which they are expressed, but their components are not. Therefore, it is common practice to use superscripts and subscripts on a vector symbol to indicate both the nature of the vector and the frame in which it is expressed. A comprehensive superscript/subscript notation will not be defined here because our needs are limited. Subscripts will be used to define either the appropriate frame of reference or the nature of the particular vector, depending on context.

In the development of the rigid-body equations of motion we shall need to express a given vector in several different coordinate frames. Position vectors are relative to a particular coordinate origin because there is no concept of absolute position. Thus when we specify a position in a new coordinate frame, the magnitude and direction of the position vector are changed. Velocity, acceleration, and higher derivatives of position may be either relative or absolute. The absolute reference for these quantities is inertial space, in which Newton's laws of motion are valid. An inertial frame can be defined as stationary with respect to the fixed stars. When we specify an absolute velocity or acceleration in a new coordinate frame its magnitude and direction are unchanged; only the coordinates of the vector are changed. The velocity or acceleration vector of a body may, if required, be specified in terms of a coordinate frame that is fixed in the body (i.e., moving with the body). This simply means that the vector is resolved along the instantaneous directions of the coordinate axes as if it emanated from the coordinate origin. Therefore, we do not have to consider the translation of the origin of the coordinate system, only its orientation. This requires an understanding of coordinate rotations, and we shall summarize the relevant theory.

Notation and Conventions

We will use three coordinate axes that are mutually perpendicular, and refer to the individual axes as the x , y , and z axes. The coordinate frames will be right-handed, by which we mean that a right-hand-thread screw rotated from the positive x -axis toward the positive y -axis, through the smaller angle (i.e.,

the 90° angle) will advance in the positive z -direction. We shall think of a vector as emanating from the coordinate origin; therefore, the components of a vector and angle between two vectors will be clearly identifiable, and two vectors will define a plane. The components of a vector along these coordinate axes will be referred to as the x , y , and z components, respectively, and unit vectors along the respective axes will be denoted by \mathbf{i} , \mathbf{j} , and \mathbf{k} . A vector, \mathbf{v} , will be denoted by a boldface symbol and represented in terms of its coordinates as a column array or a set of unit vectors multiplied by the coordinates. Thus

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k},$$

and a subscript will be added to denote the name of the vector or the reference frame as required. The boldface notation will be discontinued in Chapter 3, where most quantities are vectors in finite-dimensional spaces of arbitrary dimension. Boldface will not be used here for matrices.

Scalar and Dot Products

The dot product of vector analysis is defined by

$$\mathbf{v}_1 \cdot \mathbf{v}_2 = \|\mathbf{v}_1\| \|\mathbf{v}_2\| \cos \theta \quad (1.2-1)$$

where $\|\cdot\|$ denotes the Euclidean norm (length) of the enclosed vector, and θ is the included angle between the vectors. It is commutative and distributive; thus

$$\mathbf{v}_1 \cdot \mathbf{v}_2 = \mathbf{v}_2 \cdot \mathbf{v}_1$$

and

$$(\mathbf{v}_1 + \mathbf{v}_2) \cdot \mathbf{v}_3 = \mathbf{v}_1 \cdot \mathbf{v}_3 + \mathbf{v}_2 \cdot \mathbf{v}_3.$$

The orthogonal unit vectors satisfy the dot product relationships

$$\mathbf{i} \cdot \mathbf{i} = \mathbf{j} \cdot \mathbf{j} = \mathbf{k} \cdot \mathbf{k} = 1$$

$$\mathbf{i} \cdot \mathbf{j} = \mathbf{j} \cdot \mathbf{k} = \mathbf{k} \cdot \mathbf{i} = 0.$$

Using these relationships, the dot product of two vectors can be expressed as

$$\mathbf{v}_1 \cdot \mathbf{v}_2 = x_1 x_2 + y_1 y_2 + z_1 z_2. \quad (1.2-2)$$

This expression is identical to the scalar or inner product of linear algebra in

three-dimensional space:

$$\mathbf{v}_1^T \mathbf{v}_2 = [x_1 \quad y_1 \quad z_1] \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix},$$

where the superscript T denotes the transpose of a vector. The length of a vector is given by

$$\|\mathbf{v}\| = (\mathbf{v}^T \mathbf{v})^{1/2} = (\mathbf{v} \cdot \mathbf{v})^{1/2}. \quad (1.2-3)$$

Cross-Products

The cross-product of \mathbf{u} and \mathbf{v} , denoted by $\mathbf{u} \times \mathbf{v}$, is a vector \mathbf{w} that is normal to the plane of \mathbf{u} and \mathbf{v} and is in a direction such that $\mathbf{u}, \mathbf{v}, \mathbf{w}$ form a right-handed system. The length of \mathbf{w} is given by the product of the lengths of \mathbf{u} and \mathbf{v} and the sine of the smaller angle between them. The cross-product has the following properties:

$$\begin{aligned} \mathbf{u} \times \mathbf{v} &= -(\mathbf{v} \times \mathbf{u}) && \text{(anticommutative)} \\ a(\mathbf{u} \times \mathbf{v}) &= (a\mathbf{u}) \times \mathbf{v} = \mathbf{u} \times (a\mathbf{v}) && \text{(associative)} \\ \mathbf{u} \times (\mathbf{v} + \mathbf{w}) &= (\mathbf{u} \times \mathbf{v}) + (\mathbf{u} \times \mathbf{w}) && \text{(distributive)} \\ \mathbf{u} \cdot (\mathbf{v} \times \mathbf{w}) &= \mathbf{v} \cdot (\mathbf{w} \times \mathbf{u}) = \mathbf{w} \cdot (\mathbf{u} \times \mathbf{v}) && \text{(scalar triple product)} \\ \mathbf{u} \times (\mathbf{v} \times \mathbf{w}) &= \mathbf{v}(\mathbf{w} \cdot \mathbf{u}) - \mathbf{w}(\mathbf{u} \cdot \mathbf{v}) && \text{(vector triple product).} \end{aligned} \quad (1.2-4)$$

As an aid to remembering the form of the triple products, note the cyclic permutation of the vectors involved.

The cross-products of the unit vectors of a right-handed coordinate system satisfy the equations

$$\begin{aligned} \mathbf{i} \times \mathbf{i} &= \mathbf{j} \times \mathbf{j} = \mathbf{k} \times \mathbf{k} = 0 \\ \mathbf{i} \times \mathbf{j} &= \mathbf{k} \\ \mathbf{j} \times \mathbf{k} &= \mathbf{i} \\ \mathbf{k} \times \mathbf{i} &= \mathbf{j}. \end{aligned}$$

Also remember that $\mathbf{j} \times \mathbf{i} = -\mathbf{i} \times \mathbf{j} = -\mathbf{k}$, and so on. From these properties we can derive a formula for the cross-product of two vectors; a convenient way of remembering the formula is to write it so that it resembles the expansion of a determinant. Thus, if

$$\mathbf{u} = x_1 \mathbf{i} + y_1 \mathbf{j} + z_1 \mathbf{k} \quad \text{and} \quad \mathbf{v} = x_2 \mathbf{i} + y_2 \mathbf{j} + z_2 \mathbf{k},$$

the mnemonic is

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{vmatrix} = \mathbf{i} \begin{vmatrix} y_1 & z_1 \\ y_2 & z_2 \end{vmatrix} - \mathbf{j} \begin{vmatrix} x_1 & z_1 \\ x_2 & z_2 \end{vmatrix} + \mathbf{k} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix}. \quad (1.2-5)$$

We shall often make use of the fact that the cross-product operation can be replaced by a vector-matrix product as follows. Let ω and \mathbf{v} be the following two vectors:

$$\omega = \begin{bmatrix} P \\ Q \\ R \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Then it is easy to show (Problem 1.2-1), using the determinant formula for the cross-product, that

$$\omega \times \mathbf{v} = \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \equiv \Omega \mathbf{v}. \quad (1.2-6)$$

The same idea can be applied to the vector triple product. For example,

$$\omega \times (\omega \times \mathbf{v}) = \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix}^2 \begin{bmatrix} x \\ y \\ z \end{bmatrix} \equiv \Omega^2 \mathbf{v}. \quad (1.2-7)$$

The symbol Ω will be used throughout to denote the *cross-product matrix* corresponding to the operation $(\omega \times \cdot)$. A cross-product matrix is skew-symmetric, that is,

$$\Omega^T = -\Omega, \quad (1.2-8)$$

and therefore the square of the cross-product matrix is symmetric. Note that in the general case the matrix operations must be written in the same order as the vector operations but may be performed in any order internally.

We shall need to use the cross-product operation to represent various physical quantities and operations in mechanics. First note that the angular velocity of a rotating body can be treated as a vector. This vector is perpendicular to the plane in which the rotation takes place (i.e., parallel to the axis of rotation), with magnitude equal to the angular speed. The positive sense of the angular velocity vector is given by the right-hand rule. Now, with this vector representation of angular velocity, we may use the cross-product operation to calculate the velocities and accelerations that arise from angular motion. Consider a rigid body rotating about a fixed axis, with angular

velocity ω , and a point with position vector \mathbf{p} relative to an origin on the axis. The tangential velocity and centripetal acceleration of the point are given by

$$\text{tangential velocity, } \mathbf{v} = \omega \times \mathbf{p} \quad (1.2-9a)$$

$$\text{centripetal acceleration} = \omega \times \mathbf{v} = \omega \times (\omega \times \mathbf{p}). \quad (1.2-9b)$$

The reader should check that the order of the terms in these cross-products does indeed lead to velocities and accelerations whose positive sense corresponds to the physical effect.

In the same manner as angular velocity, torque (moment) can be represented as a vector quantity. Thus the moment about the origin of a force, \mathbf{F} , acting through a point whose position vector is \mathbf{r} is given by

$$\text{moment} = \mathbf{r} \times \mathbf{F}. \quad (1.2-10)$$

This formula will be used later to calculate angular momentum (i.e., the moment of translational momentum).

Coordinate Rotations

In aerospace applications we commonly need to express a given vector in terms of a new Cartesian coordinate frame, where the new frame has the same origin as the old, but a different orientation. If "1" and "2" are two such frames and \mathbf{v}_1 is a vector expressed in terms of frame 1, then its coordinates in frame 2 are given by

$$\mathbf{v}_2 = R\mathbf{v}_1, \quad (1.2-11)$$

where $R(3 \times 3)$ is a matrix that describes the orientation of frame 2 with respect to frame 1.

The coordinate rotation must leave the length of a vector unchanged. The change of length under the rotation above is

$$\mathbf{v}_2^T \mathbf{v}_2 = (R\mathbf{v}_1)^T R\mathbf{v}_1 = \mathbf{v}_1^T R^T R \mathbf{v}_1$$

and the length is preserved if

$$R^T R = I = R R^T. \quad (1.2-12)$$

This is the definition of an orthogonal matrix, and it makes the inverse matrix particularly easy to determine ($R^{-1} = R^T$). It also implies that the columns (and also the rows) of the rotation matrix form a set of orthonormal vectors

$$R = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3], \quad \mathbf{r}_i^T \mathbf{r}_j = \begin{cases} 0, & i \neq j \\ 1, & i = j. \end{cases} \quad (1.2-13)$$

We can go a step further than the relationship (1.2-13). The unit vectors in one frame rotate into the column vectors \mathbf{r}_i in the new frame, for example,

$$\mathbf{r}_1 = R \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Therefore, the columns of the rotation matrix form a set of unit vectors in the new coordinate frame. The rotation matrix preserves the “handedness” of a coordinate frame, so the columns of the rotation matrix satisfy the cross-product relationships

$$\begin{aligned} \mathbf{r}_1 \times \mathbf{r}_2 &= \mathbf{r}_3 \\ \mathbf{r}_2 \times \mathbf{r}_3 &= \mathbf{r}_1 \\ \mathbf{r}_3 \times \mathbf{r}_1 &= \mathbf{r}_2. \end{aligned} \quad (1.2-14)$$

Using this property, it is easy to show (Problem 1.2-2) that a rotation matrix is distributive with the cross-product operation in the following way:

$$R(\mathbf{u} \times \mathbf{v}) = (R\mathbf{u}) \times (R\mathbf{v}), \quad (1.2-15)$$

that is, the rotated cross-product of two vectors is the same as the cross-product of the two rotated vectors. This distributive rule for the rotation matrix will prove useful in the later analysis.

The orientation of one three-dimensional coordinate frame with respect to another can always be described by a sequence of three plane rotations, that is, rotations about a single axis at a time. This provides an easy way to determine the rotation matrix required to express a particular vector in terms of different, orthogonal, coordinate frames. Consider Fig. 1.2-1, in which frame 2 has been rotated relative to frame 1. The frames are right-handed with the z -axis coming out of the page, and the rotation is a right-handed rotation about the z -axis, through an angle θ . Assume that the coordinates of the vector \mathbf{v} are known in frame 1, and that we need to know its coordinates in frame 2. The required plane rotation matrix must obviously approach the identity matrix as θ approaches zero, must contain sine and cosine terms, and must leave the z -component unchanged. Simple geometry applied to the figure (Problem 1.2-3) shows that the required transformation is

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}. \quad (1.2-16)$$

The negative sine element always occurs on the row above the one containing the unit element when the second frame is reached by a right-handed

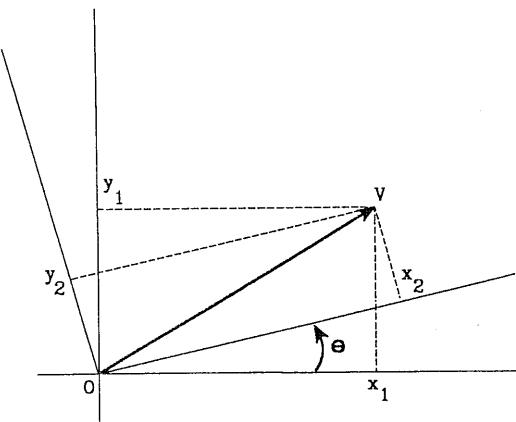


Fig. 1.2-1 Plane coordinate rotation.

rotation (note that the third row is considered as being above the first row). Note also that the plane rotation matrix is skew-symmetric, and changing the sign of the rotation yields the matrix transpose.

Three-dimensional coordinate rotations can now be built up as a sequence of plane rotations. The fact that the individual rotations are not commutative can be checked by a simple experiment. Therefore, although the order of the sequence can be chosen arbitrarily, the same order must be maintained ever after. The common aerospace industry practice will be stated and followed when we come to specific cases. Note that if a vector is expressed in a new coordinate system by a sequence of rotations as

$$\mathbf{v}_2 = R_3 R_2 R_1 \mathbf{v}_1,$$

then the inverse operation is given by

$$\mathbf{v}_1 = (R_3 R_2 R_1)^{-1} \mathbf{v}_2 = R_1^{-1} R_2^{-1} R_3^{-1} \mathbf{v}_2 = R_1^T R_2^T R_3^T \mathbf{v}_2. \quad (1.2-17)$$

A useful method of quickly deriving coordinate rotation matrices has also been given by Pio [Pio, 1964, 1966].

Properties of Linear Transformations

Consider the equation

$$\mathbf{u} = A\mathbf{v}, \quad (1.2-18)$$

in which A is an arbitrary constant matrix. Each element of the vector \mathbf{u} is a linear combination of the elements of \mathbf{v} , so this equation represents a *linear*

transformation of the vector \mathbf{v} . Now suppose that (1.2-18) represents the relationship between two Euclidean vectors expressed in a certain coordinate frame, and that we need the relationship between the vectors in a rotated coordinate frame. Let the rotation be described by the matrix R ; the respective vectors expressed in the rotated coordinate frame are

$$\mathbf{u}_1 = R\mathbf{u}, \quad \mathbf{v}_1 = R\mathbf{v}.$$

It follows that the required relationship is given by

$$\mathbf{u}_1 = R\mathbf{A}\mathbf{v} = R\mathbf{A}R^{-1}\mathbf{v}_1. \quad (1.2-19)$$

The transformation $R\mathbf{A}R^{-1}$ is a *similarity transformation* of the original coefficient matrix \mathbf{A} . Because of orthogonality, similarity transformations with rotation matrices are equivalent to *congruence transformations* ($R\mathbf{A}R^T$). In general, the coefficient matrix of a linear relationship is transformed by a similarity transformation when a coordinate rotation, or any reversible (non-singular) change of variables, is performed. Such a transformation will occur when we change coordinate systems in the rigid-body equations of motion in Chapter 2.

As an example of a similarity transformation in n dimensions, consider the linear state-space equations (1.1-3) under the nonsingular transformation of variables

$$\mathbf{x} = M\mathbf{z}, \quad (1.2-20)$$

where M is a constant (time-invariant) matrix. The result of this transformation is

$$\dot{\mathbf{z}} = (M^{-1}\mathbf{A}M)\mathbf{z} + (M^{-1}\mathbf{B})\mathbf{u} \quad (1.2-21a)$$

$$\mathbf{y} = CM\mathbf{z} + D\mathbf{u}. \quad (1.2-21b)$$

The coefficient matrix \mathbf{A} of the original state equations has been transformed by a similarity transformation, thus reflecting a different set of relationships between the new variables. This change of internal variables in the state-space system is called a *state-space transformation*. It does not change the overall input-output (i.e., from \mathbf{u} to \mathbf{y}) behavior, and the coupling of the inputs and outputs ($M^{-1}\mathbf{B}$ and CM , respectively) has changed to make this possible.

Eigenvalues and Eigenvectors

An important question that arises in the study of linear systems is whether there exists a vector such that the transformation (1.2-18), in n dimensions, produces another vector in the same direction. If \mathbf{v} is such a vector, it must

satisfy the equation

$$A\mathbf{v} = \lambda\mathbf{v}, \quad \mathbf{v} \in R^n \quad (1.2-22)$$

where λ is a (scalar) constant of proportionality. A rearrangement of (1.2-22) gives the set of homogeneous linear equations

$$(A - \lambda I)\mathbf{v} = 0, \quad (1.2-23)$$

which has a non-null solution for \mathbf{v} if, and only if, the determinant of the coefficient matrix is zero [Strang, 1980]; that is,

$$|A - \lambda I| = 0. \quad (1.2-24)$$

This determinant is an n th-order polynomial in λ , called the *characteristic polynomial* of A , so there may be up to n distinct solutions for λ . Each solution λ_i is known as an *eigenvalue* or *characteristic value* of the matrix A . The associated invariant vector defined by (1.2-22) is known as a *right eigenvector* of A (the left eigenvectors of A are the right eigenvectors of its transpose A^T).

If we consider a linear relationship such as the transformation (1.2-18) or the state-space equation (1.1-3a), the eigenvalues and eigenvectors obviously provide a rather special characterization of the relationship through its coefficient matrix. If the linear relationship is subjected to a nonsingular change of variables, such as a coordinate rotation or (1.2-20), the eigenvalues are unchanged. To show this it is only necessary to examine the eigenvalues of the coefficient matrix under a similarity transformation, as in (1.2-21a). The eigenvalues are given by

$$|(\lambda I - M^{-1}AM)| = 0,$$

which can be rewritten as

$$|(\lambda M^{-1}M - M^{-1}AM)| = 0.$$

The determinant of a product of square matrices is equal to the product of the individual determinants; therefore,

$$|M^{-1}| \times |(\lambda I - A)| \times |M| = 0. \quad (1.2-25)$$

This equation is satisfied by the eigenvalues of the matrix A .

Now consider a special similarity transformation that will reduce the linear equations to a canonical (standard) form. For simplicity, assume that all of the n eigenvalues of the coefficient matrix, A , are distinct. In this case the n eigenvectors, \mathbf{v}_i , can be shown to form a linearly independent set; therefore, they can be used to form the columns of a nonsingular transformation matrix.

This matrix is called the *modal matrix*, M , and

$$M = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n].$$

According to the eigenvector/eigenvalue defining equation (1.2-22), if M is a modal matrix, we find that

$$AM = MJ, \quad J = \text{diag}(\lambda_1 \cdots \lambda_n),$$

or

$$M^{-1}AM = J. \quad (1.2-26)$$

Returning to the state equations, (1.2-21), if M is a modal matrix there, the z -variables are *modal coordinates*. When expressed in modal coordinates the state equation (1.2-21a) becomes much simpler in form, but the state variables usually lose their original physical significance.

When some of the eigenvalues of A are repeated (i.e., multiple), it may not be possible to find a set of n linearly independent eigenvectors. Also, in the case of repeated eigenvalues, the result of the similarity transformation (1.2-26) is in general a *Jordan-form matrix* [Wilkinson and Golub, 1976]. In this case the matrix J in (1.2-26) may have some unit entries on the superdiagonal. These entries are associated with blocks of repeated eigenvalues on the main diagonal.

The physical systems that we shall analyze will have coefficient matrices with real elements. Consequently, the characteristic polynomial (1.2-24) will have real coefficients, and the eigenvalues of the coefficient matrices will be real or will occur in complex conjugate pairs. Assume that (1.2-24) has a pair of complex conjugate eigenvalues $(\alpha \pm j\beta)$, and associated eigenvectors $(\mathbf{u} \pm j\mathbf{w})$, so that

$$A(\mathbf{u} + j\mathbf{w}) = (\alpha + j\beta)(\mathbf{u} + j\mathbf{w})$$

$$A(\mathbf{u} - j\mathbf{w}) = (\alpha - j\beta)(\mathbf{u} - j\mathbf{w}).$$

The real and imaginary parts of these equations satisfy

$$A\mathbf{u} = \alpha\mathbf{u} - \beta\mathbf{w}$$

$$A\mathbf{w} = \beta\mathbf{u} + \alpha\mathbf{w};$$

therefore, we can avoid complex arithmetic in (1.2-26) by modifying it

according to

$$A[\cdots \mathbf{u}, \mathbf{w} \cdots] = [\cdots \mathbf{u}, \mathbf{w} \cdots]$$

$$\times \begin{bmatrix} \lambda_1 & 0 & \cdots & \cdot & 0 & 0 \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\ 0 & \cdots & 0 & \alpha & \beta & 0 \\ 0 & \cdots & 0 & -\beta & \alpha & 0 \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\ 0 & 0 & \cdots & \cdot & 0 & \lambda_n \end{bmatrix}. \quad (1.2-27)$$

This procedure is very useful in numerical algorithms.

Further insight into the significance of eigenvalues and eigenvectors and a better understanding of coordinate rotations can be obtained by examining the eigenvalues of the 3×3 rotation matrix. Goldstein [1981, p. 158] shows that any nontrivial rotation matrix has one, and only one, eigenvalue equal to +1, and that this corresponds to a theorem proved by the mathematician Leonhard Euler (1707–1783) for a rigid body. The other two eigenvalues are a complex conjugate pair with unit magnitude, and can be written as $(\cos \Phi \pm j \sin \Phi)$. Therefore, using a similarity transformation and the result given by (1.2-27), it is possible to transform any rotation matrix, R , to the form of a plane rotation matrix, P , for example,

$$P = \begin{bmatrix} \cos \Phi & \sin \Phi & 0 \\ -\sin \Phi & \cos \Phi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.2-28)$$

Equation (1.2-28) corresponds to a single rotation through an angle Φ about the z -axis. It shows that the orientation of one coordinate frame with respect to another is uniquely determined by a single rotation about a unique axis (the Euler axis), and this is the essence of Euler's theorem. We shall make use of these ideas later when we describe the use of four *quaternion parameters* to represent orientation.

Similarity transformations and eigenanalysis also provide insight into the linear state equation. Suppose that the transformed state equations (1.2-21) are in modal coordinates and the eigenvalues of the A matrix are distinct. Then the state and output equations become

$$\dot{\mathbf{z}} = J\mathbf{z} + M^{-1}\mathbf{B}\mathbf{u} \quad (1.2-29a)$$

$$\mathbf{y} = CM\mathbf{z} + D\mathbf{u}. \quad (1.2-29b)$$

The state equation (1.2-29a) now consists of a set of uncoupled differential equations of the form

$$\dot{z}_i = \lambda_i z_i + \mathbf{b}'_i \mathbf{u}, \quad i = 1, \dots, n,$$

where $\{\lambda_i\}$ is the set of eigenvalues and \mathbf{b}'_i is the i th row of the matrix $M^{-1}\mathbf{B}$.

Each equation has a solution of the form

$$z_i = e^{\lambda_i t} z_i(0) + p_i(t), \quad i = 1, \dots, n. \quad (1.2-30)$$

The first term on the right-hand side of (1.2-30) is the *homogeneous solution* and the second term, $p_i(t)$, is the *forced solution* for the specific input. The homogeneous solutions contain the *natural modes* of the system, and these modes are exponential functions characterized by the eigenvalues of the system A matrix. When a complex-conjugate pair of eigenvalues occurs, they combine to form a real *oscillatory mode* as follows:

$$\begin{aligned} \lambda_i &= \sigma_i + j\omega_i, \quad \lambda_i^* = \sigma_i - j\omega_i \\ z_i(0)e^{\lambda_i t} + z_i^*(0)e^{\lambda_i^* t} &= 2 \operatorname{Re}[z_i(0)e^{\lambda_i t}] \\ &= 2|z_i(0)e^{\lambda_i t}| \cos \angle z_i(0)e^{\lambda_i t} \\ &= 2|z_i(0)|e^{\sigma_i t} \cos(\omega_i t + \angle z_i(0)), \end{aligned} \quad (1.2-31)$$

where the * indicates the complex conjugate, Re is the real-part operator, $| \ |$ indicates magnitude of a complex number, and \angle indicates the angle of the complex number. It follows that the exponential damping of the mode is given by the real part of the eigenvalue, and the actual (or "damped") frequency of oscillation is given by the imaginary part.

Because the eigenvalues are invariant under a nonsingular transformation of variables, the same modes will show up in the system output irrespective of the choice of state variables. The modes are characteristic of the particular physical system. Furthermore, a necessary and sufficient condition for a linear system to be stable is that the real parts of its eigenvalues all be negative. We develop and use eigensystem techniques in later chapters; in this section we have attempted to illustrate the significance of the eigenvalues and the role played by linear transformations. These concepts will occur repeatedly in our subsequent work.

Differentiation of Vectors

To apply Newton's laws we need to define the rate of change of a vector quantity. More specifically, it is necessary to find the derivative of an absolute velocity vector and express this derivative in other coordinate systems. The other coordinate systems may be rotating and the derivative in question will correspond to the instantaneous projection of the absolute derivative on the rotating coordinate system.

In coordinate rotation we simply expressed the same vector in different coordinate frames. In differentiation we use a limiting process $\delta \mathbf{v}/\delta t$ to create a new vector, and the frame in which the derivative is taken must be specified as well as the frame in which the answer is expressed.

A subscript will be used on the derivative operator, d/dt , to indicate the frame in which the derivative was taken, and the subscript on the vector will indicate the frame in which the answer is expressed. We shall omit the subscript on the derivative operator, or use a superscript dot, to indicate differentiation of a vector in its own frame. A superscript dot will also indicate differentiation of a scalar. Thus, if \mathbf{v} is the vector

$$\mathbf{v} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k},$$

then

$$\frac{d}{dt}(\mathbf{v}) \equiv \dot{\mathbf{v}} \equiv \dot{x}\mathbf{i} + \dot{y}\mathbf{j} + \dot{z}\mathbf{k}. \quad (1.2-32)$$

Next we shall derive some important equations that will be used several times throughout Chapters 1 and 2. If R is the rotation matrix that takes vectors from a frame A to a frame B , then

$$\mathbf{v}_B = R\mathbf{v}_A, \quad (1.2-33)$$

and an expression for the derivative of \mathbf{v}_B , with respect to the A frame, is

$$\frac{d}{dt_A}(\mathbf{v}_B) = R\dot{\mathbf{v}}_A. \quad (1.2-34)$$

Compare this with

$$\frac{d}{dt_B}(\mathbf{v}_B) = \dot{\mathbf{v}}_B = \dot{R}\mathbf{v}_A + R\dot{\mathbf{v}}_A, \quad (1.2-35)$$

which follows from a straightforward application of the product rule for differentiation to (1.2-33). Note that $R(t)$ is time varying if frame B is rotating relative to frame A . Now, using (1.2-35) to substitute for the right-hand side of (1.2-34), we find that

$$\frac{d}{dt_A}(\mathbf{v}_B) = \dot{\mathbf{v}}_B - \dot{R}\mathbf{v}_A. \quad (1.2-36)$$

Equation (1.2-36) shows that the derivative of a vector, with respect to another frame, has two components. The first component is the derivative with respect to its own frame, and the second component is nonzero when its own frame is rotating relative to the frame in which the derivative was taken.

The equation is commonly expressed in the following form, as the *theorem of Coriolis*:

$$\frac{d}{dt_A}(\mathbf{v}_B) = \dot{\mathbf{v}}_B + \boldsymbol{\omega}_B \times \mathbf{v}_B. \quad (1.2-37)$$

Here $\boldsymbol{\omega}_B$ is the angular velocity vector of frame B relative to frame A . Note that all of the vectors are now expressed in the same frame. This equation can be derived by a formal limiting process (Problem 1.2-4), where the cross-product component $(\boldsymbol{\omega}_B \times \mathbf{v}_B) dt$ is the tangential component of the vector \mathbf{v}_B due to the rotation. The theorem of Coriolis is conventionally invoked in the derivation of the rigid-body equations of motion, and we shall use it later for that purpose.

A comparison of (1.2-36) and (1.2-37) shows that a formal kinematic relationship must exist between the rate of change of the rotation matrix and the angular velocity vector $\boldsymbol{\omega}_B$. To establish the relationship, substitute a cross-product matrix, Ω_B , for the cross-product operation in (1.2-37), and use (1.2-33) to substitute for \mathbf{v}_B . Then, comparing the result with (1.2-36), we see that

$$\dot{R} = -\Omega_B R. \quad (1.2-38a)$$

A useful result that follows from (1.2-38a) is

$$\Omega_B = R \dot{R}^T. \quad (1.2-38b)$$

Equation (1.2-38a) is a fundamental kinematic relationship for relative rotation between two coordinate frames. It relates the two different sets of angular rates describing the motion of the B frame relative to the A frame. These are the rates of change of the B frame relative orientation angles (in R), and the components (in Ω_B) of the relative angular velocity vector measured in the B frame. It is important in guidance and navigation computations and is referred to as the *strapdown equation*. The strapdown equation can be numerically integrated in the onboard computer of a vehicle with an inertial navigation system, in order to keep track of the orientation (attitude). Rate sensors aligned with the vehicle reference axes are used to measure the angular rates in Ω_B , and the rotation matrix is computed as a function of time. The attitude information contained in the rotation matrix can then be used to resolve the outputs of accelerometers, “strapped” to the vehicle body, into an “inertial” reference frame (i.e., an absolutely fixed reference frame). The acceleration due to gravity is known in this frame, and velocity and position are computed by integrating the resolved accelerations. This is in contrast to an inertial navigation system where the accelerometers are mounted on a stable (with respect to inertial space) platform.

The strapdown equation can also be derived directly from (1.2-37) in the following manner. According to (1.2-33), a unit vector in the A frame is a column, \mathbf{r}_i , of the rotation matrix when expressed in the B frame; thus

$$\mathbf{r}_1 = R \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_A.$$

By definition the derivative of this B -frame vector, with respect to the A frame, is a null vector. Therefore, applying the cross-product derivative formula (1.2-37), we get

$$0 = \frac{d}{dt_A}(\mathbf{r}_1) = \dot{\mathbf{r}}_1 + \boldsymbol{\omega}_B \times \mathbf{r}_1.$$

Repeating this analysis for the other columns of the rotation matrix gives

$$\dot{\mathbf{r}}_i = -\boldsymbol{\omega}_B \times \mathbf{r}_i, \quad i = 1, 2, 3, \quad (1.2-39)$$

which is the strapdown equation column by column.

Finally, we shall have occasion to use the derivative of a cross product. This follows the form of the product rule for differentiation:

$$\frac{d}{dt}(\mathbf{u} \times \mathbf{v}) = \dot{\mathbf{u}} \times \mathbf{v} + \mathbf{u} \times \dot{\mathbf{v}}. \quad (1.2-40)$$

This completes the mathematical preliminaries; we now have enough analytical tools to allow us to proceed with the derivation of the rigid-body equations of motion.

1.3 DYNAMIC ANALYSIS

Translational Motion

Newton's second law, applied to translational motion, relates force to rate of change of linear momentum. In this application the forces involved are the propulsion and aerodynamic forces and the gravitational attraction. The propulsion system produces reaction forces on the aircraft body, while the aerodynamic effects depend on the shape of the aircraft body and the motion of the air relative to the body. Accordingly, when applying Newton's second law, it is appropriate to express the force and velocity vectors in an orthogonal aircraft-body coordinate (ABC) frame. The ABC frame is centered at the aircraft cg and has a fixed alignment relative to some convenient reference lines in the aircraft body.

Let \mathbf{F}_B represent the vector sum of the aerodynamic and propulsion forces, with the subscript B indicating that they are expressed in the body frame. The body-frame gravitational attraction is $Bm\mathbf{g}$, where m is the aircraft mass and B is the matrix that rotates the \mathbf{g} vector into the ABC frame. Also let \mathbf{v}_{abs} be the absolute (i.e., measured with respect to inertial space) velocity vector of the aircraft cg, expressed in the body frame. Then Newton's law for translational motion takes the form

$$\mathbf{F}_B + Bm\mathbf{g} = \frac{d}{dt_I}(m\mathbf{v}_{\text{abs}}). \quad (1.3-1)$$

The derivative must be taken with respect to an inertial reference frame, and this is indicated by the subscript I . The gravitational attraction will be assumed to be due only to the mass of the Earth.

We must now choose an inertial reference frame for use in the development of the aircraft equations of motion. Any frame that is nonrotating and nonaccelerating (but may be moving with constant velocity) relative to the average position of the "fixed" stars may be used. A coordinate frame with its origin at the center of the sun, and with a constant orientation relative to the fixed stars, is an almost perfect inertial reference frame. A reference frame that is less perfect, but much more convenient in terms of visualizing the trajectory of the vehicle, specifying its coordinates, and specifying Earth's gravity, is an orthogonal Earth-centered inertial (ECI) reference frame. This is a coordinate frame with its origin at the center of the Earth, translating with the Earth, but with a fixed orientation relative to the stars.

Figure 1.3-1 shows the ECI frame with origin O , the ABC frame with origin P , and an intermediate north-east-down (NED) frame on the surface

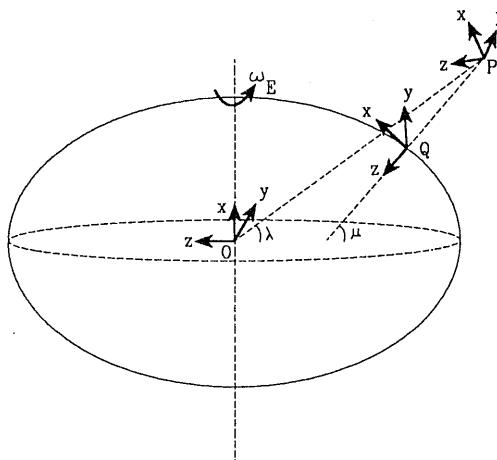


Fig. 1.3-1 Coordinate frames for kinematic analysis.

of the Earth, with origin Q . The NED frame moves with the aircraft and is vertically below the aircraft cg, so that its $x-y$ plane is tangent to the Earth's surface at Q . This frame will be required later, when a reduced set of equations of motion is constructed (the flat-Earth equations). The shape of the Earth is quite accurately ellipsoidal, but the eccentricity of the elliptical cross section has been exaggerated for the purposes of Fig. 1.3-1. The origin of the ECI frame is defined to be at the center of the ellipse, as shown in the figure. The shape of the Earth, and the angles of latitude (λ and μ) and longitude (l), will be defined more precisely in Section 1.4 when we consider navigation.

The orientation of the ABC and NED frames changes with time relative to the (fixed) ECI frame. The orientation of the ECI frame can be chosen to suit a given simulation problem. We have chosen to align the x -axis of the ECI frame along the Earth's spin axis, pointing toward the North Pole. For terrestrial navigation it is convenient to specify the direction of the ECI y - and z -axes by fixing their orientation relative to the NED frame at $t = 0$ (the initial time in the equations of motion). A suitable choice is to assume that the ECI y - and z -axes would be aligned with those of the NED frame, if the latter were at zero latitude and longitude at $t = 0$.

In the ECI frame the position of the aircraft cg will be denoted by the *inertial position vector* $\mathbf{p}(t)$. The rotation matrix that takes vectors from the ECI frame to the ABC frame is $B(t)$. Since the coordinate frames are orthogonal and the transformation is a pure rotation, the B matrix is an orthogonal matrix.

As stated earlier, the aerodynamic forces on an aircraft are created by its motion relative to the air mass. Therefore, it is convenient to split the absolute velocity vector, \mathbf{v}_{abs} , into a component of velocity relative to the air mass and a component that is the absolute velocity of the air mass. It will be assumed that the entire air mass rotates with the angular velocity vector of the Earth (i.e., no slippage and shearing). Local winds will be introduced into the equations later. Table 1.3-1 summarizes the definitions and the simple vector relationships needed to apply Newton's laws to the aircraft motion.

The state variables of the aircraft model will be the three components of each of the inertial position vector \mathbf{p} , the relative velocity vector \mathbf{v}_B , and the angular velocity vector $\boldsymbol{\omega}_B$. A differential equation will be needed for the time-varying transformation matrix B , and this will lead to the three additional states for attitude.

One set of state equations follows immediately by writing $\dot{\mathbf{p}}$ in terms of its components

$$\dot{\mathbf{p}} = B^T \mathbf{v}_{\text{abs}} = B^T \mathbf{v}_B + \boldsymbol{\omega}_E \times \mathbf{p}. \quad (1.3-2)$$

This same relationship, expressed in body coordinates, will provide an

TABLE 1.3-1. Variables Used in the Equations of Motion

In ECI Coordinates

Position vector of aircraft cg:	\mathbf{p}
Absolute velocity of aircraft cg:	$\dot{\mathbf{p}}$
Absolute angular velocity of Earth's rotation:	$\boldsymbol{\omega}_E$
Absolute velocity of surrounding air at P :	$\boldsymbol{\omega}_E \times \mathbf{p}$
Rotation matrix from ECI to ABC coordinates:	$B(t)$
Earth's gravitational acceleration:	\mathbf{g}

In ABC Coordinates

Absolute angular velocity of ABC frame:	$\boldsymbol{\omega}_B$
Absolute velocity of surrounding air:	$B(\boldsymbol{\omega}_E \times \mathbf{p})$
Relative velocity of aircraft cg with respect to air mass:	\mathbf{v}_B
Absolute velocity of aircraft cg:	\mathbf{v}_{abs}
Net applied force and net applied torque:	$\mathbf{F}_B, \mathbf{T}_B$

expression for \mathbf{v}_{abs} for use in Newton's law:

$$\mathbf{v}_{abs} = \mathbf{v}_B + B(\boldsymbol{\omega}_E \times \mathbf{p}). \quad (1.3-3)$$

Substituting this into the Newton's law equation, (1.3-1), gives

$$\mathbf{F}_B + Bm\mathbf{g} = \dot{m}\mathbf{v}_{abs} + m \frac{d}{dt_i} [\mathbf{v}_B + B(\boldsymbol{\omega}_E \times \mathbf{p})]. \quad (1.3-4)$$

In the case of missiles the \dot{m} term may be quite significant, but for aircraft it is negligible. It is sufficient to consider the dynamics of an aircraft at a number of fixed, specified, overall weights. The \dot{m} term will be dropped from the equations at this point in the interest of compactness.

Now performing the differentiation indicated in (1.3-4), using (1.2-37), (1.2-34), and (1.2-40), the result is

$$\frac{1}{m} \mathbf{F}_B + B\mathbf{g} = (\dot{\mathbf{v}}_B + \boldsymbol{\omega}_B \times \mathbf{v}_B) + B(\boldsymbol{\omega}_E \times \dot{\mathbf{p}}). \quad (1.3-5)$$

In order to leave only one set of derivatives in these equations it is necessary to substitute for $\dot{\mathbf{p}}$, using the state equations already obtained. The resulting term can be simplified by using the distributive property of the cross-product, the rotation matrix distributive property with cross-products, and the orthogonality of the rotation matrix. Then, rearranging the equations to put the

derivatives on the left-hand side, we get (Problem 1.3-3) the state equation

$$\dot{\mathbf{v}}_B = \frac{1}{m} \mathbf{F}_B - (\boldsymbol{\omega}_B + B\boldsymbol{\omega}_E) \times \mathbf{v}_B + B[\mathbf{g} - \boldsymbol{\omega}_E \times (\boldsymbol{\omega}_E \times \mathbf{p})]. \quad (1.3-6)$$

In (1.3-6) the translational velocity \mathbf{v}_B is a relative velocity but the angular velocity $\boldsymbol{\omega}_B$ is absolute. Relative angular velocity can be introduced into the equation by defining

$$\boldsymbol{\omega}_R \equiv \boldsymbol{\omega}_B - B\boldsymbol{\omega}_E, \quad (1.3-7)$$

where $\boldsymbol{\omega}_R$ is the angular velocity of the body frame relative to the Earth, expressed in body coordinates. Equation (1.3-6) then becomes

$$\dot{\mathbf{v}}_B = \frac{1}{m} \mathbf{F}_B - (\boldsymbol{\omega}_R + 2B\boldsymbol{\omega}_E) \times \mathbf{v}_B + B[\mathbf{g} - \boldsymbol{\omega}_E \times (\boldsymbol{\omega}_E \times \mathbf{p})]. \quad (1.3-8)$$

This equation is now consistent but is not necessarily more useful than (1.3-6).

Strictly, the relative angular velocity $\boldsymbol{\omega}_R$ should be used to calculate aerodynamic effects due to angular rates. This is not really necessary because aerodynamic moments are significant only at angular rates much greater than $\|B\boldsymbol{\omega}_E\|$. Furthermore, the commonly used rate sensors (rate gyros) measure absolute rates (i.e., the components of $\boldsymbol{\omega}_B$), and absolute rates occur in the state equations for the angular motion (derived in the next section). Therefore, we shall continue to use (1.3-6), and later calculate aerodynamic moments from $\boldsymbol{\omega}_B$. In contrast to the angular velocity, the absolute linear velocity of the Earth's atmosphere can be very large (e.g., at the Equator), and at the same time the relative velocity \mathbf{v}_B (that generates the aerodynamic forces) may be quite small. External (in the airflow) sensors are used to measure quantities relative to the airflow.

The only thing that now remains to be done with the state equations (1.3-6) is to relate the gravitational acceleration to the state variables, and this will be done in the next subsection.

The Gravitational Attraction Force

The mass-attraction force between two bodies is directly proportional to the product of their masses and inversely proportional to the square of the distance (r) between their centers of mass. When this inverse square law is applied to the Earth's gravitational attraction, it is convenient to combine the constant of proportionality with the mass of the Earth. The result is a constant known as the *Earth-mass gravitational constant*, GM , and the magnitude of the force acting on an aerospace vehicle of mass m is then $(GM \times m)/r^2$. The direction of this force is, to a close approximation,

aligned along the ECI position vector, pointing toward the center of the Earth. The length of the ECI position vector yields the separation r in this inverse-square-law relationship. From the facts above, it follows that the gravitational acceleration can be written as

$$\mathbf{g} = -\frac{\text{GM}}{\|\mathbf{p}\|^3} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}. \quad (1.3-9)$$

The component $(\text{GM}/\|\mathbf{p}\|^2)$ provides the correct magnitude for \mathbf{g} , and the remaining denominator component, $\|\mathbf{p}\|$, serves to reduce the \mathbf{p} vector to unit length. The Earth-mass gravitational constant has the value $\text{GM} = 1.4076431 \times 10^{16} \text{ ft}^3/\text{s}^2$ (excluding the mass of the Earth's atmosphere; see [WGS-84, 1987]).

The gravitational attraction varies with the length of \mathbf{p} because of the inverse square law, and also with the latitude angle of \mathbf{p} because of the nonspherical shape of the Earth. If the vehicle is constrained to move on or near the surface of the Earth, it experiences both of these effects (because of the variation in the Earth's radius). The deviation of the \mathbf{g} vector from the local normal (to the Earth's surface) varies smoothly, from zero at the Equator and the poles to a maximum of about 6 minutes of arc at ± 45 degrees latitude. The variation in $\|\mathbf{g}\|$ at sea level is quite significant, being from 32.199 ft/s^2 (to five digits) at the Equator to 32.257 ft/s^2 at the poles. Therefore, it is useful to have a gravitational model that will apply to high-altitude hypersonic flight around the Earth, or yield an accurate value of $\|\mathbf{g}\|$ at sea level for a given latitude. The latter value may then be used for simulated flight over a small area of the Earth's surface at the given latitude.

Equation (1.3-9) is easily modified [Britting, 1971; Miller and Leskiw, 1987] to account for the effect of the Earth's oblateness on \mathbf{g} . The components p_x , p_y , and p_z of the ECI position vector in (1.3-9) are, respectively, replaced by \bar{p}_x , \bar{p}_y , and \bar{p}_z , from

$$\begin{aligned} \bar{p}_x &= p_x \left[1 + 1.5J_2 \left(\frac{r_E}{\|\mathbf{p}\|} \right)^2 (3 - 5 \sin^2 \lambda) \right] \\ \bar{p}_y &= p_y \left[1 + 1.5J_2 \left(\frac{r_E}{\|\mathbf{p}\|} \right)^2 (1 - 5 \sin^2 \lambda) \right] \\ \bar{p}_z &= p_z \left[1 + 1.5J_2 \left(\frac{r_E}{\|\mathbf{p}\|} \right)^2 (1 - 5 \sin^2 \lambda) \right], \end{aligned} \quad (1.3-10)$$

where J_2 ($J_2 = 1.08263 \times 10^{-3}$) is a gravitational harmonic constant, r_E is the Earth's equatorial radius ($r_E = 2.092565 \times 10^7 \text{ ft}$), and λ is a latitude angle calculated from $\sin^{-1}(p_x/\|\mathbf{p}\|)$ (λ is actually the *geocentric latitude*; see Section 1.4). These constants (in metric form) are given in [WGS-84, 1987].

In Section 1.4 formulae are given for the angle λ in terms of the usual geographic latitude used on maps and for the radius of the Earth as a function of λ . The sea-level magnitude of the gravitational acceleration may then be calculated for any geographic latitude. Alternatively, (1.3-9) and (1.3-10) may be used directly to calculate $\mathbf{g}(\mathbf{p})$, within the equations of motion. Equations (1.3-10) assume that the ECI x -axis is aligned with the polar axis; hence the x -component of these equations is the one that differs from the other two components (which are identical because of symmetry around the polar axis).

Higher-order corrections for the Earth's oblateness use additional harmonic constants and other correction terms in (1.3-10). Minor anomalies (deviations from the theoretical value of \mathbf{g}) occur due to mountain ranges and variations in the density of the Earth's crust. Highly accurate models of the gravitational field are required for inertial guidance of long-range missiles and satellite navigation problems, and a description of the U.S. World Geodetic System [WGS-84, 1987] contains relevant information. More information on the shape of the Earth and its gravitational field can be found in Wells [Wells, 1967] and the Chapter 14 bibliography therein, although these references are pre-WGS-72 and contain obsolete values of the various constants. Additional references are given in Section 1.4.

In the state equation (1.3-6), the last term $(\mathbf{g} - \boldsymbol{\omega}_E \times \boldsymbol{\omega}_E \times \mathbf{p})$ contains the Earth's gravitational acceleration and the centripetal acceleration due to rotating in the ECI frame with the Earth's angular velocity vector. It is interesting to examine this term because it also corresponds to the acceleration sensed by a plumb bob or pendulum suspended above the Earth. The Earth angular rate vector and the ECI position are of the form

$$\boldsymbol{\omega}_E = \begin{bmatrix} \omega_x \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (1.3-11)$$

where ω_x is the Earth's rotation rate (7.292115×10^{-5} rad/s). Therefore, the vector triple product reduces to

$$\boldsymbol{\omega}_E \times (\boldsymbol{\omega}_E \times \mathbf{p}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix} \mathbf{p} = -\omega_x^2 \begin{bmatrix} 0 \\ p_y \\ p_z \end{bmatrix}. \quad (1.3-12)$$

This vector lies in a plane parallel to the Earth's equatorial plane, and if the inertial position vector is moving with the Earth (i.e., a geostationary trajectory), it corresponds to the centripetal acceleration for a circular orbit at a perpendicular distance of $\sqrt{p_y^2 + p_z^2}$ from the Earth's axis.

It is an observed fact that at a point on the surface of the Earth, the vector

$$\mathbf{g}' \equiv \mathbf{g} - \boldsymbol{\omega}_E \times (\boldsymbol{\omega}_E \times \mathbf{p}) \quad (1.3-13)$$

is accurately normal to the surface (actually, to the surface of the “reference ellipsoid” described in Section 1.4). A plumb bob or pendulum at the Earth’s surface aligns itself with this vector. The \mathbf{g}' vector is commonly called the “gravity” vector. It is slightly smaller in magnitude than \mathbf{g} and varies from 32.088 ft/s² at the Equator to 32.257 ft/s² (the same as \mathbf{g}) at the poles. It is important to distinguish between \mathbf{g} and \mathbf{g}' when using (1.3-6).

Accelerometer Measurements

In Chapter 4 and later chapters we shall need to make use of a linear accelerometer as a feedback control sensor. This is an appropriate point at which to consider what quantity this sensor actually measures. A linear accelerometer contains a mass that is constrained to move along a “sensitive axis” and is restrained by a spring. An electrical transducer is used to measure the deviation of the mass from a neutral position and produce a corresponding output signal. The accelerometer measures absolute (inertial) accelerations and resolves them in body-fixed axes (assuming that it is fastened to the rigid vehicle). The steady-state output is proportional to the component, along the sensitive axis, of the *difference between the acceleration of the case and the gravitational acceleration*. An accelerometer at a point p on a rigid body therefore measures a component of the acceleration \mathbf{a}'_p , given by

$$\mathbf{a}'_p = \mathbf{a}_p - \mathbf{g}, \quad (1.3-14a)$$

where \mathbf{a}_p and \mathbf{g} are, respectively, the absolute (inertial) acceleration of the point and the gravitational acceleration at this point. The equation must be resolved into body axes.

If the accelerometer is exactly at the cg of an aircraft (an unlikely situation), and aligned with one of the body axes, then substituting the quantities defined for the rigid aircraft into (1.3-14a), its output is proportional to a component of the vector

$$\mathbf{a}'_p = \frac{\mathbf{F}_B + Bm\mathbf{g}}{m} - B\mathbf{g} = \frac{\mathbf{F}_B}{m}. \quad (1.3-14b)$$

The force \mathbf{F}_B is the vector sum of the aerodynamic and thrust forces expressed in body axes, as used in our equations of motion. When the aerodynamic and propulsion forces are zero (e.g., outside the atmosphere) the vehicle is in “free fall” and the accelerometer reads zero. If the accelerometer is aligned along the aircraft z -axis and is almost vertical, then

by multiplying the numerator and denominator of (1.3-14b) by g_0 (the local magnitude of \mathbf{g} at sea level), we see that the accelerometer reads 1 g-unit when the lifting force is equal to the weight of the aircraft. When the aircraft is following a horizontal flight path it will usually have a nose-up attitude (to produce the required angle of attack), so the accelerometer reading will not be exactly equal to 1 g (but will normally be quite close).

Next, we shall determine what is measured if the accelerometer is at a point p , with position vector \mathbf{r} relative to the cg, and the aircraft has a nonzero angular velocity vector $\boldsymbol{\omega}$ (with respect to the inertial reference frame). The instantaneous velocity of the point p , in inertial space, is given by

$$\mathbf{v}_p = \mathbf{v}_{cg} + \boldsymbol{\omega} \times \mathbf{r}.$$

The inertial acceleration is obtained by differentiating this equation with respect to the inertial frame; thus

$$\mathbf{a}_p = \frac{d(\mathbf{v}_{cg})}{dt_I} + \frac{d(\boldsymbol{\omega} \times \mathbf{r})}{dt_I},$$

and this equation must be expressed in body axes to determine the accelerometer output. The first term on the right is simply $(\mathbf{F}_B + Bmg)/m$, and in the second term we shall use the theorem of Coriolis to differentiate $\boldsymbol{\omega} \times \mathbf{r}$. The vector \mathbf{r} is constant in body axes, and $\boldsymbol{\omega}$ is $\boldsymbol{\omega}_B$ in the previously defined notation. Therefore, we obtain

$$\mathbf{a}_p = \frac{\mathbf{F}_B + Bmg}{m} + \dot{\boldsymbol{\omega}}_B \times \mathbf{r} + \boldsymbol{\omega}_B \times (\boldsymbol{\omega}_B \times \mathbf{r}).$$

Now, substituting the accelerometer equation (1.3-14a), expressed in body axes, for the left-hand side of this equation, we obtain

$$\mathbf{a}'_p = \frac{\mathbf{F}_B}{m} + \dot{\boldsymbol{\omega}}_B \times \mathbf{r} + \boldsymbol{\omega}_B \times (\boldsymbol{\omega}_B \times \mathbf{r}). \quad (1.3-14c)$$

Equation (1.3-14c) can be built into our airplane models to simulate the measurements of an accelerometer. It contains a "centripetal" component $\boldsymbol{\omega}_B \times (\boldsymbol{\omega}_B \times \mathbf{r})$ and a "tangential" component $\dot{\boldsymbol{\omega}}_B \times \mathbf{r}$. If, for example, the accelerometer is aligned perpendicularly to the longitudinal axis of the aircraft, and the aircraft is undergoing a pitching motion, the centripetal term does not contribute to the accelerometer output; only the component $\dot{\boldsymbol{\omega}}_B \times \mathbf{r}$ is important. This fact will be used later when we measure "normal acceleration" and use it for feedback purposes.

Finally, we should note that an accelerometer has bias, nonlinearities, internal noise, and internal dynamics that limit its speed of response, and

these effects are not included in (1.3-14a). They become important when very small accelerations are to be measured or when the measured acceleration is changing rapidly. If necessary, the bandwidth limitations can be modeled by one of the electrical circuit models described in Chapter 3.

Summary of the Translational Equations

We now have two translational-motion state equations, (1.3-2) and (1.3-6). These two sets of state equations are coupled and nonlinear. The $\dot{\mathbf{p}}$ equation contains \mathbf{v}_B , and the $\dot{\mathbf{v}}_B$ equation contains \mathbf{p} , hence the coupling that prevents each set from being solved in isolation. The nonlinearity of the equations is evident in the vector products of angular and translational velocities, and in products with the time-varying B matrix.

The first state equation, (1.3-2), simply equates velocity to rate of change of position. The three inertial position coordinates are required as state variables because of the rotation of the air mass with respect to the inertial frame, and because the variation of the gravitational attraction can be expressed in terms of these position coordinates.

It is important to consider the significance of (1.3-2). In round numbers, aerodynamically controlled vehicles are limited to altitudes below 100,000 ft, which is a small distance compared with the Earth's radius of about 21 million feet. Therefore, the magnitude of the inertial position vector, \mathbf{p} , will be roughly constant and equal to the Earth's radius. Consequently, (1.3-2) can be discarded in many aircraft and missile simulations. The \mathbf{p} vector couples into (1.3-6) and this effect can be maintained by replacing the \mathbf{g} vector and centripetal acceleration $[\boldsymbol{\omega}_E \times (\boldsymbol{\omega}_E \times \mathbf{p})]$ by the \mathbf{g}' vector [Equation (1.3-13)]. The magnitude of \mathbf{g}' at sea level and latitude 45°, $g'_0 = 32.17$ ft/s², is then commonly used for the magnitude of the vector, and it is normal to the surface of the Earth (i.e., along the z-axis of the NED frame). This leads naturally to the so-called "flat-Earth" equations of motion, using the NED frame as if it were an inertial reference frame. Position is then tracked by resolving the \mathbf{v}_B velocity vector into the NED frame and integrating these components to obtain three NED position states.

As a matter of general interest, consider the components of (1.3-2). The tangential velocity component $\boldsymbol{\omega}_E \times \mathbf{p}$ will vary in magnitude from about 1500 ft/s above the Equator to zero over the poles. Therefore, if the relative velocity of the vehicle with respect to the atmosphere (\mathbf{v}_B) is zero (i.e., sitting on the ground), the vehicle may have a velocity of up to 1500 ft/s with respect to inertial space. It is this fact that makes it desirable to launch space rockets from near the Equator in an easterly direction. Conversely, (1.3-2) shows that remaining stationary in inertial space may correspond to an atmospheric speed exceeding Mach 1.0 (1117 ft/s at sea level in the standard atmosphere).

In the second set of state equations, (1.3-6), the term $(\boldsymbol{\omega}_B + B\boldsymbol{\omega}_E) \times \mathbf{v}_B$ is a tangential acceleration component resulting from the combination of the

total angular velocity and the translational velocity. The $B\omega_E$ component of the total angular velocity is usually negligible compared to the body-axes angular rates of an aircraft, and its cross-product with \mathbf{v}_B is also negligible for normal speeds. Consequently, the flat-Earth equation obtained from (1.3-6) is written simply as

$$\dot{\mathbf{v}}_B = \frac{1}{m} \mathbf{F}_B - \boldsymbol{\omega}_B \times \mathbf{v}_B + B \mathbf{g}'_0, \quad (1.3-15)$$

where

$$\mathbf{g}'_0 = \begin{bmatrix} 0 \\ 0 \\ g'_0 \end{bmatrix}, \quad g'_0 = 32.17 \text{ ft/s}^2.$$

This equation is adequate for the purposes of analysis and design in the following chapters.

The as-yet unknown variables remaining in the state equations are \mathbf{F}_B , B , and $\boldsymbol{\omega}_B$. The applied force \mathbf{F}_B is created by the propulsion system and the aerodynamic effects and will be considered in Chapter 2. We now consider the problem of deriving additional state equations involving the remaining variables, B and $\boldsymbol{\omega}_B$. Here we need to recall the initial description of the characteristics of a state model and to realize that we have no states that specify rotational kinetic energy. This omission will be rectified by the dynamic analysis of rotational motion in the next section. Following this we shall perform a kinematic analysis to find angular relationships that involve the B matrix.

Angular Motion

The angular accelerations of the vehicle can be determined by applying Newton's second law to the rate of change of angular momentum of the aircraft. Thus if \mathbf{H}_B is the angular momentum vector of the rigid vehicle, and \mathbf{T}_B is the net torque acting about the aircraft cg, then

$$\mathbf{T}_B = \frac{d}{dt_I} (\mathbf{H}_B). \quad (1.3-16)$$

The torque is generated by the aerodynamic control surfaces, by any reaction-control thrusters, and by any components of the engine thrust not acting through the cg (e.g., due to thrust vectoring control). As before, the subscript I indicates that the rate of change is with respect to the inertial reference frame; the subscript B indicates that we shall again be working in body coordinates, and the ABC frame is again rotating with the angular velocity vector $\boldsymbol{\omega}_B$.

In order to determine the angular momentum vector, consider an element of mass, δm , with position vector \mathbf{r} (in ABC coordinates). Its translational momentum is the infinitesimal quantity

$$\delta(\text{momentum}) = (\boldsymbol{\omega}_B \times \mathbf{r}) \delta m.$$

The "angular momentum" of this particle is the moment of the translational momentum about the origin, or

$$\delta \mathbf{H}_B = [\mathbf{r} \times (\boldsymbol{\omega}_B \times \mathbf{r})] \delta m.$$

If $\boldsymbol{\omega}_B$ and \mathbf{r} have components

$$\boldsymbol{\omega}_B = P\mathbf{i} + Q\mathbf{j} + R\mathbf{k}$$

$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k},$$

and the vector triple product formula is applied to this equation, then

$$\delta \mathbf{H}_B = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} (x^2 + y^2 + z^2) \delta m - \begin{bmatrix} x \\ y \\ z \end{bmatrix} (Px + Qy + Rz) \delta m.$$

The angular momentum, \mathbf{H}_B , of the vehicle is obtained (in theory) by integrating the components of $\delta \mathbf{H}_B$ over the entire mass,

$$\mathbf{H}_B = \begin{bmatrix} P \int (y^2 + z^2) dm - Q \int xy dm - R \int xz dm \\ Q \int (x^2 + z^2) dm - R \int yz dm - P \int yx dm \\ R \int (x^2 + y^2) dm - P \int zx dm - Q \int zy dm \end{bmatrix}.$$

The various integrals in the angular momentum vector are defined to be the moments and cross-products of inertia, for example,

$$\text{moment of inertia about } x\text{-axis} = J_{xx} = \int (y^2 + z^2) dm$$

$$\text{cross-product of inertia } J_{xy} \equiv J_{yx} = \int xy dm.$$

On substituting these definitions into the angular momentum, we obtain expressions for the components of \mathbf{H}_B that are bilinear in P, Q, R , and the

inertias. This allows us to write \mathbf{H}_B as the vector matrix product

$$\mathbf{H}_B = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{yz} & J_{zz} \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \equiv J\boldsymbol{\omega}_B, \quad (1.3-17)$$

and J will be referred to as the *inertia matrix* of the rigid body.

With the angular momentum expressed in terms of the inertia matrix and angular velocity vector of the complete rigid body, we are ready to apply Newton's second law as expressed in (1.3-16). The differentiation is performed using the theorem of Coriolis [equation (1.2-37)], and the inertia matrix in (1.3-17) is assumed constant. In practice, the inertia matrix will change in an abrupt manner if the aircraft is releasing stores (dropping equipment) and will change in a gradual manner as fuel is used up. If the performance of the aircraft control systems is degraded by these effects, control system parameters must be changed in flight as a function of estimated mass properties. However, the most important effect is usually the variation of the aircraft cg, and this must be kept between predetermined limits by careful organization of the payload. If it is found necessary to consider the variation of mass properties, this would not be done by considering derivatives of the moments of inertia. Instead, at the control systems design stage, a number of "nominal" designs would be performed for configurations with different mass properties, and the flight control computer would be programmed to interpolate smoothly between these designs in flight. Finally, it is important to realize that had we applied Newton's second law to the angular momentum in the inertial frame rather than the body frame, then the inertia matrix would have been time varying and the equations much more difficult to handle.

We now proceed by differentiating the angular momentum given by (1.3-17) according to Newton's law (1.3-16). This gives

$$\dot{\mathbf{H}}_B + \boldsymbol{\omega}_B \times \mathbf{H}_B = \mathbf{T}_B;$$

therefore,

$$J\dot{\boldsymbol{\omega}}_B = -\boldsymbol{\omega}_B \times (J\boldsymbol{\omega}_B) + \mathbf{T}_B$$

or

$$\dot{\boldsymbol{\omega}}_B = -J^{-1}(\boldsymbol{\omega}_B \times (J\boldsymbol{\omega}_B)) + J^{-1}\mathbf{T}_B. \quad (1.3-18)$$

Equation (1.3-18) is the basic equation for the angular motion; it relates the derivatives of the body-axes angular rates P , Q , and R to the rates themselves and to the torques exerted on the airframe. It constitutes the next set of three state equations. The equations are nonlinear because the angular rate vector occurs twice in the first term on the right-hand side. We shall examine this nonlinearity in more detail as we progress.

The inverse of the inertia matrix occurs in (1.3-18), and because of symmetry this has a relatively simple form:

$$J^{-1} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_2 & k_4 & k_5 \\ k_3 & k_5 & k_6 \end{bmatrix}, \quad (1.3-19a)$$

where

$$\begin{aligned} k_1 &= \frac{J_{yy}J_{zz} - J_{yz}^2}{\Delta}, & k_2 &= \frac{J_{yz}J_{zx} + J_{xy}J_{zz}}{\Delta} \\ k_3 &= \frac{J_{xy}J_{yz} + J_{zx}J_{yy}}{\Delta}, & k_4 &= \frac{J_{zz}J_{xx} - J_{zx}^2}{\Delta} \\ k_5 &= \frac{J_{xy}J_{zx} + J_{yz}J_{xx}}{\Delta}, & k_6 &= \frac{J_{xx}J_{yy} - J_{xy}^2}{\Delta} \end{aligned}$$

and

$$\Delta = J_{xx}J_{yy}J_{zz} - 2J_{xy}J_{yz}J_{zx} - J_{xx}J_{yz}^2 - J_{yy}J_{zx}^2 - J_{zz}J_{xy}^2.$$

Usually, neither the full inertia matrix nor its inverse are required. First note that a set of aircraft axes can be chosen such that the cross-products of inertia are zero (i.e., the inertia matrix is diagonal). These axes are called the *principal axes*. However, this step is not necessary since for most aircraft the $x-z$ plane is a plane of symmetry. Under this condition, for every product $y_i z_j$ or $y_i x_j$ in an inertia computation there is a product that is identical in magnitude but opposite in sign. Therefore, only the J_{xz} cross-product of inertia is nonzero. A notable exception is an oblique-wing aircraft [Travassos et al., 1980], which does not have a plane of symmetry. Under the plane-of-symmetry assumption the inertia matrix and its inverse reduce to

$$J = \begin{bmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{bmatrix}, \quad J^{-1} = \frac{1}{\Gamma} \begin{bmatrix} J_z & 0 & J_{xz} \\ 0 & \frac{\Gamma}{J_y} & 0 \\ J_{xz} & 0 & J_x \end{bmatrix}, \quad (1.3-19b)$$

where $\Gamma = J_x J_z - J_{xz}^2$ and the double-subscript notation on the moments of inertia has been dropped.

With the simple inertia matrix given by (1.3-19b), Equation (1.3-18) can be expanded further. First, replace the $\omega_B \times$ operation in (1.3-18) by the P, Q, R cross-product matrix, and let the components of the applied torque

vector be

$$\mathbf{T}_B = \begin{bmatrix} \bar{L} \\ M \\ N \end{bmatrix} \quad (1.3-20)$$

(\bar{L} will be used for the rolling moment component of the torque vector, and L will be used for "lift"). Then the angular acceleration equations can be expanded to

$$\begin{aligned} \Gamma \dot{P} &= J_{xz} [J_x - J_y + J_z] PQ - [J_z(J_z - J_y) + J_{xz}^2] QR + J_z \bar{L} + J_{xz} N \\ J_y \dot{Q} &= (J_z - J_x) PR - J_{xz}(P^2 - R^2) + M \\ \Gamma \dot{R} &= [(J_x - J_y) J_x + J_{xz}^2] PQ - J_{xz}[J_x - J_y + J_z] QR + J_{xz} \bar{L} + J_x N. \end{aligned} \quad (1.3-21)$$

Equations (1.3-21) are nonlinear and highly coupled. For example, consider a solid rectangular body with three planes of symmetry, so that all of the cross-products of inertia are zero, and no driving torques. The moment equations then reduce to the *gyroscopic equations*:

$$\begin{aligned} \dot{P} &= \frac{(J_y - J_z) QR}{J_x} \\ \dot{Q} &= \frac{(J_z - J_x) PR}{J_y} \\ \dot{R} &= \frac{(J_x - J_y) PQ}{J_z}. \end{aligned} \quad (1.3-22)$$

These equations are symmetrical and are inherently coupled because angular rates about any two axes produce an acceleration about the third (a gyroscopic effect; see Problem 1.3-7). This *inertia coupling* has important consequences for aircraft maneuvering rapidly at high angles of attack; we examine its effects in Chapter 4.

In the analysis of the angular motion we have so far neglected the angular momentum of any spinning rotors. The effects may be quite significant; for example, a number of World War I aircraft had a single "rotary" engine that had a fixed crankshaft and rotating cylinders. The gyroscopic effects caused by the angular momentum of the engine gave these aircraft tricky handling characteristics. In the case of a small jet with a single turbofan engine on the longitudinal axis the effects are smaller but noticeable in maneuvering flight, as can be observed in a computer simulation. To represent the effect, we

shall add a constant vector to the angular momentum vector in (1.3-17). Thus

$$\mathbf{H}_B = J\boldsymbol{\omega}_B + \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix}. \quad (1.3-23)$$

If this analysis is carried through, the effect is to add the following terms, respectively, to the right-hand sides of the three equations (1.3-21):

$$\begin{aligned} & J_z(Rh_y - Qh_z) + J_{xz}(Qh_x - Ph_y) \\ & - Rh_x + Ph_z \\ & J_{xz}(Rh_y - Qh_z) + J_x(Qh_x - Ph_y). \end{aligned} \quad (1.3-24)$$

This completes the application of Newton's laws to the aircraft motion. We are still lacking some state equations since there is no attitude information to incorporate in the B matrix. Therefore, we turn our attention to the kinematics (the study of the relative motion between parts) of the problem.

1.4 KINEMATIC ANALYSIS

Around-the-Earth Navigation

To perform the analysis we must first establish some facts about the shape of the Earth and the method of specifying coordinates in an ECI frame. A good approximation to the sea-level surface of the Earth is an ellipsoid of revolution [Vanicek and Krakiwsky, 1982; *Encyclopaedia Britannica*, 1987; Kuebler and Sommers, 1981], the solid figure generated by rotating the ellipse

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \quad b > a > 0 \quad (1.4-1)$$

about its x -axis, when the x -axis is coincident with the Earth's axis of rotation. Figure 1.4-1 shows a cross section through the Earth in a plane containing the axis of rotation; this plane figure is the generating ellipse. The great circle passing through point Q has a radius r_s (the sea-level radius), and the equatorial radius is $b = r_E$ ($r_E = 2.092565 \times 10^7$ ft). The length of the semiminor axis is given by

$$a = b(1 - f), \quad (1.4-2)$$

where $f = 1/298.257$ is the *Earth flattening parameter* [WGS-84, 1987].

The commonly used position coordinates on the Earth's surface are the latitude and longitude angles. Longitude is measured in degrees east and

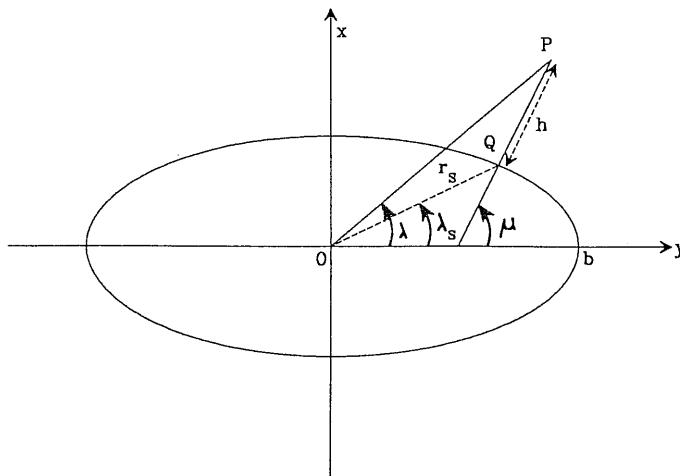


Fig. 1.4-1 Variables used for around-the-earth navigation.

west of the prime meridian, which is defined to pass through the site of the Royal Observatory in Greenwich, England. For the purposes of navigation in the nonrotating ECI frame, we shall use the *celestial longitude angle*, λ , defined as an angle measured in the ECI y - z plane (Earth equatorial plane), from the negative z -axis, and in a counterclockwise direction when viewed from the North Pole. Therefore, because of the Earth's rotation, a geostationary observer would have a continuously increasing celestial longitude angle.

Measurement of latitude is slightly more complex because of the oblateness of the Earth. In Fig. 1.4-1 the *geocentric latitude* is defined as the angle λ , and the angle μ is the *geodetic latitude*. Geodetic latitude is the latitude commonly used on geographical maps and charts. For the point Q at sea level, with geocentric latitude λ_s , it is easy to show (Problem 1.4-1) by using the geometry of the figure and the slope of the generating ellipse at Q that

$$(1-f)^2 \tan \mu = \tan \lambda_s. \quad (1.4-3)$$

Point Q has the polar coordinates (r_s, λ_s) and using the equation of the ellipse (1.4-1), an expression for the sea-level radius can readily be derived (Problem 1.4-1); thus

$$r_s^2 = \frac{r_E^2}{1 + [1/(1-f)^2 - 1]\sin^2 \lambda_s}. \quad (1.4-4)$$

Figure 1.4-1 also shows the aircraft inertial position vector OP , and defines the height, h , in terms of the normal to the Earth's surface at Q . This

is an appropriate point at which to note how we may keep track of inertial position as a state trajectory evolves. We already have the equation for the derivatives $\dot{\mathbf{p}}(t)$ in terms of \mathbf{p} , namely (1.3-2). It is convenient to use $\mathbf{p}(t)$, rather than (h, l, λ) , in the state vector. However, (h, l, λ) are convenient quantities for visualization of the trajectory, and we shall now examine their relationship to \mathbf{p} .

A convenient assumption is that a trajectory starts with zero celestial longitude (i.e., above the ECI negative z -axis) and with specified altitude, $h(0)$, and geodetic latitude, $\mu(0)$. Then using the geometry of Fig. 1.4-1 over the z -axis, the inertial position vector can be initialized as (see also Problem 1.4-2)

$$\mathbf{p}(0) = \begin{bmatrix} p_x(0) \\ p_y(0) \\ p_z(0) \end{bmatrix} = \begin{bmatrix} r_s \sin \lambda_s(0) + h(0) \sin \mu(0) \\ 0 \\ -[r_s \cos \lambda_s(0) + h(0) \cos \mu(0)] \end{bmatrix}. \quad (1.4-5)$$

The quantities $\lambda_s(0)$ and r_s are calculated from $\mu(0)$ using (1.4-3) and (1.4-4), respectively.

Next, consider the problem of calculating (h, l, λ) from \mathbf{p} . The celestial longitude and geocentric latitude can readily be calculated from the components of \mathbf{p} , using

$$\begin{aligned} \tan l &= \frac{p_y}{-p_z} \\ \tan \lambda &= \frac{p_x}{\sqrt{p_y^2 + p_z^2}}. \end{aligned} \quad (1.4-6)$$

A four-quadrant inverse tangent function should be used to calculate l since it covers the range 0° to 360° . The geographic longitude can then be calculated from

$$l_G = \text{principal value}(l_G(0) + l - \omega_x t), \quad (1.4-7)$$

where ω_x is the Earth's angular rate, and "principal value" indicates a conversion of the argument to an angle in the range -180° to $+180^\circ$. The height h is difficult to calculate because of the oblateness. Since we are not using h for navigational purposes, it can be approximated with adequate accuracy, by

$$h \approx \|\mathbf{p}\| - r_s. \quad (1.4-8)$$

Attitude (Orientation) Representation

Returning to the kinematic analysis, the right-handed coordinate frames that will be used for the analysis were shown in Fig. 1.3-1. The ECI frame is

centered in the Earth at O , with its y and z axes lying in the equatorial plane and its x -axis coincident with the Earth's spin axis. A locally level (tangential to the Earth's surface) north–east–down frame is shown at Q . This frame is directly underneath P , the position of the aircraft cg, and QP is the height of the aircraft above the surface of the Earth. Notice that the oblateness of the Earth has been exaggerated to show that the height vector is not coincident with the inertial position vector OP . The aircraft-body coordinate (ABC) frame is shown with its origin at P . Its x , y , z axes are assumed to be aligned, respectively, forward, starboard, and down in the aircraft.

We shall assume that the NED and ABC frames reach their particular orientations through the following sequence of rotations.

ECI to NED Frame: A vector in ECI coordinates can be expressed in terms of NED coordinates by performing the rotations (refer to Fig. 1.3-1)

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{NED}} = B_\mu B_l \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{ECI}}.$$

A rotation B_l through the celestial longitude angle (about the ECI x -axis) is performed first. The x -component is left unchanged and this is a right-handed rotation; therefore, B_l is as shown below. The rotation B_μ is performed second; it leaves the y -component unchanged, and μ is in a left-handed sense. The result is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{NED}} = \begin{bmatrix} \cos \mu & 0 & \sin \mu \\ 0 & 1 & 0 \\ -\sin \mu & 0 & \cos \mu \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos l & \sin l \\ 0 & -\sin l & \cos l \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{ECI}}.$$

Let B_G denote the complete transformation from the inertial frame to the geographic frame; then B_G is given by

$$B_G = B_\mu B_l = \begin{bmatrix} \cos \mu & -\sin \mu \sin l & \sin \mu \cos l \\ 0 & \cos l & \sin l \\ -\sin \mu & -\cos \mu \sin l & \cos \mu \cos l \end{bmatrix}. \quad (1.4-9)$$

NED to ABC Frame: Remember that the ABC frame is aligned x –forward, y –starboard, and z –down in the aircraft. The sequence of rotations conventionally used in the aircraft industry to describe the instantaneous attitude with respect to a reference frame (in this case, NED) is as follows. Starting from the reference frame:

1. Rotate about the z -axis, nose right (positive “yaw” ψ).
2. Rotate about the new y -axis, nose up (positive “pitch” θ).
3. Rotate about the new x -axis, right wing down (positive “roll” ϕ).

Conversely, if going from the vehicle-body frame to the reference frame, the sequence roll, pitch, yaw must be followed. The yaw, pitch, and roll (or bank) angles ψ, θ, ϕ are commonly referred to as *Euler angles*. In terms of coordinate transformations we have

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ABC} = B_\phi B_\theta B_\psi \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{NED}.$$

The individual rotation matrices can be written down immediately with the help of Fig. 1.2-1 and the rules established in Section 1.2. Thus we get

$$\begin{aligned} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ABC} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \\ &\quad \times \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{NED}. \end{aligned}$$

Let B_B denote the complete transformation from the geographic frame to the body frame; then B_B is given by

$$B_B = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & \sin \phi \cos \theta \\ \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi & \cos \phi \cos \theta \end{bmatrix}, \quad (1.4-10)$$

and the overall transformation from ECI to ABC is denoted by the matrix $B(t)$, where

$$B = B_B B_G. \quad (1.4-11)$$

The B matrix describes the attitude of the aircraft as it changes with time. Its elements are functions of the two known angles μ and l (calculated from \mathbf{p}) and the as-yet-unknown Euler angles ϕ, θ , and ψ . The attitude of the aircraft at any instant is the doubly integrated effect of the angular accelerations that have occurred throughout the trajectory. The dynamics of the angular motion gave us a set of differential equations for the angular rate vector $\boldsymbol{\omega}_B$ in body axes. These body-axes rates produce a corresponding set of attitude angular rates, depending on the particular instantaneous attitude. This relationship is inherent in the strapdown equation derived in Section 1.2. The B matrix derived above satisfies the strapdown equation:

$$\dot{B} = -\Omega B, \quad (1.4-12)$$

where Ω is the cross-product matrix of the angular rates as measured with

respect to the axes of the rotating frame, that is,

$$\Omega = \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix}. \quad (1.4-13)$$

The strapdown equation (1.4-12) consists of nine coupled differential equations. It is therefore highly redundant for our purposes, since only three Euler angles are necessary to keep track of attitude as opposed to nine B -matrix elements. From the strapdown equation we can derive a set of three state equations, expressing Euler angle rates as functions of the Euler angles and the body-axes angular rates. Our rigid-body equations will then be complete, with differential equations for all the motion variables. The appropriate equations will now be derived.

Three-Variable Attitude Propagation

The cross-product relationship (1.2-39), from which the strapdown equation was derived, can be written in terms of the columns of the B matrix as

$$\dot{\mathbf{b}}_i = -\boldsymbol{\omega} \times \mathbf{b}_i, \quad i = 1, 2, 3 \quad (1.4-14)$$

This is a perfectly general relationship for one coordinate frame rotating with respect to another; the B matrix contains the relative Euler angles and the $\boldsymbol{\omega}$ vector contains the relative angular velocities. Equation (1.4-10) is also a general expression for the rotation matrix from one frame to the other in terms of the relative Euler angles. Using (1.4-10) and evaluating the equations (1.4-14) for the first and second elements of \mathbf{b}_3 and the first element of \mathbf{b}_2 leads to the following equations for the Euler angle derivatives (Problem 1.4-3):

$$\frac{d}{dt} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix}. \quad (1.4-15)$$

As a matter of general interest, note that the coefficient matrix above is not orthogonal. This equation will be written symbolically as

$$\dot{\Phi} = \mathcal{E}(\Phi)\boldsymbol{\omega}, \quad \text{where } \Phi = [\phi, \theta, \psi]^T. \quad (1.4-16)$$

Equations (1.4-15) represents the set of state equations that we were seeking as an alternative to (1.4-12). The angular rates P , Q , and R become the elements of the absolute angular velocity vector, $\boldsymbol{\omega}_B$, of the ABC frame.

The new Euler angles can be distinguished from the Euler angles ϕ, θ, ψ , used to represent the orientation of the ABC frame relative to NED, by writing them as ϕ_1, θ_1, ψ_1 . Then, if the relative attitude ϕ, θ, ψ is required, (1.4-11) gives

$$B_B(\phi, \theta, \psi) = B(\phi_1, \theta_1, \psi_1) B_G^T(\mu, l), \quad (1.4-17)$$

and the attitude angles can readily be found from elements in the first row and last column of B_B . Thus

$$\begin{aligned} \phi &= \text{ATAN2}(b_{23}, b_{33}) \\ \theta &= -\sin^{-1}(b_{13}) \\ \psi &= \text{ATAN2}(b_{12}, b_{11}), \end{aligned} \quad (1.4-18)$$

where b_{ij} is the element in the i th row and j th column of B_B , and ATAN2 is the FORTRAN four-quadrant inverse tangent function. Precautions must be taken against division by zero and against inverse trigonometric function arguments of magnitude greater than unity. Also, a convention for choosing the signs of the angles must be adopted. Recall that the NED frame is a locally level frame on the Earth's surface, so these Euler angles are useful for visualization and feedback control of attitude with respect to the Earth.

The state equations (1.4-16) have the advantages of using physically meaningful variables and of being minimal (i.e., this is the minimum number of first-order differential equations that can be used). They also have some significant disadvantages. First, inspection of the equations shows that a division by zero occurs if the pitch attitude variable (θ) reaches $\pm 90^\circ$. In digital simulation this exact number has almost zero probability of occurrence, but significant computational errors may occur in the vicinity of the singularity. Second, the Euler angles may integrate up to values outside the normal $\pm 90^\circ$ range of pitch, and the normal $\pm 180^\circ$ range of the bank and yaw angles. This "wraparound" problem may make it difficult to determine the attitude uniquely. Finally, note that the equations are linear in P, Q, R , but nonlinear in terms of the desired Euler angles. In the next subsection it is shown that a set of attitude variables can be found such that the resulting first-order differential equations are linear, and this presents some analytical benefits in certain applications.

Equations (1.4-16) are commonly used for aircraft simulation, but are not suitable for around-the-Earth flight, all-attitude flight, and simulation of spinning bodies. We therefore consider an alternative representation of attitude which requires four variables.

Four-Variable Attitude Propagation

There are a number of other ways to represent the orientation of a rotated coordinate frame besides the Euler angles of the preceding subsection

[Giardina et al., 1981]. These methods involve four, five, or even six variables in place of the three Euler angles. They have all been considered for the purpose of avoiding the mathematical singularity of the Euler angle representation, and maximizing the speed of computer processing in navigation calculations. However, one of these methods is now used almost exclusively in all of the various spacecraft, missile, and aircraft applications. This is the so-called *quaternion* four-variable representation.

In Section 1.2 we described Euler's theorem, which shows that any general rotation matrix is *similar* to a unique plane rotation matrix, corresponding to a rotation about the Euler axis. Following this idea, assume that the rotation matrix, B , is similar to the plane rotation matrix B_P . Then a rotation matrix T exists such that

$$B = T^T B_P T.$$

This relationship can be used to express B in terms of four variables. First, assume that the Euler axis is described by three angles of inclination α, β, γ to the reference axes x, y, z , respectively. Further, assume that the T rotation produces a new frame such that the B_P rotation takes place about the new x -axis, and that the new y -axis remains in the old $x-y$ plane. Let the B_P rotation be through an angle δ ; then the expression for B takes the form [Robinson, 1958]

$$B = \begin{bmatrix} \cos \alpha & d_{21} & d_{31} \\ \cos \beta & d_{22} & d_{32} \\ \cos \gamma & 0 & d_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \delta & \sin \delta \\ 0 & -\sin \delta & \cos \delta \end{bmatrix} \begin{bmatrix} \cos \alpha & \cos \beta & \cos \gamma \\ d_{21} & d_{22} & 0 \\ d_{31} & d_{32} & d_{33} \end{bmatrix}, \quad (1.4-19)$$

where the elements d_{ij} are to be determined. Note that because of orthogonality

$$\cos^2 \alpha + \cos^2 \beta + \cos^2 \gamma = 1.0. \quad (1.4-20)$$

The unknown elements d_{ij} can be found by solving the simultaneous equations that result from the orthogonality conditions imposed on T . Some sign ambiguities that occur in this analysis can be resolved by examining the conditions under which T must reduce to the identity matrix. When this is done, T is found to be

$$T = \begin{bmatrix} \cos \alpha & \cos \beta & \cos \gamma \\ -\frac{\cos \beta}{\sin \gamma} & \frac{\cos \alpha}{\sin \gamma} & 0 \\ -\frac{\cos \alpha}{\tan \gamma} & -\frac{\cos \beta}{\tan \gamma} & \sin \gamma \end{bmatrix}. \quad (1.4-21)$$

The B matrix that results from multiplying out (1.4-19) is a complicated function of α, β, γ , and δ . The substitutions

$$\begin{aligned} q_0 &= \cos \frac{\delta}{2} \\ q_1 &= \cos \alpha \sin \frac{\delta}{2} \\ q_2 &= \cos \beta \sin \frac{\delta}{2} \\ q_3 &= \cos \gamma \sin \frac{\delta}{2} \end{aligned} \quad (1.4-22)$$

produce the following four-variable representation of B :

$$B = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \quad (1.4-23)$$

The orthogonality conditions on T and the substitutions (1.4-22) lead to

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1.0. \quad (1.4-24)$$

The q_i variables, usually referred to as the *quaternion parameters* are also attributed to Euler. They are related to the Euler angles by [Shoemake, 1985]

$$\begin{aligned} q_0 &= \pm (\cos(\phi/2) \cos(\theta/2) \cos(\psi/2) + \sin(\phi/2) \sin(\theta/2) \sin(\psi/2)) \\ q_1 &= \pm (\sin(\phi/2) \cos(\theta/2) \cos(\psi/2) - \cos(\phi/2) \sin(\theta/2) \sin(\psi/2)) \\ q_2 &= \pm (\cos(\phi/2) \sin(\theta/2) \cos(\psi/2) + \sin(\phi/2) \cos(\theta/2) \sin(\psi/2)) \\ q_3 &= \pm (\cos(\phi/2) \cos(\theta/2) \sin(\psi/2) - \sin(\phi/2) \sin(\theta/2) \cos(\psi/2)). \end{aligned} \quad (1.4-25)$$

Whichever sign is chosen in (1.4-25), the same choice must be used for all; (1.4-23) is obviously unaffected by the choice of sign.

A kinematic relationship, analogous to the strapdown equation and to (1.4-15), can be found to link the derivatives of the quaternion parameters to the angular rates measured about the reference axes. Unlike the nonlinear relationships involving the Euler angles, the quaternion parameters satisfy a set of linear differential equations, with the angular rates as their coefficients. The differential equations can be used to "propagate" the quaternion representation of attitude forward in time.

The quaternion representation of attitude is widely used in space and missile applications [Robinson, 1958; Mortensen, 1974; Ickes, 1970]; it gives a unique representation of attitude and does not suffer from the singularity and wraparound problems of the Euler equations. Also, the linearity of the quaternion differential equations leads to some computational advantages. Note that because of (1.4-24), none of the quaternion elements can exceed unit magnitude.

Equations (1.4-25) allow us to calculate the initial values of the quaternion parameters (to initialize our quaternion differential equations) given the initial values of the Euler angles. When the initial Euler angles are not directly available (e.g., if multiple Euler rotations are involved, or vector analysis led directly to an initial B matrix) it is convenient to initialize the quaternion from the B matrix. This can be achieved by using (1.4-23). If terms on the main diagonal of (1.4-23) are added together, the following relationships are obtained:

$$\begin{aligned} 4q_0^2 &= 1 + b_{11} + b_{22} + b_{33} \\ 4q_1^2 &= 1 + b_{11} - b_{22} - b_{33} \\ 4q_2^2 &= 1 - b_{11} + b_{22} - b_{33} \\ 4q_3^2 &= 1 - b_{11} - b_{22} + b_{33}. \end{aligned} \quad (1.4-26a)$$

These relationships give the magnitudes of the quaternion elements but not the signs. The off-diagonal terms in (1.4-23) yield the additional relationships

$$\begin{aligned} 4q_0 q_1 &= b_{23} - b_{32} & 4q_1 q_2 &= b_{12} + b_{21} \\ 4q_0 q_2 &= b_{31} - b_{13} & 4q_2 q_3 &= b_{23} + b_{32} \\ 4q_0 q_3 &= b_{12} - b_{21} & 4q_1 q_3 &= b_{13} + b_{31}. \end{aligned} \quad (1.4-26b)$$

From the first set of equations, (1.4-26a), the quaternion element with the largest magnitude (at least one of the four must be nonzero) can be selected. The sign associated with the square root can be chosen arbitrarily, and then this variable can be used as a divisor with (1.4-26b) to find the remaining quaternion elements. An interesting quirk of this algorithm is that the quaternion may change sign if the algorithm is restarted with a new set of initial conditions. This will have no effect on the rotation matrix given in (1.4-23). Algorithms like this are discussed in Shoemake [1985] and Shepperd [1978].

We have now seen how the four quaternion parameters can be initialized from the Euler angles and how they can be used to compute the nine elements of the B matrix. Note also that the Euler angles can be recovered from the quaternion parameters via the B -matrix elements and (1.4-18). In the next subsection we derive the linear differential equations that allow the quaternion to be propagated forward in time.

The Quaternion Differential Equations

A classic textbook on analytical dynamics is the book by Whittaker [1937]. He attributes the four-parameter representation of attitude to Euler and provides the starting point that we shall use to derive differential equations for the quaternion parameters.

There are a number of different ways in which a set of Euler angles can be defined. Whittaker uses Euler's original definition and is able to derive quite simply the following expressions for the reference-axes angular rate components in terms of the quaternion parameters and their derivatives:

$$\begin{aligned} P &= 2(q_0\dot{q}_1 + q_3\dot{q}_2 - q_2\dot{q}_3 - q_1\dot{q}_0) \\ Q &= 2(-q_3\dot{q}_1 + q_0\dot{q}_2 + q_1\dot{q}_3 - q_2\dot{q}_0) \\ R &= 2(q_2\dot{q}_1 - q_1\dot{q}_2 + q_0\dot{q}_3 - q_3\dot{q}_0). \end{aligned} \quad (1.4-27)$$

These equations are derived by taking the equivalent of (1.4-16) inverted, and substituting for the Euler angle rates using the equivalent of (1.4-25).

Equations (1.4-27) are bilinear in terms of the q_i and their derivatives and can therefore be written as

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = 2 \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}.$$

If we now differentiate the constraint (1.4-24) and use it to augment these equations, we find that the 4×4 coefficient matrix is orthogonal; and when inverted, the equations are bilinear in the q_i and the angular rates. Then, by inspection, we can write the following differential equation (Problem 1.4-4) for the quaternion parameters:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} 0 & P & Q & R \\ -P & 0 & -R & Q \\ -Q & R & 0 & -P \\ -R & -Q & P & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = -\frac{1}{2} \Omega_q \mathbf{q}. \quad (1.4-28)$$

We shall refer to the vector \mathbf{q} as the quaternion; note that the 4×4 coefficient matrix, Ω_q , contains our original 3×3 cross-product matrix, Ω , as a submatrix. It is easy to check by direct multiplication that because of its special structure, Ω_q has the idempotent property

$$\Omega_q^2 = -(P^2 + Q^2 + R^2)I = -\|\omega_B\|^2 I, \quad (1.4-29)$$

so the even and odd powers of Ω_q are given, respectively, by

$$\begin{aligned}\Omega_q^{2n} &= -\|\omega_B\|^{2n} I \\ \Omega_q^{2n+1} &= -\|\omega_B\|^{2n} \Omega_q.\end{aligned}\tag{1.4-30}$$

Equation (1.4-28) is the set of differential equations that we were seeking; they can be included with our aircraft state equations in place of (1.4-15), and they eliminate the problem of the singularity at $\theta = 90^\circ$. They are coupled to the rest of the state equations through the state variables P , Q , and R that occur in the Ω_q matrix, and through reconstructing the B matrix for use in the other state equations. Euler angles need only be calculated when needed for feedback control and for visualization of the attitude. They are also needed at the beginning of a simulation to initialize the quaternion. In a simulation the quaternion differential equations may be numerically integrated along with the rest of the state equations. In a strapdown inertial navigation system a different approach must be used. The angular rates P , Q , and R are then no longer simulation variables but come from the rate-gyro hardware and must be sampled, quantized, and digitized for computer processing. Therefore, a discrete-time version of the quaternion differential equation (1.4-28) is required; we consider this problem as an illustrative example in Chapter 3.

Nonminimality

The state equations with quaternion attitude propagation are not “minimal.” We previously anticipated three state variables (ϕ, θ, ψ) to describe attitude; instead, we have the four quaternion variables. In later chapters we examine and seek to control the *natural modes* of our state model. The natural modes are components of the time response, and the number of modes is directly related to the number of state variables. A nonminimal model will introduce extra modes which are simply artifacts of the method of computation and are not readily associated with real aircraft behavior. The result is some inconvenience when we come to perform control system design with the aircraft model.

1.5 ASSEMBLING THE STATE EQUATIONS

In this section we assemble the complete set of rigid-body equations in preparation for use in aircraft models, derive the flat-Earth equations, and incorporate local wind into the flat-Earth set.

The Round-Earth Equations

For the complete state model the relevant state equations are (1.3-2), (1.3-6), (1.3-18), and (1.4-28). These equations are now assembled in matrix form with the vector cross-product operations ($\omega_B \times$) and ($\omega_E \times$) replaced by the matrices Ω_B and Ω_E , respectively. This is the way in which the equations will be programmed for computer purposes. The equations are

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}}_B \\ \dot{\boldsymbol{\omega}}_B \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \Omega_E & B^T & 0 & 0 \\ B\mathbf{g}(\mathbf{p}) - B\Omega_E^2 & -(\Omega_B + B\Omega_E) & 0 & 0 \\ 0 & 0 & -J^{-1}\Omega_B J & 0 \\ 0 & 0 & 0 & -\frac{1}{2}\Omega_q \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{v}_B \\ \boldsymbol{\omega}_B \\ \mathbf{q} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\mathbf{F}_B}{m} \\ J^{-1}\mathbf{T}_B \\ 0 \end{bmatrix}. \quad (1.5-1)$$

The state vector contains 13 elements: three position coordinates, three translational velocity coordinates, three angular velocity coordinates, and four quaternion variables. The coefficient matrix contains the submatrices

$$\Omega_E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{bmatrix}, \quad \Omega_B = \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix},$$

$$\Omega_q = \begin{bmatrix} 0 & P & Q & R \\ -P & 0 & -R & Q \\ -Q & R & 0 & -P \\ -R & -Q & P & 0 \end{bmatrix}, \quad (1.5-2)$$

where ω_x is the Earth's rotation rate, and P , Q , and R are the body-axes angular rates of the aircraft.

Although the equations have been written in a linear form to emphasize the state vector, they are nonlinear because the coefficient matrix contains B , Ω_B , and Ω_q , which are functions of the state variables. Also, the gravitational acceleration is a function of the inertial position vector according to (1.3-9). The forces and moments on the right-hand side are not constants but functions of the states \mathbf{v}_B and $\boldsymbol{\omega}_B$; we shall derive the appropriate relationships in Chapter 2.

These state equations are valid for simulation of high-speed flight around the rotating Earth. For spacecraft trajectories more detailed consideration

would need to be given to the gravity model and other mass attraction effects and to disturbance forces such as the solar wind. For studies of maneuvering flight over small regions of the Earth's surface and design of the relevant control system, the $\dot{\mathbf{v}}_B$ equation is overcomplicated. In the next subsection this equation is replaced by the flat-Earth equation (1.3-15).

The Flat-Earth Equations

For these reduced equations we take the NED frame on the surface of the Earth as an inertial reference frame. This frame is both accelerating and rotating; however, the accelerations associated with the Earth's rate are negligible compared to the accelerations that can be produced by a maneuvering aircraft. Furthermore, when designing control systems for maneuvering an aircraft, there is no need to perform accurate navigation over the surface of the Earth. Therefore, as explained in Section 1.3, the $\dot{\mathbf{v}}_B$ equation can be replaced by (1.3-15), and the $\dot{\mathbf{p}}$ equation can be replaced by a simpler equation, as shown below.

Velocity in the NED frame is calculated using the rotation matrix (1.4-10); thus

$$\dot{\mathbf{p}}_{\text{NED}} \equiv \mathbf{v}_{\text{NED}} = B_B^T \mathbf{v}_B, \quad (1.5-3)$$

and \mathbf{p}_{NED} is defined to be the position vector relative to the NED geographic frame. We now have distance-down, distance-north, and distance-east states for tracking maneuvers and performing approximate navigation for flight simulation. Note that since we are now using only the ABC and NED frames, we can replace the B rotation matrix by the B_B matrix as we reduce the round-Earth equations.

In the flat-Earth model it will also be acceptable, in general, to drop the quaternion representation of attitude and use the Euler angle equation, (1.4-16). The flat-Earth equations [obtained from (1.3-15), (1.3-18), (1.4-16), and (1.5-3)] are

$$\begin{aligned} \dot{\mathbf{v}}_B &= -\Omega_B \mathbf{v}_B + B_B \mathbf{g}'_0 + \frac{\mathbf{F}_B}{m} && \text{(force equation)} \\ \dot{\boldsymbol{\omega}}_B &= -J^{-1} \Omega_B J \boldsymbol{\omega}_B + J^{-1} \mathbf{T}_B && \text{(moment equation)} \quad (1.5-4) \\ \dot{\boldsymbol{\Phi}} &= \mathcal{E}(\boldsymbol{\Phi}) \boldsymbol{\omega}_B && \text{(attitude equation)} \\ \dot{\mathbf{p}}_{\text{NED}} &= B_B^T \mathbf{v}_B && \text{(navigation equation)}, \end{aligned}$$

and the state vector is now

$$\mathbf{x}^T = [\mathbf{v}_B^T, \boldsymbol{\omega}_B^T, \boldsymbol{\Phi}^T, \mathbf{p}_{\text{NED}}^T]. \quad (1.5-5)$$

The reader should examine the nonlinearity and coupling of these equations. The moment equation is driven by the applied torques, it is nonlinear through Ω_B and ω_B occurring together in one term, and it couples into the other equations through P , Q , and R . The Attitude (or kinematic equation) is driven by ω_B from the moment equation, and is nonlinear through $\mathcal{E}(\Phi)$ and its product with ω_B . The force equation needs ω_B and Φ from the other equations for its Ω_B and B_B matrix coefficients. The NED position equation no longer couples back into the other equations, because gravity is assumed constant and Earth rotation is neglected. The expressions to be derived in Chapter 2 for the forces and torques will create additional coupling in the flat-Earth equations.

Inclusion of Wind

The aerodynamic forces on an aircraft are created by its motion relative to the surrounding air. The air itself is in motion relative to the NED “inertial” reference frame. The NED frame is locally level (i.e., x - y plane tangential to the Earth reference ellipsoid at the chosen location), and we assume that the local wind has north, east, and down velocity components W_N , W_E , and W_D , respectively. We shall also assume that the wind velocity vector is constant over a region much larger than the size of the aircraft. This means that we do not have to consider wind shearing effects and torques exerted on the aircraft; for a more advanced treatment the reader is referred to Etkin [1972].

The velocity of the aircraft cg with respect to the air is given by

$$\mathbf{v}_R = \mathbf{v}_B - B_B \begin{bmatrix} W_N \\ W_E \\ W_D \end{bmatrix}. \quad (1.5-6)$$

When this equation is added to the flat-Earth equations, the wind components must be supplied as inputs. Then \mathbf{v}_R , rather than \mathbf{v}_B , must be used in the calculation of aerodynamic forces and moments.

1.6 SUMMARY

This chapter provides sufficient background material to enable the reader to deal with many of the dynamical problems that occur in the modern aerospace industry. Thus the derived equations of motion may be applied to vehicles intended to reach hypersonic speeds and perhaps go into orbit, or to slowly moving aircraft near the surface of the Earth. The design of these vehicles requires large computer simulations involving the equations of motion, and controlling them may require programming onboard computers with algo-

rithms that employ many of the concepts described here. A given control algorithm may employ many coordinate rotations and make use of the principles of inertial navigation. It may be necessary to provide "tight" control of the trajectory of a vehicle in order to meet thermal constraints as well as to minimize fuel costs and limit structural loads. The trajectories involved may well require an understanding of motion around the rotating Earth with accurate modeling of the Earth's gravitational attraction.

REFERENCES

- Becker, R. A., *Introduction to Theoretical Mechanics*. New York: McGraw-Hill, 1954.
- Britting, K. R., *Inertial Navigation Systems Analysis*. New York: Wiley Interscience, 1971.
- Encyclopaedia Britannica*, 15th ed., Vol. 17, *Macropaedia*. Chicago: Encyclopaedia Britannica, 1987, pp. 530-539.
- Etkin, B., *Dynamics of Atmospheric Flight*. New York: Wiley, 1972.
- Giardina, C. R., J. Heckathorn, and D. Krasnjanski, "A comparative study of strapdown algorithms," *J. Inst. Navig.*, vol. 28, no. 2, 1981.
- Goldstein, H., *Classical Mechanics*, 2nd ed., Reading, Mass.: Addison-Wesley, 1981.
- Greenwood, D. T., *Principles of Dynamics*. Englewood Cliffs, N.J.: Prentice Hall, 1965.
- Ickes, B. P., "A new method for performing digital control system attitude computations using quaternions," *AIAA J.*, vol. 8, no. 1, Jan. 1970, pp. 13-17.
- Kalman, R. E., "Mathematical description of linear dynamical systems," *J.S.I.A.M. Control Ser.*, ser. A, vol. 1, no. 2, 1963, pp. 152-192.
- Kuebler, W., and S. Sommers, "The Role of the Earth's shape in navigation: An example," *J. Inst. Navig.*, vol. 28, no. 1, Spring 1981.
- McRuer, D., I. Ashkenas, and D. Graham, *Aircraft Dynamics and Automatic Control*. Princeton, N.J.: Princeton University Press, 1973.
- Meirovitch, L., *Methods of Analytical Dynamics*. New York: McGraw-Hill, 1970.
- Miller, K. S., and D. M. Leskiw, *An Introduction to Kalman Filtering with Applications*. Malabar, Fla.: Krieger Publishing Co., 1987.
- Mortensen, R. E., "Strapdown guidance error analysis," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-10, no. 4, July 1974, pp. 451-457.
- Perkins, C. D., and R. E. Hage, *Airplane Performance, Stability, and Control*. New York: Wiley, 1949.
- Pio, R. L., "Symbolic representation of coordinate transformations," *IEEE Trans. Aerosp. Navig. Electron.*, vol. 11, no. 2, June 1964, pp. 128-134.
- Pio, R. L., "Euler angle transformations," *IEEE Trans. Autom. Control*, vol. AC-11, no. 4, Oct. 1966, pp. 707-715.
- Robinson, A. C., "On the use of quaternions in simulation of rigid-body motion," *WADC Tech. Rep. 58-17*, Wright Air Development Center, Dayton, Ohio, Dec. 1958.
- Shepperd, S. W., "Quaternion from rotation matrix," *AIAA J. Guid. Control*, vol. 1, no. 3, May-June 1978, pp. 223-224.

- Shoemake, K., "Animating rotation with quaternion curves," *Computer Graphics*, vol. 19, no. 3, 1985, pp. 245-254.
- Strang, G., *Linear Algebra and Its Applications*. New York: Academic Press, 1980.
- Travassos, R. H., N. K. Gupta, K. W. Iliffe, and R. Maine, "Determination of an oblique wing aircraft's aerodynamic characteristics," paper 80-1630, *AIAA Flight Mech. Conf.*, 1980, p. 608.
- Vanicek, P., and E. J. Krakiwsky, *Geodesy: The Concepts*. Amsterdam: North-Holland, 1982.
- Wells, D., *Theory and Problems of Lagrangian Dynamics*, Schaum's Outline Series. New York: McGraw-Hill, 1967.
- WGS-84, "Department of Defense World Geodetic System 1984; its definition and relationships with local geodetic systems," *DMA TR-8350.2*, Defense Mapping Agency, U.S. Naval Observatory, Washington, D.C., 1987.
- Whittaker, E. T., *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*, 4th ed. Cambridge: Cambridge University Press, 1937.
- Wilkinson, J. H., and G. Golub, "Ill conditioned eigensystems and the computation of the Jordan canonical form," *SIAM Rev.*, vol. 18, Oct. 1976, pp. 578-619.

PROBLEMS

Section 1.2

- 1.2-1 Derive the cross-product matrix used in Equation (1.2-6).
- 1.2-2 Prove the distributive property of the rotation matrix with the cross-product operation, as given in (1.2-15).
- 1.2-3 Derive the plane rotation matrix given in (1.2-16).
- 1.2-4 Derive the theorem of Coriolis, (1.2-37), by considering infinitesimal changes in length and direction of a rotating vector in a time interval dt .

Section 1.3

- 1.3-1 The ECI position coordinates of a celestial object are (x, y, z) . Determine the ENU (east, north, up) position coordinates of the object with respect to a tracking station on the surface of the Earth at celestial longitude ℓ , geodetic latitude λ , and sea-level altitude. Assume a spherical Earth (of radius R), align the ECI frame with its z -axis pointing up toward the North Pole, and assume that the ENU frame is obtained by rotating first through the longitude angle and then through the latitude angle. Assume also that longitude is measured east from the ECI x -axis. Show that the Earth's radius appears in only one of the required coordinates.
- 1.3-2 Show that the rotation matrix between two coordinate frames can be calculated from a knowledge of the position vectors of two different

objects if the position vectors are known in each frame. Specify the rotation matrix in terms of the solution of a matrix equation. Show how this technique could be used to determine vehicle attitude by taking telescope bearings on two known stars, given a star catalog.

- 1.3-3 Fill in all of the steps in the derivation of the state equation (1.3-6).
- 1.3-4 Write and test a program to evaluate g_0 and g'_0 (values at the Earth's surface) as functions of geodetic latitude.
- 1.3-5 Determine the body-axes force equations [equivalent to (1.3-6)] for a vehicle following a trajectory around the Earth outside the atmosphere (use absolute velocity v_{abs} instead of relative velocity v_b). Write a complete set of vector state equations for such a vehicle using the quaternion representation of attitude, and indicate how the state vector can be initialized.
- 1.3-6 Derive the conditions for a body to remain in a geostationary orbit of the Earth. Use the gravity model and geodetic data to determine the geostationary altitude. What are the constraints on the latitude and inclination of the orbit?
- 1.3-7 Derive a set of linear state equations from (1.3-22) by considering perturbations from a steady-state condition with angular rates P_e , Q_e , and R_e . Find expressions for the eigenvalues of the coefficient matrix when only one angular rate is nonzero, and show that there is an unstable eigenvalue if the moment of inertia about this axis is either the largest or the smallest of the three inertias. Deduce any practical consequences of this result.

Section 1.4

- 1.4-1 Derive (1.4-3) and (1.4-4).
- 1.4-2 Obtain the equation, corresponding to (1.4-5), to initialize the inertial position vector when the initial longitude is nonzero.
- 1.4-3 Fill in all the steps in the derivation of the Euler angle rate equation, (1.4-15).
- 1.4-4 Derive the quaternion ODE, (1.4-28), from (1.4-27).

CHAPTER 2

BUILDING THE AIRCRAFT MODEL

2.1 INTRODUCTION

Model building is a fundamental process. An aircraft designer has a mental model of the type of aircraft that is needed, uses physical models to gather wind-tunnel data, and designs with mathematical models that incorporate the experimental data. The modeling process is often iterative; a mathematical model based on the laws of physics will suggest what experimental data should be taken, and the model may then undergo considerable adjustment in order to fit the data. In building the mathematical model we recognize the onset of the law of diminishing returns and build a model that is good enough for our purposes but has known limitations. Some of the limitations of the models involve uncertainty in the values of their parameters. Later, we attempt to characterize this uncertainty mathematically and allow for it in control system design.

Because of the high cost of building and flight testing a real aircraft, the importance of aircraft mathematical models goes far beyond control system design. The mathematical model is used, in conjunction with computer simulation, to evaluate the performance of the prototype aircraft and hence improve the design. It can also be used to drive training simulators, to reconstruct the flight conditions involved in accidents, and to study the effects of modifications to the design. Furthermore, other mathematical models are used in all aspects of the aircraft design (e.g., structural models for studying stress distribution and predicting fatigue life).

All of the succeeding chapters will make use of the mathematical models of this chapter in some form, and thus demonstrate the importance of

modeling in the design of aircraft control systems. The rigid-body equations of motion that we derived in Chapter 1 form the skeleton of the aircraft model. In this chapter we add some muscles to the skeleton by modeling the aerodynamic forces and moments that drive the equations. By the end of the chapter we will have the capability, given the basic aerodynamic data, to build mathematical models that can be used for computer simulation or for control systems design. We start by considering some basic elements of aerodynamics.

In the aerospace industry it is necessary for a wide range of specialists to work together; thus flight control engineers must be able to work with the aerodynamicists as well as with structural and propulsion engineers. Each must have some understanding of the terms and mathematical models used by the other. This is becoming increasingly important as designers seek to widen aircraft performance envelopes by integrating the many parts of the whole design process. Furthermore, at the prototype stage the controls designer must work closely with the test pilots to make the final adjustments to the control systems. This may take many hours of simulator time and flight test, and the flight controls engineer involved in the final stages of the design is also likely to be a qualified pilot.

2.2 BASIC AERODYNAMICS

Airfoil Section Aerodynamics

The mathematical model used by the control engineer will usually contain aerodynamic data for the aircraft as a whole. However, to gain the necessary insight, we start by examining the aerodynamic forces on an airfoil. Figure 2.2-1 shows the cross section of an airfoil (i.e., a body shaped to produce lift with minimal drag) and defines some of the terms used. The flow field around

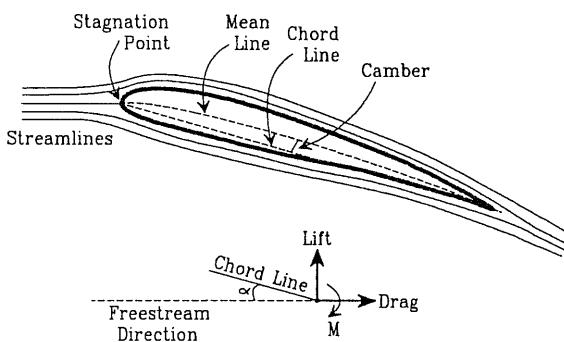


Fig. 2.2-1 Airflow around an airfoil and definition of terms.

the airfoil is represented by the *streamlines* shown in the figure (the flow direction at any point is tangential to the streamline passing through that point). The figure illustrates attached flow, that is, the streamlines follow the surface of the airfoil and do not reverse direction anywhere over the surface. This is a two-dimensional situation; the cross section is uniform and the span (length) of this airfoil is assumed to be infinite, so that the flow field does not change in the direction perpendicular to the plane of the diagram. The initial direction of the flow field is defined by the free-stream velocity vector. This is the velocity measured ahead of the airfoil at a sufficient distance that the flow there is unaffected by the presence of the airfoil.

The shape of the airfoil determines its aerodynamic properties, and some of the important geometrical parameters are shown in the figure. The *chord line* is a straight line drawn from the leading edge to the trailing edge, and is the reference line for describing the shape. An airfoil may be symmetric or, more usually, asymmetric with respect to the chord line. The *mean line* (or *camber line*) is a line drawn from leading edge to trailing edge midway between the two surfaces, and the difference between the mean line and the chord line shows the amount of camber. The shape of the upper and lower surfaces, the amount of camber, the thickness, and the leading-edge radius combine to determine the aerodynamic properties and the useful speed range.

Now imagine an experiment in which the airfoil is pivoted about an axis perpendicular to the cross section, passing through the chord line at an arbitrary distance back from the leading edge. The angle that the chord line makes with the free-stream velocity vector is the airfoil *angle of attack*, usually denoted by α (hereinafter referred to as "alpha") and shown as a positive quantity in the figure. In our hypothetical experiment, with a given airfoil at a fixed free-stream airspeed and density, the aerodynamic forces and moments are determined solely by alpha. Elementary mechanics tells us that in this situation, the aerodynamic effects can be represented by a force normal (because of symmetry) to the axis and a couple acting around the axis.

The aerodynamic force is conventionally resolved into two perpendicular components, the *lift* and *drag* components, shown in the figure. Lift is defined to be perpendicular to the free-stream velocity vector, and the drag component is parallel to it. Lift and drag normally increase as alpha is increased. The moment of the couple is indicated in the figure, and the positive reference direction is shown there. By definition, the moment of the couple is zero when the axis is chosen to pass through the center of pressure (cp) of the airfoil (i.e., the cp is the point through which the total force can be thought to be acting). This is not a particularly convenient location for the axis since experiments show that the location of the cp changes with alpha. There is another special location for the axis—the *aerodynamic center* (ac) of the airfoil. The ac is a point at which the aerodynamic "pitching" moment tends to be invariant with respect to alpha (within limits). It is normally about one quarter-chord back from the leading edge and may be slightly above or

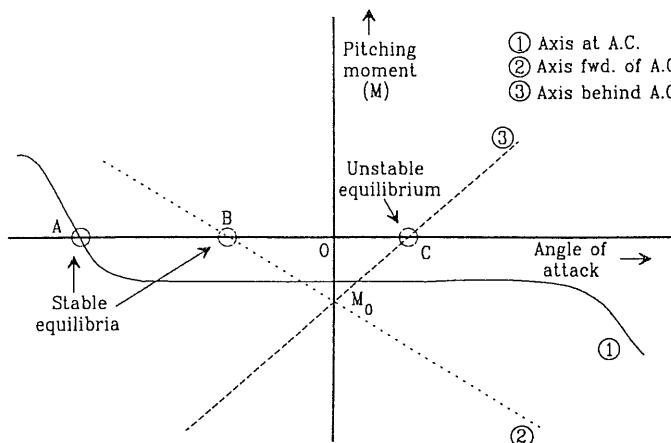


Fig. 2.2-2 Airfoil moment about different axes.

below the chord line; its position tends to move back toward the cp at high speeds. The ac is ahead of the cp, so the pitching moment measured at the ac will be a nose-down (negative) moment that is approximately constant with alpha.

The aerodynamic center is important when we come to consider the stability of the airfoil in an airflow. It is obvious (by reductio ad absurdum) that if we move the pivot axis forward of the ac, we will measure a negative pitching moment that becomes more negative as alpha is increased. This is shown in curve 2 of Fig. 2.2-2; point B on this curve is the angle of attack where the pitching moment becomes zero. If we attempt to increase alpha away from point B, a negative pitching moment is generated; and conversely, decreasing alpha generates a positive moment. These are restoring moments that tend to hold alpha at the value B. Therefore, neglecting any moment due to its weight, the airfoil will settle into a *stable equilibrium* condition at point B when allowed to pivot freely.

When the axis is at the aerodynamic center, as in curve 1 of the figure, there is a stable equilibrium at point A. This point is at a large negative value of alpha outside the normal range of operation. When we place the pivot axis behind the ac, as in curve 3, the pitching moment increases with alpha. There is an equilibrium condition at point C, but this is an *unstable equilibrium* because any small perturbation in alpha creates a moment that drives the angle of attack out of this region of operation. With the sign conventions we have chosen, we see that a stable equilibrium is associated with a negative slope to the pitching moment curve and unstable equilibrium with a positive slope. If the airfoil had to support the weight of an aircraft, a stable equilibrium point would have to occur at a positive angle of attack.

This would require curve 2 to be shifted upwards (i.e., M_0 positive); in practice, the horizontal tail of the aircraft provides the additional positive pitching moment.

The stability of this hypothetical experiment has been analyzed by considering the *static* balance of the pitching moment and the effect of small perturbations. We shall refer to the condition of a steady-state force tending to restore equilibrium as *positive stiffness*, and note that it is associated with a *negative slope* of the moment versus angle curve. In this experiment only a single degree of freedom was involved: rotation around a fixed axis. The static analysis was sufficient to determine whether the equilibrium was stable or unstable (and to determine the stability boundary), but not sufficient to determine the dynamics of the motion when the equilibrium is disturbed.

The motion of an airplane in the vertical plane involves three degrees of freedom, one rotational and two translational. An analysis of the stability of the motion requires that a steady-state trajectory be defined and an analysis of small perturbations in the motion be performed. From this analysis the *dynamic modes* (i.e., the time-dependent behavior of the system in response to an impulsive input) can be determined. A pilot's ability to control an airplane is linked to the stability of the modes, so *dynamic stability* is of critical importance. Dynamic stability analyses will be performed in later chapters. Here we note that positive stiffness is neither necessary nor sufficient to ensure dynamic stability, but the aircraft dynamic stability conditions will later be seen to be dominated by the static stability condition.

We must now formally define the expressions for the forces and moments generated by the airfoil. The positive directions for lift, drag, pitching moment, and alpha have been defined in Fig. 2.2-1, and the force and moment measurements are typically made at some point in the airfoil close to the ac (usually at the quarter-chord point). The force components and the moment of the couple are modeled by the following equations, involving lift, drag, and moment coefficients C_l , C_d , and C_m , respectively:

$$\begin{aligned} \text{lift per unit span} &= \bar{q}cC_l(\alpha) \\ \text{drag per unit span} &= \bar{q}cC_d(\alpha) \\ \text{pitching moment per unit span} &= \bar{q}c^2C_m(\alpha) \end{aligned} \quad (2.2-1)$$

dynamic pressure, $\bar{q} \equiv \frac{1}{2}\rho V_T^2$ (pressure units),

where ρ is the air density (0.002377 slug/ft³ at sea level), V_T is the free-stream airspeed, and c is the chord. The dynamic pressure equation models the way in which the forces and moments depend on airspeed and density. The reference length for the span is the chord length c , and the product $\bar{q}c$ has the dimensions of force per unit length.

The dimensionless *lift coefficient* C_l measures the effectiveness of the airfoil at producing lift. For a given airfoil, in a flow of given airspeed and

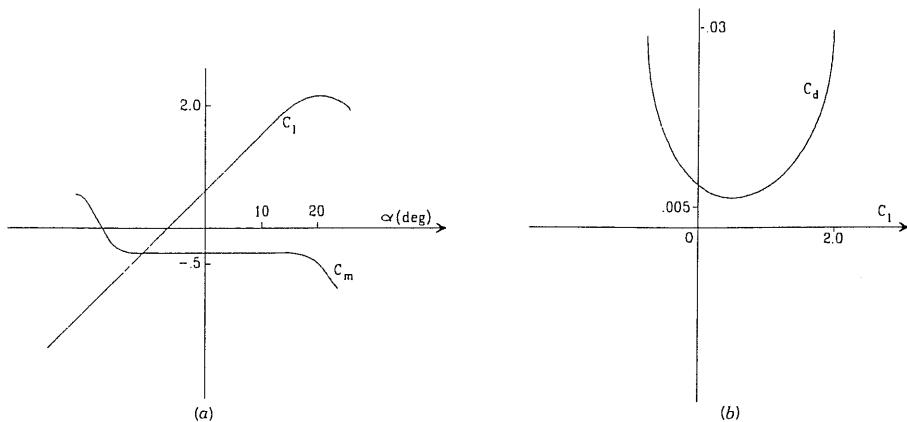


Fig. 2.2-3a, b Typical plots of lift, drag, and moment coefficients.

density, the lift coefficient can be shown to be a function of angle of attack only. This function is linear at low values of alpha and positive at zero angle of attack for cambered airfoils. The drag equation has the same form as the lift equation, and the *drag coefficient* C_d is also dependent only on alpha, although it is commonly presented as a function of lift coefficient. Typical plots of lift and drag coefficients, with representative values, are shown, respectively, in Fig. 2.2-3a and b. The moment equation is different from the lift and drag equations in that it requires an additional length variable to make it dimensionally correct. The airfoil chord c is used once again for this purpose. A typical plot of the *pitching moment coefficient* C_m is also shown in Fig. 2.2-3a.

It is instructive to consider the effects of large values of alpha, and a wide range of airspeeds, on the aerodynamic coefficients. The curvature of the airfoil causes a nonuniform distribution of pressure and velocity over its surface. Thus the airflow slows as it approaches the leading-edge *stagnation point* (see Fig. 2.2-1) and then accelerates around the upper and lower surfaces. The pressure is correspondingly higher in regions of low velocity, and lower in regions of high velocity. If the force normal to the chord line is computed by integrating this pressure distribution over the upper surface, and this is repeated for the lower surface, the net force normal to the chord has components that are the lift and drag forces on the airfoil. The airfoil pressure distribution changes significantly at high values of alpha and at high airspeeds, and this produces corresponding changes in the forces and moments. We shall consider first the effect of large alpha.

As shown in Fig. 2.2-3, for a fixed subsonic free-stream airspeed, the lift coefficient is a linear function of alpha, the drag coefficient is an approxi-

mately parabolic function of lift, and the pitching-moment coefficient is approximately constant, until some critical value of alpha is reached. Wind-tunnel flow-visualization studies show that beyond this value of alpha, the flow can no longer follow the upper surface of the airfoil. There is a region above the upper surface, near the trailing edge, where the velocity is low and the flow reverses direction in places, in a vortex motion. As the angle of attack is increased further, the beginning of the region of separated flow moves toward the leading edge of the airfoil. The pressure distribution over the airfoil is changed in such a way that the lift component of the aerodynamic force falls off rapidly and the drag component increases rapidly. The airfoil is said to be *stalled*, and this condition is normally avoided in flight. The pitching moment (about our axis through the aerodynamic center) also changes rapidly, typically becoming more negative.

Now consider the effect of varying the speed of the free-stream airflow. At low speeds the dynamic pressure acting on the airfoil is small relative to the static atmospheric pressure. For example, at 300 mph (440 ft/s) at sea level, the dynamic pressure is

$$\bar{q} = 0.5 \times 0.002377 \times 440^2 = 230 \text{ lb/ft}^2 \text{ (psf).}$$

This dynamic pressure of about 1.6 lb/in² (psi) is to be compared with the static pressure of approximately 14.7 psi at sea level. As the airspeed is increased, the changes in dynamic pressure become more significant relative to the static pressure, and changes begin to occur in the airfoil aerodynamic coefficients. These are referred to as *compressibility effects*.

The compressibility changes are commonly expressed as a function of *Mach number*. The Mach number M , at some point in the airflow, is the local airspeed divided by the speed of sound in air for the free-stream conditions (1117 ft/s at sea level in the standard atmosphere [Yuan, 1967]). The Mach number is proportional to the square root of the ratio of dynamic to static pressure. In the calculation of dynamic pressure above, the Mach number is approximately 0.4, but compressibility effects may be noticed at Mach numbers as low as 0.3. An example of these effects is the variation of the slope of the lift curve with Mach number. The theoretical prediction of this slope is given by the Prandtl-Glauert correction [Anderson, 1984; Kuethe and Chow, 1984]:

$$\frac{dC_L}{d\alpha} = \frac{dC_L/d\alpha|_{M=0}}{\sqrt{1 - M^2}}, \quad M < 1.$$

This equation holds until sonic speed first occurs in the flow.

The different regions of Mach number, where very significant changes in aerodynamic effects can occur, are identified as follows:

Subsonic Speeds: $M < 1.0$

Transonic Speeds: $0.8 \leq M \leq 1.2$

Supersonic Speeds: $1.0 < M < 5.0$

Hypersonic Speeds: $5.0 \leq M$.

Because the velocity varies over the airfoil surface, the airspeed will in general reach sonic values at some point on the surface when the free-stream velocity is still subsonic. This free-stream Mach number, called the *critical Mach number*, defines the beginning of transonic flow for the airfoil.

The transonic regime has the most complex flows, with the appearance of many unique phenomena. For example, at a free-stream Mach number slightly greater than the critical Mach number, a sharp increase in drag coefficient occurs. This is often called the *drag divergence Mach number*. Compression and expansion waves [Anderson, 1984; Kuethe and Chow, 1984] begin to form in the flow field at this Mach number. Compression waves coalesce to form a *shock*, a surface in the flow where virtually discontinuous changes in pressure and velocity occur.

Another important feature of the transonic region is the shift in position of the airfoil aerodynamic center. At low subsonic Mach numbers ($M \ll 1$) it is usually at a distance back from the leading edge equal to about 25% of the chord. In the transonic region its position may change erratically, and at higher speeds it tends to shift aft to the 50% chord position. We will consider the practical implications of this when we discuss the complete aircraft.

Planform Characteristics

A conventional aircraft uses airfoils for the wings, horizontal tail, vertical tail, and possibly for additional surfaces such as horizontal canards (notable exceptions to this configuration are the flying wing aircraft, such as the Northrop YB series [Anderson, 1976] and the more modern B2 bomber). But unlike two-dimensional aerodynamics, real airfoils are finite in length and subjected to interfering flows from other bodies. Therefore, the three-dimensional aerodynamic analysis of a complete aircraft is considerably more complex than the section aerodynamics of an airfoil.

The “planforms” of the aerodynamic surfaces have a large impact on their aerodynamic characteristics. Among the most important parameters of the planform are the aspect ratio and the leading-edge sweep angle. These and other parameters are defined in Table 2.2-1. An explanation of the calculation of the mean geometric chord can be found in various aerodynamics texts (e.g., [Dommash et al., 1967]). The aspect ratio is a measure of span relative

TABLE 2.2-1. Important Wing-Planform Parameters

b = wing span (i.e., tip to tip)	λ = taper ratio (tip chord/root chord)
c = wing chord (varies along span)	
\bar{c} = mean geometric chord	Λ = leading-edge sweep angle
S = wing area (total)	$A = b^2/S$ = aspect ratio

to chord. High-aspect-ratio wings act more "two-dimensional," in that the tip effects are less pronounced. These tend to be used for slower aircraft since a high aspect ratio generates high lift at low speed. At higher speeds this is not needed, so low aspect ratio is used because of the structural advantages of a short span.

Figure 2.2-4 shows a number of distinctive planforms. Low-speed aircraft, ranging from light general-aviation types to military heavy-lift transport aircraft, have stiff moderate-aspect-ratio wings with no sweepback (cf. Fig. 2.2-4d). Aircraft designed to reach transonic speeds and beyond have highly swept wings. The effect of the sweep is to postpone the transonic drag rise, since the component of the airflow perpendicular to the leading edge has its speed reduced by the cosine of the sweep angle. Large jet airliners designed to cruise efficiently at high subsonic Mach numbers have swept wings with a high aspect ratio (Fig. 2.2-4e). This produces the highest ratio of lift to induced drag (the increase in drag that occurs when lift is produced). In the case of high-speed fighter aircraft the requirement for high maneuverability causes a dramatic change to very-low-aspect-ratio wings (Fig. 2.2-4b). The stubby wings allow the aircraft structure to be designed to withstand very high acceleration loads during maneuvers. They also reduce the moment of inertia about the longitudinal axis and the aerodynamic damping moments during rolling, thus promoting a high maximum roll rate.

Wing sweep has the disadvantage of reducing the lift-curve slope of the wing (i.e., less lift at a given alpha) and producing suboptimal performance at low speeds. A way to overcome this when a high lift-to-drag ratio is required over a wide envelope is to use a variable-sweep wing, as exemplified by the F-14 and B-1B aircraft (Fig. 2.2-4a). This is a heavy and costly solution. For commercial aircraft that are usually optimized for one cruise condition, the most common method of achieving adequate lift at low speeds is to increase the camber of the wing by means of leading- and trailing-edge devices (flaps and slats). These may then be deployed manually for landing. A more specialized solution is to use an automatic maneuvering flap, as in the case of the F-16 leading-edge flap, which is deployed automatically as a function of angle of attack when the Mach number is low. More recently, the concept has been taken to its logical conclusion in the *mission adaptive wing* [DeCamp, 1987], tested on an F-111 aircraft.

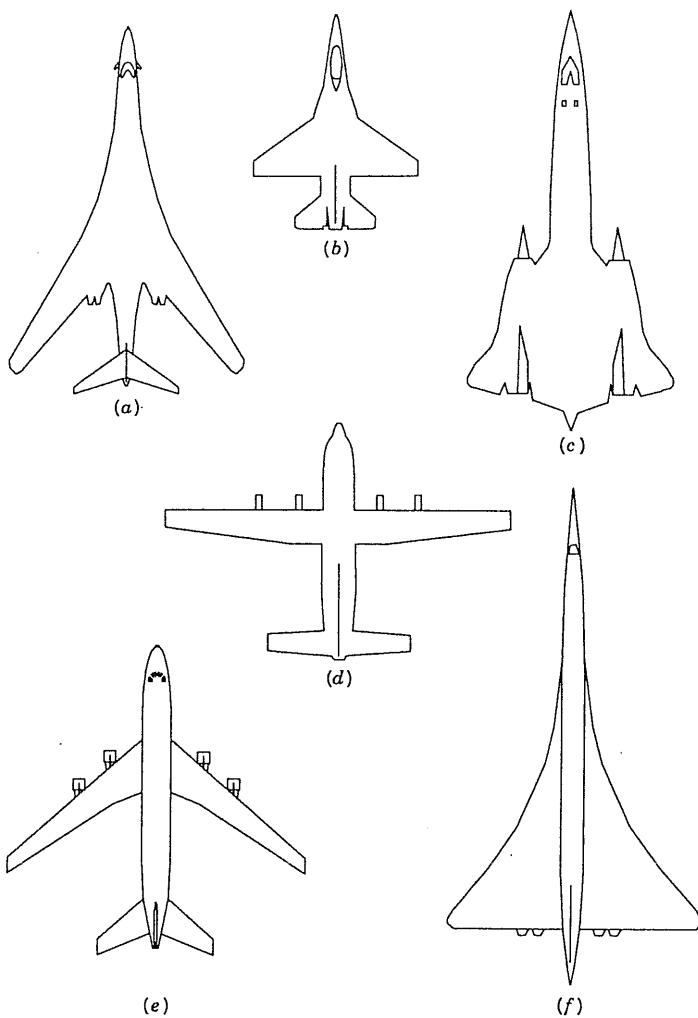


Fig. 2.2-4 Types of aircraft wing planform.

The low lift-curve slope of the swept wing can also be offset to some extent by using *vortex lift*. A conventional wing sheds vortices from the wingtips, and these are actually part of a complete vortex sheet that is spread underneath the airplane and rolls up over the edges of the wings. By extending the leading edge of the wing forward along the fuselage, strong stable vortices can be generated close to the fuselage and allowed to roll back over the inboard portion of the wing. This modifies the pressure distribution over the upper surface of the wing in such a way that the net effect is an increase in lift. The vortex lift becomes stronger with increasing alpha and delays the

onset of flow separation over the inboard portion of the wing (thus providing a higher maximum lift coefficient). Wing planforms that create this effect are shown in Fig. 2.2-4*b*, *c*, and *f*, representing the F-16, SR-71, and Concorde aircraft, respectively.

The F-16 has sharp-edged highly swept forebody *stakes* to generate the vortices, and the design goal was to achieve maximum maneuverability through the use of vortex lift. The Concorde has an *ogee* wing with very large initial sweep angle, with the design aim of increasing the lift at low speed and reducing the movement of the aerodynamic center between low-speed and supersonic cruise conditions. Some description of the design of these wings can be found in the AIAA case studies [Droste and Walker, no date] and [Rech and Leyman, no date]. The SR-71 Mach 3-plus, high-altitude, strategic reconnaissance aircraft [Drendel, 1982] has a blended wing–body with chines. This blending reduces wing–body wave interference drag at cruise speed while Vortex lift effects may be useful during takeoff and landing.

Vortices are also shed from a conventional forebody at high alpha, and a long forebody overhang (as in the case of the shark nose on the F-5) presents difficult design problems. This is because any slight asymmetry in the shed vortices causes pressure differentials at the nose and leads to a relatively large (and unpredictable) yawing moment because of the long lever arm from the aircraft center of gravity.

2.3 AIRCRAFT FORCES AND MOMENTS

The equations of motion derived in Chapter 1 are driven by the aerodynamic forces and moments acting at the cg of the complete rigid aircraft. In Section 2.2 we have covered enough basic aerodynamics to understand how these forces and moments come about. We now begin to examine how they can be measured and expressed.

Definition of Axes and Angles

The aerodynamic forces and moments on an aircraft are produced by the relative motion with respect to the air and depend on the orientation of the aircraft with respect to the airflow. In a uniform airflow these forces and moments are unchanged after a rotation around the free-stream velocity vector. Therefore, only two orientation angles (with respect to the relative wind) are needed to specify the aerodynamic forces and moments. The angles that are used are the *angle of attack* (α) and the *sideslip angle* (β). They are known as the *aerodynamic angles* and will now be defined by means of coordinate rotations in three dimensions. Note that the aerodynamic forces and moments are also dependent on angular rates, but for the moment we are concerned only with their dependence on orientation.

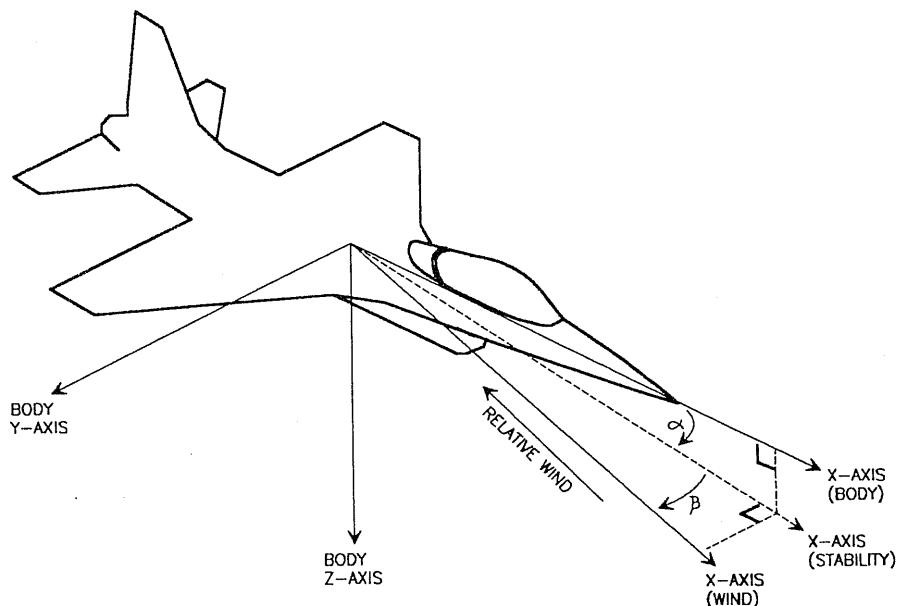


Fig. 2.3-1 Definition of aircraft axes and angles.

The angle of attack usually specified in the aerodynamic data for an aircraft is measured with respect to a fuselage reference line and denoted by α_{frl} . We shall assume that our aircraft body-fixed axes are aligned with this fuselage reference line; otherwise, the alpha used in the body-axes equations of motion would differ by a constant from the alpha used in the aerodynamic data base.

Figure 2.3-1 shows an aircraft with the relative wind on its side (i.e., sideslipping) and with the conventional right-handed (forward, starboard, and down) set of body-fixed axes illustrated. The relative wind vector is equal and opposite to the cg relative velocity vector v_B used in the equations of motion. The angles of attack and sideslip are defined by performing a plane rotation about the body y -axis, followed by a plane rotation about the new z -axis, such that the final x -axis is aligned directly into the relative wind. The first rotation defines the *stability axes*, and the angle of attack is the angle between the body-fixed x -axis and the stability x -axis. Alpha is positive if the rotation about the body-fixed y -axis was negative; thus a positive alpha is shown in the figure. The second rotation leads to a set of *wind axes*, and the sideslip angle is the angle between the stability x -axis and the wind x -axis. Beta is positive if the rotation about the stability z -axis was positive, and a positive beta is shown in the figure.

An aircraft of conventional shape must fly more or less directly into the apparent wind in order to have low drag; therefore, beta is usually very small in steady flight. Alpha must be large enough to generate the required lift but is also usually quite small. Therefore, although the stability axes and wind axes have a variable orientation depending on the flight condition, they essentially point forward, starboard, and down, the same as the body-fixed axes. Note that technically all three sets of axes are "body axes" but only one is body-fixed; we shall drop the term "fixed" and simply refer to them as body, stability, and wind axes.

A left-handed wind-axes system, aligned backwards, left, and "up" relative to the aircraft, has often been used in the past for wind tunnel data [Pope, 1954]. Lift L , drag D , and cross-wind force C were defined naturally in these axes as the aerodynamic force components along the respective positive axes. In the right-handed wind axes that we have defined, we use $-L$ and $-D$ for the z and x force components, and define a component Y for the aerodynamic sideforce measured along the positive y -axis. The symbol Y is also commonly used for the aerodynamic sideforce component along the body y -axis, but it will be clear from the context which axes are being used.

Following our rules for finding rotation matrices, the transformation from body to stability axes is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{STAB}} = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{BODY}}, \quad (2.3-1a)$$

and the rotation from stability axes to wind axes is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{WIND}} = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{STAB}}. \quad (2.3-1b)$$

We shall denote these rotations by S_α and S_β , and the complete rotation from body to wind axes by S (not to be confused with reference area). Therefore, if \mathbf{v}_{BODY} is an arbitrary vector in the body axes, and \mathbf{v}_{WIND} is the same vector expressed in wind axes, then

$$\mathbf{v}_{\text{WIND}} = S_\beta S_\alpha \mathbf{v}_{\text{BODY}} = S \mathbf{v}_{\text{BODY}}, \quad (2.3-2a)$$

where

$$S = \begin{bmatrix} \cos \alpha \cos \beta & \sin \beta & \sin \alpha \cos \beta \\ -\cos \alpha \sin \beta & \cos \beta & -\sin \alpha \sin \beta \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}. \quad (2.3-2b)$$

This transformation will be used extensively in later sections.

TABLE 2.3-1. Force, Moment, and Velocity Definitions

$$\mathbf{F}_B = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} F_{x_A} \\ F_{y_A} \\ F_{z_A} \end{bmatrix} + \begin{bmatrix} F_{x_T} \\ F_{y_T} \\ F_{z_T} \end{bmatrix} \equiv \mathbf{F}_{B_A} + \mathbf{F}_{B_T} \quad (2.3-3a)$$

$$S\mathbf{F}_B = \mathbf{F}_W = \mathbf{F}_{W_A} + \mathbf{F}_{W_T} = \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix} + S\mathbf{F}_{B_T} \quad (2.3-3b)$$

$$\mathbf{T}_B = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \bar{L}_A \\ M_A \\ N_A \end{bmatrix} + \begin{bmatrix} \bar{L}_T \\ M_T \\ N_T \end{bmatrix} \equiv \mathbf{T}_{B_A} + \mathbf{T}_{B_T} \quad (2.3-4a)$$

$$\mathbf{T}_W = \begin{bmatrix} \bar{L}_W \\ M_W \\ N_W \end{bmatrix} = \begin{bmatrix} \bar{L}_{W_A} \\ M_{W_A} \\ N_{W_A} \end{bmatrix} + S\mathbf{T}_{B_T} \equiv \mathbf{T}_{W_A} + \mathbf{T}_{W_T} \quad (2.3-4b)$$

$$\mathbf{v}_B = \begin{bmatrix} U \\ V \\ W \end{bmatrix} = S^T \mathbf{v}_W = S^T \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} V_T \cos \alpha \cos \beta \\ V_T \sin \beta \\ V_T \sin \alpha \cos \beta \end{bmatrix} \quad (2.3-5)$$

Definition of Forces and Moments

The forces and moments at the aircraft cg have components due to aerodynamic effects and to engine thrust; these components will be denoted, respectively, by the subscripts *A* and *T*. The velocity vector of the cg, expressed in wind axes, has (by definition) an *x*-component equal to the true airspeed V_T , and no other components. The various quantities and their relationships to the body-axes quantities, defined in Chapter 1, are given in Table 2.3-1.

The thrust component F_{y_T} can be produced by unbalanced engine power because in a multiengined aircraft, the (inboard) engines may be toed-in to align them with the airflow from the forebody. The component F_{z_T} may result from a design in which the thrust vector is aligned to pass through the cg, when an engine is above or below the cg. From (2.3-5) in Table 2.3-1 we get the following expressions for alpha, beta, and true airspeed:

$$\begin{aligned} \tan \alpha &= \frac{W}{U} \\ \sin \beta &= \frac{V}{V_T} \\ V_T &= (U^2 + V^2 + W^2)^{1/2}. \end{aligned} \quad (2.3-6)$$

Alpha, beta, and V_T are the three most important variables determining the aerodynamic forces and moments, so these equations will prove very useful in conjunction with the body-axes equations of motion.

Force and Moment Coefficients

The forces and moments acting on the complete aircraft are defined in terms of dimensionless aerodynamic coefficients in the same manner as for the airfoil section. Thus we have

$$\begin{aligned}
 \text{drag, } D &= \bar{q}SC_D \\
 \text{lift, } L &= \bar{q}SC_L \\
 \text{sideforce, } Y &= \bar{q}SC_Y \\
 \text{rolling moment, } \bar{L} &= \bar{q}SbC_l \\
 \text{pitching moment, } M &= \bar{q}S\bar{c}C_M \\
 \text{yawing moment, } N &= \bar{q}SbC_N,
 \end{aligned} \tag{2.3-7}$$

where

$$\begin{aligned}
 \bar{q} &= \text{free-stream dynamic pressure} \\
 S &= \text{wing reference area} \\
 b &= \text{wing span} \\
 \bar{c} &= \text{wing mean geometric chord.}
 \end{aligned}$$

Note that the dimensionless force coefficients have been specified for wind axes; the body-axes force coefficients C_X , C_Y , and C_Z are also in common use. The moment coefficients may be specified in wind, stability, or body axes, but the same symbols are used in each case.

The various dimensionless coefficients C_D , C_L , ..., C_N are primarily dependent on the aerodynamic angles alpha and beta and less dependent on a number of other variables. In the next subsection the dependence on the rates of change of the aerodynamic angles and on the components P , Q , and R of the cg angular velocity is described. The coefficients are also dependent on control surface deflections; otherwise, the vehicle would not be controllable. Furthermore, although the effect of airspeed is accounted for through the dynamic pressure, the aerodynamic coefficients are still airspeed dependent at other than low subsonic Mach numbers. This is because the flexibility of the airframe causes it to change shape when large aerodynamic forces are generated at the higher Mach numbers, and because the nature of the flow field changes at high Mach numbers. The aerodynamic coefficients are also dependent on other factors, such as engine power level, configuration effects (e.g., landing gear, external tanks, etc.), and ground-proximity effects.

Because of the complicated functional dependence of the aerodynamic coefficients, each "total" coefficient is modeled as a sum of components that are, individually, functions of fewer variables. Before describing this "component buildup," we shall consider the origin of the rate-dependent components and how they are modeled.

Rate-Dependent Aerodynamic Forces

When an aircraft has an angular velocity, in addition to the translational motion, the flow field around the aircraft is modified and the forces and moments are changed. The effects are modeled, to a first approximation, by adding angular-rate (P , Q , and R)-dependent correction terms to the force and moment equations. Also, when the translational velocity is changing, the angles of attack and sideslip must change. The effects cannot occur simultaneously over the entire aircraft (the airflow takes a significant time to pass from nose to tail), so time-dependent forces and moments are created, in addition to the airspeed and orientation-dependent effects. These effects are modeled as force and moment increments that are proportional to the rates of change of the aerodynamic angles, that is, "alpha-dot" and "beta-dot" effects.

An examination of angular-rate-dependent effects immediately raises the question of how rapidly the airflow adjusts to changes in the aircraft state. The difficult problems of characterizing "unsteady" flows are still the subject of current research, and more accurate aerodynamic models require nonlinear differential equations involving higher derivatives of alpha and beta.

Methods exist to estimate the alpha-dot and beta-dot effects [Perkins and Hage, 1949; Hoak, 1970], and they can be measured in a suitably equipped wind tunnel, although not with very good accuracy. In the case of alpha-dot, for example, the wind-tunnel model can be subjected to an oscillatory plunging motion. Unfortunately, as might be expected, the results are dependent on the frequency of the oscillation. Empirical criteria have been formulated to determine frequency limits below which a quasi-steady assumption (i.e., instantaneous readjustment) may be made about the flow [Duncan, 1952].

In general the rate-dependent effects are modeled with an equation of the form

$$\text{dimensionless force or moment component} = \frac{k}{2V_T} C(\alpha, \beta, M) \times \text{rate.} \quad (2.3-8)$$

The coefficient, C , of the angular rate is known as a *damping derivative*, or in the case of the coefficients of P , Q , and R , as a *rotary derivative*. The damping derivative is typically a function of the variables indicated. The

factor $k/2$ is some characteristic length (e.g., wing semispan) which, when multiplied by the rate, is equivalent to a tangential velocity. When this is divided by V_T we obtain an equivalent perturbation in the local angle of attack or sideslip angle of an aerodynamic surface. This can be thought of as the source of the aerodynamic force or moment.

The angular-rate-dependent damping effects are usually not very significant in the force equations. The force damping derivatives that are sometimes important are those describing the increment in lift due to alpha-dot ($d\alpha/dt$) and pitch rate, that is, C_{L_a} and C_{L_q} . Of lesser importance are the coefficients for the increments in sideforce due to roll rate and yaw rate, C_{Y_p} and C_{Y_r} . Other damping terms, including those dependent on beta-dot ($d\beta/dt$), are usually assumed negligible. Damping is more important in the moment equations, and the damping derivatives C_{m_a} and C_{m_q} are described in the following sections.

The rotary derivatives can be measured with special wind-tunnel apparatus by causing either the direction of the airflow or the aircraft model to oscillate [Queijo, 1971]. More commonly, they are estimated from geometric data [Hoak et al., 1970] and their values refined from flight-test data [Maine and Iliffe, 1980]. The values of these coefficients are quite strongly affected by compressibility effects, and a dynamic-pressure-dependent correction factor is often used in conjunction with them.

Component Buildup

The "total" aerodynamic coefficients in (2.3-7) are usually expressed as a baseline component, plus incremental or correction terms which we indicate by the symbol Δ . The baseline component is primarily a function of alpha, beta, and Mach number. Mach dependence can be removed from the baseline component and treated as a correction term in the case of data for subsonic speeds. The incremental terms that must be added will depend on flight conditions, and typical terms are given below. Data for these equations are derived from wind-tunnel tests on models, flight tests, and aerodynamic-prediction computer programs, and are compiled in the aerodynamic database in tabular or graphical form. The extend to which individual force and moment equations can be broken down into summations of simpler functions of fewer variables depends on the complexity of the airframe, the range of validity required, and the skill of the stability and control engineer.

Component buildups of the dimensionless coefficients, for a conventional subsonic aircraft, are typically of the form shown in Table 2.3-2. The baseline components are the first terms on the right-hand side; they are followed by the most important correction terms. The ellipses indicate that additional corrections are often needed. The variables "el," "rdr," and "ail" are, respectively, the deflections of the elevator, rudder, and ailerons. The equations will now be described in detail, in the same order as they are presented in the table.

TABLE 2.3-2. Typical Aerodynamic Component Buildup Equations

$$C_D \equiv C_D(C_L) + \Delta C_D(\text{el}) + \Delta C_D(\beta) + \Delta C_D(M) + \dots$$

$$C_L \equiv C_L(\alpha, T_C) + \Delta C_L(\text{el}) + \Delta C_L(M) + \Delta C_{L_{ST}}(\alpha, T_C) + \dots$$

$$C_Y \equiv C_Y(\beta) + \Delta C_Y(\text{rdr}) + \dots$$

$$C_I \equiv C_I(\beta) + \Delta C_I(\text{ail}) + \Delta C_I(\text{rdr}) + \frac{b}{2V_T} [C_{I_p}P + C_{I_r}R] + \dots$$

$$C_M \equiv C_M(C_L, T_C) + \Delta C_M(\text{el}) + \Delta C_{M_{ST}}(\alpha, T_C) + \Delta C_M(M)$$

$$+ \frac{\bar{c}}{2V_T} [C_{m_q}Q + C_{m_a}\dot{\alpha}] + \frac{x_R C_L}{\bar{c}} + \dots$$

$$C_N \equiv C_N(\beta) + \Delta C_N(\text{rdr}) + \Delta C_N(\text{ail}) + \frac{b}{2V_T} [C_{n_p}P + C_{n_r}R] + \dots$$

Drag Coefficient C_D

In the equation for the total drag coefficient, C_D , the baseline component $C_D(C_L)$ is typically the drag of the trimmed aircraft (i.e.; control surfaces set to maintain the same steady flight condition). It is usually expressed as a function of the lift coefficient and is then referred to as the *drag polar*. The drag polar can usually be fitted accurately with a parabolic equation of the

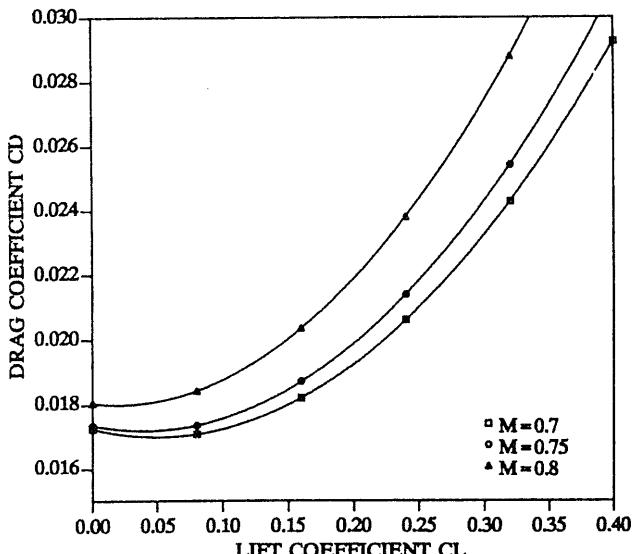


Fig. 2.3-2 Drag polar with compressibility effects.

form

$$C_D(C_L) = k(C_L - C_{L_{DM}})^2 + C_{DM}, \quad (2.3-9)$$

up to the maximum value of the lift coefficient, and for subsonic conditions. Note that the minimum drag, C_{DM} , can occur at a nonzero value, $C_{L_{DM}}$, of the lift coefficient. Drag tends to be independent of Mach number at low Mach and begins to increase rapidly as the transonic regime is entered (compressibility effects). This is shown in Fig. 2.3-2, where the drag polars begin to coalesce below about Mach 0.7 (see, e.g., Roskam [1979]).

Correction terms are added to the baseline drag component to correct for changes in the aircraft configuration (flaps, gear, external tanks, etc.), for control surface deflections, for ground effect, for compressibility, and so on. Ground effects produce greater lift for a given drag; they are usually negligible beyond one wingspan above the ground. In Table 2.3-2 we have indicated only the correction terms for elevator deflection, sideslip, and compressibility effects.

Lift Coefficient C_L

The total lift coefficient, C_L , has the baseline component $C_L(\alpha, T_C)$. The thrust coefficient T_C applies to propeller aircraft and is used to account for propeller wash over the wings. It is defined by normalizing engine thrust in the same way as the nondimensional coefficients; thus

$$T_C = \frac{\text{thrust}}{\bar{q}S_D}, \quad (2.3-10)$$

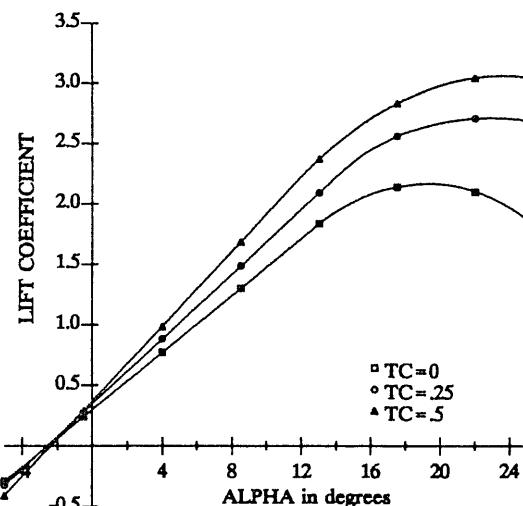


Fig. 2.3-3 Lift curves with thrust coefficient as a parameter.

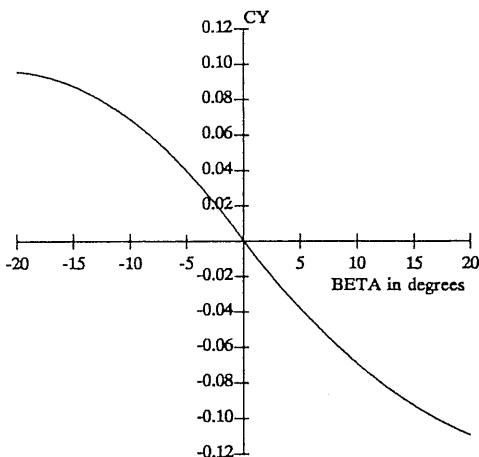


Fig. 2.3-4 Typical side-force coefficient versus sideslip.

where S_D is the area of the disc swept out by a propeller blade. A typical family of lift curves with alpha as the independent variable and T_C as a parameter is shown in Fig. 2.3-3. The remaining incremental terms in the lift coefficient equation are, respectively, the effect of elevator deflection, the effect of Mach number, and the correction for nonlinear behavior of the lift curve at high angles of attack (i.e., near stall).

Side-Force Coefficient C_Y

In the case of a symmetrical aircraft, side force is created mainly by sideslipping motion (i.e., $\beta \neq 0$) and by rudder deflection. A typical variation of side force as a function of sideslip is shown in Fig. 2.3-4. The side-force coefficient has negative values when beta is positive (wind on the right-hand side of the aircraft) because side force was defined to be positive along the positive y -axis. Values of C_Y of the order of 0.1 are typical for a medium-sized aircraft at a few degrees of sideslip. The curve is strongly influenced by flap settings, the effects of propeller wash, and compressibility effects, and these effects may be included in the baseline component. The correction terms added to the baseline component may include the effects of rudder deflection, landing gear deployment, and rudder tab setting.

Rolling Moment C_l

The baseline term in the equation for the rolling moment C_l is $C_l(\beta)$, the *dihedral effect* or rolling moment dependence on beta. A typical variation of $C_l(\beta)$ is shown in Fig. 2.3-5, and values on the order of 0.01 can be expected

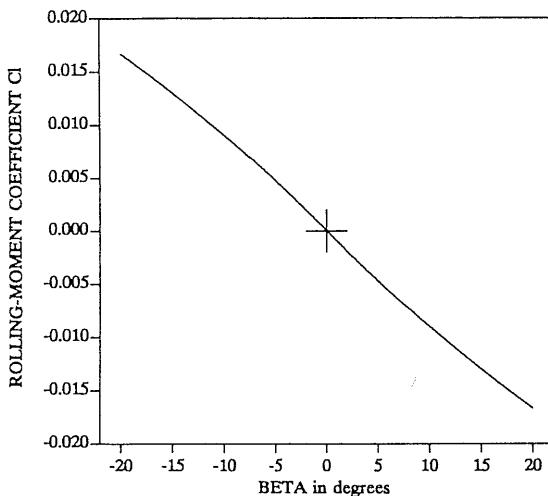


Fig. 2.3-5 Typical rolling moment coefficient versus sideslip.

at a few degrees of sideslip for a medium-sized aircraft. The other terms in the rolling moment equation represent, respectively, the components due to aileron deflection, rudder deflection, roll rate, and yaw rate.

Consider now the basic dihedral effect; the effect of sideslip is to create a lateral component of the relative wind, and there are three separate effects of this lateral component on the horizontal aerodynamic surfaces. These will now be described.

First, note that the angle by which the wings of an aircraft are canted up above the body-axes $x-y$ plane is called the *dihedral angle*, and a negative dihedral angle is called an *anhedral angle*. "Dihedral" is often very noticeable on small low-wing (wing root attached at the bottom of the fuselage) aircraft, while a well-known example of anhedral is the tail of the McDonnell Douglas F-4. Dihedral (or anhedral) gives one wing a positive angle of attack (in a spanwise direction) to the lateral component of the relative wind, and the other wing receives a similar negative angle of attack. Referring to Fig. 2.3-1, it is easy to see that positive beta creates a negative rolling moment when the wings have dihedral. This same effect applies to the horizontal tail.

The second effect of sideslip on the horizontal surfaces is that on the windward side of the fuselage some of the lateral airflow is diverted up and over the fuselage, and some is diverted under the fuselage. This flow will modify the angle of attack of the wings, depending on their position on the fuselage. On the windward side of the aircraft, above the centerline of the fuselage, the upward component of the relative wind is increased. Therefore, the angle of attack of that wing is also increased (assuming that it was operating at a positive alpha). In this case a positive beta again corresponds to a negative rolling moment. Again this effect applies to the horizontal tail.

The third and final effect of sideslip on the horizontal surfaces occurs when they are swept back. In this case the lateral component of the relative wind makes it more nearly perpendicular to the leading edge of the windward wing than is the case for the leeward wing. Therefore, the windward wing develops more lift, and the outcome is again a negative rolling moment for positive beta.

Finally, the lateral component of the relative wind acting on the vertical tail will generate a rolling moment about the cg. Depending on the aircraft angle of attack and the location of the center of pressure of the vertical tail, this rolling moment could be positive or negative. Usually, positive beta will produce a negative rolling moment component. As indicated earlier, all of the effects above are included in the airplane dihedral effect as measured in the wind tunnel. In the case of propellers, producing a strong wash over the wings, the dihedral effect can be expected to diminish with increasing thrust coefficient. Wing flaps can also produce dramatic changes in the dihedral effect.

The dihedral effect is responsible for giving an aircraft an inherent tendency to fly with wings level. This tendency to resist rolling, called *roll stiffness*, is analogous to a restoring spring torque in a mechanical system. It comes about because gravity causes the aircraft to begin to sideslip if the wings are not level; the dihedral effect may then generate a restoring moment that returns the aircraft to level flight. It is not possible to determine from the dihedral effect if the wings-level condition is a stable equilibrium condition because the rolling and yawing motions of an aircraft are coupled (interdependent). In Chapter 3 we provide the tools for a stability analysis that takes into account all of the dynamic coupling effects.

The last two terms in the rolling moment equation are the roll damping due to roll rate and yaw rate, respectively. The factor $Pb/2$ in the roll damping is the tangential velocity at the wingtip, and $Pb/2V_T$ is equivalent to a perturbation in the angle of attack at the wingtip. The perturbation in angle of attack over the entire wing, due to rolling, is the source of the damping moment. A similar argument applies to the term $Rb/2V_T$. The coefficients C_{l_p} and C_{l_r} are obviously dimensionless and are the rotary derivatives for the roll damping.

Control Effects on Rolling Moment

We turn now to the effects of the lateral/ directional control surfaces on the rolling moment; these are represented by the increments in rolling moment due to aileron and rudder deflections. The rudder is intended to provide (yaw) directional control, so the "cross-control" effect on rolling moment is an unwanted effect. This effect comes about because the center of pressure of the rudder is normally above the stability x -axis. As angle of attack increases, the body x -axis becomes more inclined to the stability x -axis, and this effect typically decreases. With respect to the ailerons, we will use "ail" to mean the variable used to produce rolling moments, and physically it may

correspond to some combination of differential motions of ailerons, horizontal tail surfaces, or spoilers. Conventional ailerons mounted outboard on the trailing edge of the wings become ineffective and then reverse their effect as high subsonic speeds are approached. This is because the aileron lift component produced by a downward deflection twists the wing in the direction that reduces its angle of attack and hence reduces the wing-lift component.

Spoilers, which are uncambered surfaces deflected upward above the trailing edge of the wing, "spoil" the lift on that portion of the wing and thus provide roll control. The twisting effect on the wing is reduced and control reversal can be avoided. Mounting the ailerons further inboard reduces the effect of wing twist but also reduces their moment arm. However, the X-29 forward-swept wing aircraft is an example of combined inboard and outboard "flaperons" being made to work very effectively up to high alpha [Kandebo, Scott, 1988]. The effectiveness of both ailerons and spoilers is reduced by cross flows on the wing and hence by wing sweep. Therefore, for swept-wing aircraft, another approach to roll control is to use (in addition to ailerons) differential control of the horizontal-tail control surfaces (e.g., the F-16 aircraft). In conclusion, the rolling moment equation can be expected to need correction terms for aileron effectiveness, and these may need to be a function of dynamic pressure, lift coefficient, and aileron deflection itself.

Pitching Moment C_M

The second moment equation in Table 2.3-2 is that for the total pitching moment C_M . It contains a baseline component, shown here as $C_M(C_L, T_C)$, and a number of correction terms. Consider first the correction term $(x_R C_L)/\bar{c}$; the purpose of this term is to correct for any x -displacement (x_R) of the aircraft cg from the aerodynamic data reference position. If x_R is not zero, the lift force will provide a contribution to the pitching moment. This moment balance will now be considered in more detail.

The moment balance on the longitudinal axis of the aircraft is critical to both performance and stability. The major aerodynamic force, namely the lift, creates moments on this axis. If the lift force generated by the wings also creates a large moment about the cg, the horizontal tail must carry a significant load. If this turns out to be a download, the overall effect is additional drag with a net reduction in lift and reduced load-carrying efficiency. In Section 2.2 we saw that the moment about the airfoil aerodynamic center was constant and relatively small; and for positive *pitch stiffness*, the axis about which the airfoil pivoted needed to be slightly ahead of the ac. Therefore, the cg of the aircraft should be close to the ac of the wings, and the pitch stiffness of the complete aircraft must be analyzed. We also saw in Section 2.2 that the zero-alpha moment, M_0 , needed to be positive in order to obtain equilibrium with a positive angle of attack (and therefore provide arbitrarily large lift). For a conventional aircraft this is achieved by giving the horizontal tail a negative incidence, so that it provides a positive contribution to the total pitching moment. Efficiency can be improved by reducing the

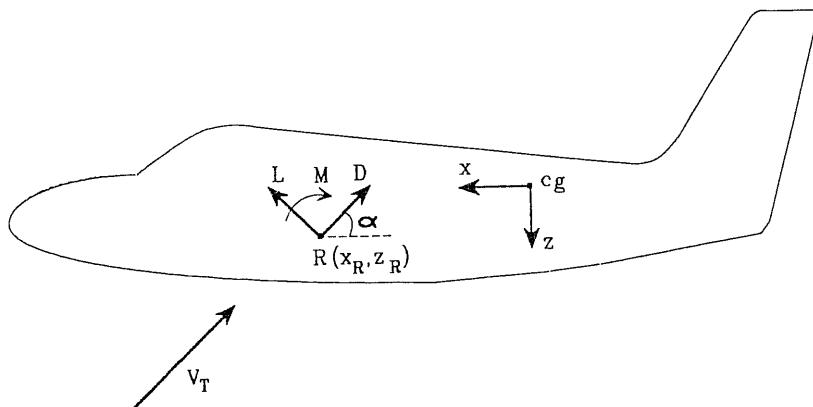


Fig. 2.3-6 Diagram for analysis of pitch stiffness.

pitch stiffness [this is done in relaxed static stability (RSS) designs], but then the flight control system must be designed to deal with the pitch instability that may occur with some flight conditions or cg positions.

The pitch stiffness can be analyzed by considering Fig. 2.3-6. In this figure the point $R(x_R, 0, z_R)$ is the reference position used to measure the aerodynamic data for the complete aircraft, and L , D , and M are the lift, drag, and pitching moment obtained from the database. The origin for the body-axes coordinate system is at the cg of the aircraft as shown. It is evident by inspection of the figure that the moment about the cg is given by

$$M_{cg} = M + (L \cos \alpha + D \sin \alpha)x_R + (L \sin \alpha - D \cos \alpha)z_R. \quad (2.3-11)$$

Now divide through the equation by $\bar{q}S\bar{c}$ to obtain the nondimensional pitching moment coefficient on the left-hand side. The result is

$$C_{M_{cg}} = C_M + (C_L \cos \alpha + C_D \sin \alpha) \frac{x_R}{\bar{c}} + (C_L \sin \alpha - C_D \cos \alpha) \frac{z_R}{\bar{c}}.$$

Under steady level flight conditions the $C_L \cos \alpha$ term is by far the largest aerodynamic component on the right-hand side; consequently, the x_R displacement is weighted much more strongly than z_R . For simplicity, we shall analyze the pitch stiffness with this term only, and also assume that the angle of attack is small ($\cos \alpha \approx 1$). The approximate equation is

$$C_{M_{cg}} = C_M + \frac{x_R C_L}{\bar{c}}. \quad (2.3-12)$$

This equation now contains the approximate correction term that was added

to the pitching moment equation in Table 2.3-2; the effect is often neglected in the rolling and yawing moment equations.

To determine whether the aircraft will have positive pitch stiffness, we must differentiate (2.3-12) to discover whether $C_{M_{cg}}$ increases or decreases with increasing alpha; thus

$$\frac{dC_{M_{cg}}}{d\alpha} = \frac{dC_M}{d\alpha} + x_R \frac{dC_L/d\alpha}{\bar{c}}. \quad (2.3-13)$$

If the sign of $dC_{M_{cg}}/d\alpha$ is negative, the aircraft has positive pitch stiffness. The first term on the right-hand side is the slope of the pitching moment of the complete aircraft, measured at the aerodynamic data reference point; this can be designed to be negative to produce positive pitch stiffness. The second term contains $dC_L/d\alpha$ which we know to be a positive quantity, and therefore, depending on x_R/\bar{c} , the aircraft may have positive or negative pitch stiffness. Note that the analysis assumes that the control surfaces are locked in position.

Next, assume that the cg position is moved rearward so that x_R is increased; a position will be reached where the pitch stiffness becomes zero. This position is known as the *neutral point*, which is effectively the aerodynamic center for the complete aircraft. The distance of the neutral point behind the actual cg, divided by \bar{c} , is called the *static margin*. Therefore, for positive pitch stiffness, a positive static margin is required.

A conventionally balanced aircraft is usually designed to have a minimum positive static margin of between 3 and 5% (0.03 to 0.05). This is for safety reasons and to allow some margin for cg variations with load conditions. Aircraft that operate into the transonic and supersonic regions pay a price for this low-speed static stability. The aerodynamic center of an airfoil tends to shift aft from $0.25\bar{c}$ toward $0.5\bar{c}$ in going from high subsonic speeds to supersonic speeds (see Section 2.2). This causes a corresponding movement in the aircraft neutral point and a large increase in the static margin. The undesirable consequences are increased trim drag (and therefore reduced range or fuel economy) and reduced maneuverability. Some modern military aircraft (notably the F-16) have minimized these penalties by using a reduced, or negative, static margin at subsonic speeds. Since negative pitch stiffness normally leads to dynamic instability in pitch, these aircraft must use an automatic control system to restore pitch stability. This is described in later chapters.

Turning now to the other terms shown in the pitching moment equation, the baseline pitching moment coefficient is normally expressed as a function of lift coefficient or of angle of attack. Figure 2.3-7 shows a typical variation of this component with alpha, for a propeller aircraft at low subsonic speeds. The graph shows a pitching moment that decreases with increasing alpha; this is a basic requirement for stability in pitch as described above. Thrust coefficient is shown as a parameter, and it can be seen that a strong propeller

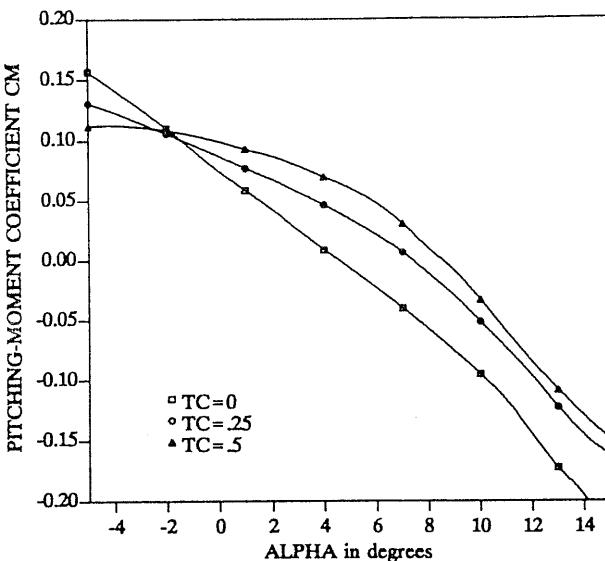


Fig. 2.3-7 Pitching moment curves with thrust coefficient as a parameter.

slipstream (high T_C) tends to reduce the dependence on alpha at low values of alpha. The terms $\Delta C_{M_{ST}}(\alpha, T_C)$ and $\Delta C_M(M)$ in the pitching moment equation are, respectively, to correct the basic data for high-alpha effects (near stall) and for movement of the effective aerodynamic center of the aircraft with Mach number. Additional terms will be needed to account for landing gear, flaps, ground effect, and so on.

The damping terms in the pitching moment equation are of the same form as those in the rolling moment equation, but nondimensionalized by the factor $\bar{c}/2V_T$ rather than $b/2V_T$. The pitching moment due to pitch rate C_{m_q} is created by the same mechanism as the other damping derivatives (i.e., angular rates causing tangential velocity components that change the local angle of attack). The longitudinal position of the aircraft cg is usually between the fore and aft edges of the wing, so that C_{m_q} is chiefly determined by the changing lift of the horizontal tail as it pitches about the cg on its long moment arm.

The pitching moment due to alpha-dot (coefficient C_{m_d}) is created by the aerodynamic interaction between the wings and the horizontal tail. We noted earlier the vortex-sheet motion of the airflow behind the aircraft wing. This airflow is felt at the horizontal tail as a downwash, which reduces the effective angle of attack of the tail surface. If the wing angle of attack is changed, there will be a propagation delay proportional to $1/V_T$ before the effect is felt at the tail. Depending on the magnitude of the rate $\dot{\alpha}$, the wing angle of attack may change significantly before the resulting change in downwash at the tail occurs. This delaying effect on the pitching moment

contribution of the horizontal tail can be approximated by adding a component proportional to $\dot{\alpha}$ to the total aircraft pitching moment equation. Thus the linear (when V_T is constant) damping term $(\bar{c}/2V_T)C_{M_\alpha}\dot{\alpha}$ is often added to the pitching moment equation.

As in the case of the rolling moment damping derivatives, C_{M_q} and C_{M_α} can be determined experimentally by special wind tunnel measurements. They are difficult to separate, and both pitch oscillations and vertical oscillatory motion (plunging) must be used.

Control Effects on Pitching Moment

The pitching moment increment due to elevator deflection $\Delta C_M(\text{el})$ is more complicated than implied by the single term in our moment equation. First, because of wing downwash, propeller slipstream, and flap wake effects, the effective dynamic pressure at the horizontal tail is not the same as at the wings. Therefore, the elevator moment term is often multiplied by a correction factor $\bar{q}_{\text{TAIL}}/\bar{q}$, and this factor may be a function of angle of attack, with thrust coefficient and flap settings as parameters. Second, the elevator effectiveness decreases with increasing dynamic pressure because of aeroelastic effects. This effect begins at quite low subsonic Mach numbers and may progress to actual control reversal at a high subsonic Mach number. It is accounted for by multiplying the elevator moment coefficient by another correction factor. Control reversal is one of the effects that determine the limits of the flight envelope of an aircraft.

Yawing Moment C_N

The final moment equation is that for the yawing moment C_N , with baseline component $C_N(\beta)$. A typical plot of this component is shown in Fig. 2.3-8. A symmetrical (about the $x-z$ plane) aircraft has no yawing moment when there is no sideslip, so the curve passes through the origin and is skew-symmetric, as shown. $C_N(\beta)$ is easily measured in the wind tunnel; but it may be important to determine the dependence on alpha also, when the angle of attack is large.

To obtain some insight into the variation of $C_N(\beta)$ it is necessary to consider the moment contributions of the tail, wings, and fuselage. The effect of the vertical tail is to cause the aircraft to weathercock into the apparent wind; that is, positive sideslip produces a positive yawing moment as shown in the figure. This is positive yaw stability and is also known as *weathercock stability*. The contribution of the wings to this yawing moment is in the same sense as the vertical tail contribution. This is because, with wing sweep, the windward wing develops more lift and therefore more induced drag, as described earlier. Because of interference effects between the wings and fuselage, the effect of the fuselage on the yawing moment is complicated, and it is found that the contribution is actually negative when β is positive. This effect can cause $C_N(\beta)$ to change sign at high alpha. The shape of the plot

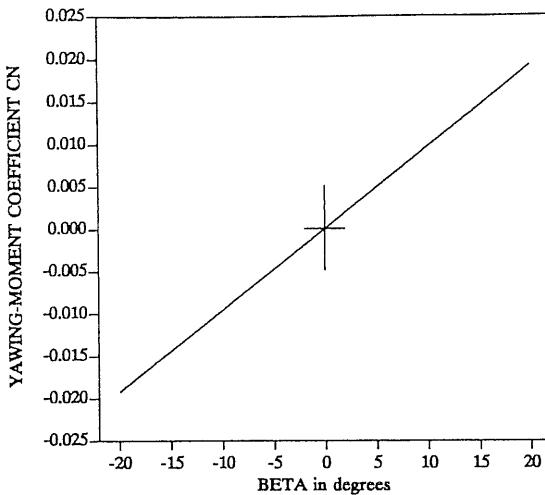


Fig. 2.3-8 Typical variation of yawing moment with beta.

can be strongly influenced by the combination of flaps and propeller slipstream, becoming asymmetric and no longer passing through the origin. Mach number can also have a strong effect on $C_N(\beta)$.

Control Effects on Yawing Moment

The rudder usually forms a part of the trailing edge of the vertical tail; it is the primary control over yawing moment. Its action is relatively uncomplicated. The vertical tail is no longer a symmetric airfoil when the rudder is deflected, and thus begins to produce "lift." The resulting side force is such that deflection of the rudder trailing edge to the right produces a positive yawing moment. All-moving vertical fins are sometimes used for rudder control, as for example on the SR-71, where large yawing moments can occur as a result of an engine "unstart."

The normal differential operation of the ailerons produces a difference in drag between the two sets, and hence a cross-control effect on the yawing moment. As described earlier, roll control can be obtained in a number of different ways, so the cross-control effects on yawing moment can vary greatly.

Data Handling

It should be clear from the foregoing description of aerodynamic forces and moments that the aerodynamic database for a given aircraft can be rather large. It may easily involve 100 or more large data tables, and many will be

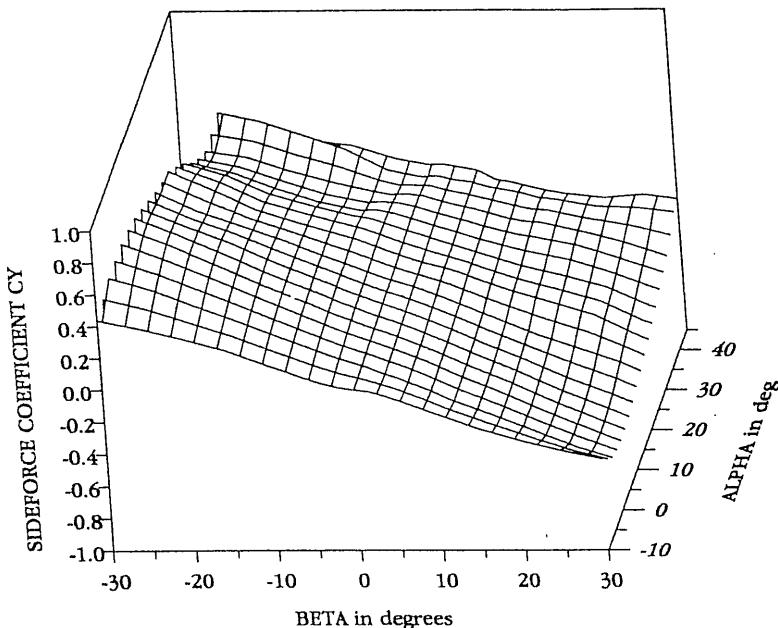


Fig. 2.3-9 Three-dimensional view of aerodynamic side-force coefficient.

two- or three-dimensional. A large aerodynamic database must be handled efficiently within an aircraft company; it represents thousands of hours of planning, model testing, flight testing, and computer simulation. It must be kept current, with all changes fully documented, and be accessible to many different users. The control engineer will have access to the database through a computer workstation and will be able to call up the appropriate force and moment routines for the equations of motion. However, our purpose here is not to describe database management but to make the control engineer aware of the database and his or her dependence on it, and to set up the mini-database needed for our models.

The data can be in terms of the stability axes or in terms of body axes; an example of the latter is given by Nguyen et al. [1979] for low-speed F-16 model data, taken at the NASA Dryden and Langley Research Centers. A three-dimensional plot made from one of the two-dimensional (two independent variables) tables of this F-16 data is shown in Fig. 2.3-9. A reduced data set derived from this report is listed in the appendices and is used for the F-16 model given in Chapter 3.

Aerodynamic lookup-table data are discrete, whereas aircraft models require data at arbitrary values of the independent variables. This problem is solved by using an interpolation algorithm with the data. In the appendices we have provided a simple interpolation algorithm for use with the F-16 data.

Basic aerodynamic data are usually quite rough (scattered data points). This is because of the inaccuracies associated with measuring aerodynamic data, the sensitivity to small changes in the independent variables, and fusion of data from different sources or test runs. The data can be smoothed and regenerated at new uniform increments of the independent variables as required, by means of a spline algorithm [IMSL, 1980].

2.4 THE NONLINEAR AIRCRAFT MODEL

In this section we first expand the vector equations of motion from Chapter 1 and then, using the aerodynamic knowledge gained from previous sections, we describe how the aerodynamic forces and moments can be incorporated into these equations. The flat-Earth vector equations (1.5-4) will be used, and when these are expanded the standard 6-DOF (degrees of freedom) equations used for conventional aircraft control design and flight simulation will be obtained. The elements of the state vector will comprise, respectively, the components of the velocity vector \mathbf{v}_B , the vector of Euler angles Φ , the angular rate vector $\boldsymbol{\omega}_B$, and the position vector \mathbf{p}_{NED} . Therefore, we have the state vector

$$\mathbf{X}^T = [U \ V \ W \ \phi \ \theta \ \psi \ P \ Q \ R \ p_N \ p_E \ h], \quad (2.4-1)$$

where h is altitude in the NED frame.

When the matrix-vector products in (1.5-4) are multiplied out (Problem 2.4-1) using the components above and the information given in Chapter 1, the standard set of *body-axes state equations* [(2.4-2)–(2.4-5)] shown in Table 2.4-1 are obtained.

The moment equations in Table 2.4-1 were previously expanded as (1.3-21), but here we have introduced the constants c_i [McFarland, 1975], defined by

$$\begin{aligned} \Gamma c_1 &= (J_y - J_z)J_z - J_{xz}^2, & \Gamma c_2 &= (J_x - J_y + J_z)J_{xz}, \\ \Gamma c_3 &= J_z, & \Gamma c_4 &= J_{xz} \\ c_5 &= \frac{J_z - J_x}{J_y}, & c_6 &= \frac{J_{xz}}{J_y}, \\ c_7 &= \frac{1}{J_y}, & \Gamma c_8 &= J_x(J_x - J_y) + J_{xz}^2, \\ \Gamma c_9 &= J_x, \end{aligned} \quad (2.4-6)$$

where

$$\Gamma = J_x J_z - J_{xz}^2 \quad [\text{as in (1.3-19b)}].$$

TABLE 2.4-1. The Flat-Earth, Body Axes 6-DOF Equations

Force Equations

$$\begin{aligned}\dot{U} &= RV - QW - g'_0 \sin \theta + \frac{F_x}{m} \\ \dot{V} &= -RU + PW + g'_0 \sin \phi \cos \theta + \frac{F_y}{m} \\ \dot{W} &= QU - PV + g'_0 \cos \phi \cos \theta + \frac{F_z}{m}\end{aligned}\quad (2.4-2)$$

Kinematic Equations

$$\begin{aligned}\dot{\phi} &= P + \tan \theta(Q \sin \phi + R \cos \phi) \\ \dot{\theta} &= Q \cos \phi - R \sin \phi \\ \dot{\psi} &= \frac{Q \sin \phi + R \cos \phi}{\cos \theta}\end{aligned}\quad (2.4-3)$$

Moment Equations

$$\begin{aligned}\dot{P} &= (c_1 R + c_2 P)Q + c_3 \bar{L} + c_4 N \\ \dot{Q} &= c_5 PR - c_6(P^2 - R^2) + c_7 M \\ \dot{R} &= (c_8 P - c_2 R)Q + c_4 \bar{L} + c_9 N\end{aligned}\quad (2.4-4)$$

Navigation Equations

$$\begin{aligned}\dot{p}_N &= U \cos \theta \cos \psi + V(-\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi) \\ &\quad + W(\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \\ \dot{p}_E &= U \cos \theta \sin \psi + V(\cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi) \\ &\quad + W(-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi) \\ \dot{h} &= U \sin \theta - V \sin \phi \cos \theta - W \cos \phi \cos \theta\end{aligned}\quad (2.4-5)$$

Note that the moment equations may require additional terms representing the angular momentum of spinning rotors, as given in (1.3-24). In the navigation equations (2.4-5) \dot{p}_N , \dot{p}_E , and \dot{h} are, respectively, the north, east, and vertical components of the aircraft velocity in the locally level geographic frame on the surface of the Earth.

The force and moment components (F_x , F_y , F_z , \bar{L} , M , N) in the 6-DOF equations must be broken into aerodynamic and thrust contributions as in Table 2.3-1. Then the aerodynamic contributions can be obtained from (2.3-7) and a set of equations like Table 2.3-2. Also, a table of installed

engine thrust versus altitude and speed (or Mach number) is needed, with throttle setting as a parameter. A number of the aerodynamic force and moment components contain dependence on the control surface deflections, these are the control inputs to the model. Throttle setting is another control input; and the implied input vector of the nonlinear model is

$$\mathbf{U}^T = [\text{thl} \quad \text{el} \quad \text{ail} \quad \text{rdr}], \quad (2.4-7)$$

where the elements of the vector are, respectively, throttle setting, elevator deflection, aileron deflection, and rudder deflection.

Incorporating Aerodynamic Forces and Moments

The aerodynamic force and moment components in (2.4-2) and (2.4-4) depend on the aerodynamic angles and the true airspeed; therefore, we must use (2.3-6) to calculate these quantities. It is usually convenient to go one step further and replace the state variables U , V , and W in the 6-DOF model by V_T , β , and α . Then U , V , and W can be calculated from (2.3-5) for the force equations, and from (2.3-6) we can derive the following expressions for the new state derivatives:

$$\begin{aligned}\dot{V}_T &= \frac{U\dot{U} + V\dot{V} + W\dot{W}}{V_T} \\ \dot{\beta} &= \frac{\dot{V}V_T - V\dot{V}_T}{V_T^2 \cos \beta} \\ \dot{\alpha} &= \frac{U\dot{W} - W\dot{U}}{U^2 + W^2}.\end{aligned}\quad (2.4-8)$$

The new state vector is

$$\mathbf{X}^T = [V_T \quad \beta \quad \alpha \quad \phi \quad \theta \quad \psi \quad P \quad Q \quad R \quad p_N \quad p_E \quad h]. \quad (2.4-9)$$

At this point a difficulty arises because the aerodynamic force coefficients can contain alpha-dot or beta-dot dependence. However, the alpha-dot and beta-dot derivatives are not available until *after* the force equations have been evaluated. In terms of the state-variable formulation, the body-axes state equations cannot be arranged in a form that allows an explicit solution for the derivatives. It will be shown in the next subsection that this problem can be avoided by a change of axes, provided that the aerodynamic forces are linearly dependent on $\dot{\alpha}$ and $\dot{\beta}$.

If alpha-dot and beta-dot dependence is present only in the aerodynamic moment coefficients, implicit state equations can be avoided simply by evaluating the force equations before the moment equations. This is a

common situation; the rate derivative C_{L_α} is frequently neglected as are all the beta-dot derivatives, while the derivative C_{m_α} is significant.

Assuming that neither alpha-dot nor beta-dot dependence is present in the force equations, the calculations that must typically be performed in an explicit state model are as follows:

1. Compute Mach number and dynamic pressure from a standard atmosphere model. Then compute engine thrust for use in the force equations.
2. Compute the aerodynamic coefficients for the force equations, compute U, V, W , from V_T, α, β , evaluate the force equations, and then compute the state derivatives $\dot{V}_T, \dot{\alpha}, \dot{\beta}$.
3. Compute the aerodynamic coefficients for the moment equations, using $\dot{\alpha}$ and $\dot{\beta}$ if necessary, and then evaluate the moment equations.
4. Evaluate the kinematic equations and the navigation equations.

In Chapter 3 a simulation model is built as outlined above, incorporating the aerodynamic, propulsion, and standard atmosphere data given in the appendices.

Nonlinear Model in Other Coordinates

In addition to the body-axes equations it is important to be able to express the equations of motion in wind axes or in stability axes (as a special case). The wind axes are the natural axes for the aerodynamic forces. When the rigid-body force equations are expressed in the same axes in which the aerodynamic coefficients are measured, explicit nonlinear state equations can easily be found when the C_{L_α} and C_{L_β} coefficients are present. Furthermore, the wind-axes equations will later lead to *linear, small-perturbation equations* whose coefficients contain the partial derivatives of the aerodynamic forces and moments with respect to the wind-axes variables α , β , and V_T . These are the *aerodynamic derivatives* (or "stability derivatives"); methods exist to estimate their values for a complete aircraft, and the values are commonly specified in aerodynamic databases.

The wind axes change their orientation relative to the aircraft body during flight; and if the equations of motion are expressed in any axes that are not body-fixed, time-varying moments of inertia are introduced into the moment equations. This is not necessarily a disadvantage, since a time-varying transformation of the fixed inertias can easily be handled in a digital simulation. Furthermore, linearization requires equations for small perturbations about a given flight condition, and then a fixed transformation of the inertias can be used.

Conversion of the body-axes equations to wind-axes equations, one equation at a time, is an extremely laborious process. An alternative which is much less laborious, and very instructive, is to transform the vector-matrix

equations using the body-to-wind-axes rotation matrix S [equation (2.3-2b)]. Using this approach, we start from the flat-Earth equations (1.5-4), premultiply throughout by S , and make use of the property $S\dot{S}^T = \Omega_R$ [equation (1.2-38b)], where Ω_R is a cross-product matrix containing the relative angular rates of the wind axes with respect to the body axes. The wind-axes quantities defined in Table 2.3-1 will also be needed. For simplicity, the engine thrust vector will be assumed to be parallel to the body x -axis. The body-axes thrust force component will be denoted by F_T (the double-subscript notation of Table 2.3-1 has been dropped in this simple case) and the thrust moment by M_T . The force equations will be converted first.

Wind-Axes Force Equations

The subscript W will be used to denote wind-axes quantities, and in the case of the force equations in (1.5-4), the steps in the conversion to wind-axes are:

$$\begin{aligned}\frac{1}{m}\mathbf{F}_B &= \dot{\mathbf{v}}_B + \boldsymbol{\omega}_B \times \mathbf{v}_B - B_B \mathbf{g}'_0 \\ \frac{1}{m}S\mathbf{F}_B &= S \frac{d}{dt}(S^T \mathbf{v}_W) + (S\boldsymbol{\omega}_B) \times (S\mathbf{v}_B) - SB_B \mathbf{g}'_0 \\ \frac{1}{m}\mathbf{F}_W &= (SS^T \dot{\mathbf{v}}_W + S\dot{S}^T \mathbf{v}_W) + \boldsymbol{\omega}_W \times \mathbf{v}_W - SB_B \mathbf{g}'_0 \\ \frac{1}{m}\mathbf{F}_W &= \dot{\mathbf{v}}_W + \Omega_R \mathbf{v}_W + \boldsymbol{\omega}_W \times \mathbf{v}_W - SB_B \mathbf{g}'_0.\end{aligned}\quad (2.4-10)$$

The second and third terms on the right-hand side of (2.4-10) are equivalent to $(\boldsymbol{\omega}_R + \boldsymbol{\omega}_W) \times \mathbf{v}_W$; and $(\boldsymbol{\omega}_R + \boldsymbol{\omega}_W)$ is the absolute angular velocity vector of the body axes expressed in wind axes plus the angular velocity of the wind axes relative to the body axes (i.e., the absolute angular velocity of the wind axes). If we combine these terms, (2.4-10) is the original body-axes equation but with body-axes quantities replaced by wind-axes quantities. We will not do this because the Ω_R term contains the derivatives of α and β that will be used as state derivatives in the new coordinates. Also, the components of $\boldsymbol{\omega}_W$, that is, the body-axes rates expressed in wind axes, are often quoted in wind tunnel data. Therefore, we now expand (2.4-10) as it stands.

The cross-product matrix for the wind-axes relative angular velocity is [from (1.2-38b)] given by

$$\Omega_R = S\dot{S}^T$$

Therefore,

$$\Omega_R = S_\beta S_\alpha (S_\beta \dot{S}_\alpha + \dot{S}_\beta S_\alpha)^T$$

and

$$\Omega_R = S_\beta (S_\alpha \dot{S}_\alpha^T) S_\beta^T + S_\beta \dot{S}_\beta^T,$$

which expands to

$$\Omega_R = \begin{bmatrix} 0 & -\dot{\beta} & -\dot{\alpha} \cos \beta \\ \dot{\beta} & 0 & \dot{\alpha} \sin \beta \\ \dot{\alpha} \cos \beta & -\dot{\alpha} \sin \beta & 0 \end{bmatrix}. \quad (2.4-11)$$

The vector \mathbf{v}_W has no y - and z -components, and its x -component is the true airspeed V_T ; therefore, the term $(\dot{\mathbf{v}}_W + \Omega_R \mathbf{v}_W)$ in (2.4-10) reduces to

$$\dot{\mathbf{v}}_W + \Omega_R \mathbf{v}_W = \begin{bmatrix} \dot{V}_T \\ \dot{\beta} V_T \\ \dot{\alpha} V_T \cos \beta \end{bmatrix}. \quad (2.4-12)$$

This vector contains all of the state derivatives for the wind-axes state equations.

Now, if the vector ω_W is assumed to have components P_W , Q_W , and R_W , (2.4-10) expands to

$$\begin{aligned} m\dot{V}_T &= F_T \cos \alpha \cos \beta - D + mg_1 \\ m\dot{\beta} V_T &= -F_T \cos \alpha \sin \beta + Y - mV_T R_W + mg_2 \\ m\dot{\alpha} V_T \cos \beta &= -F_T \sin \alpha - L + mV_T Q_W + mg_3, \end{aligned} \quad (2.4-13)$$

where the components of the gravity vector are given by

$$\begin{aligned} g_1 &= g'_0(-\cos \alpha \cos \beta \sin \theta + \sin \beta \sin \phi \cos \theta + \sin \alpha \cos \beta \cos \phi \cos \theta) \\ g_2 &= g'_0(\cos \alpha \sin \beta \sin \theta + \cos \beta \sin \phi \cos \theta - \sin \alpha \sin \beta \cos \phi \cos \theta) \\ g_3 &= g'_0(\sin \alpha \sin \theta + \cos \alpha \cos \phi \cos \theta). \end{aligned} \quad (2.4-14)$$

Equations (2.4-13) are the wind-axes force equations. It is evident that their cross-product terms involving velocities and angular rates are simpler than those of the body-axes equations because we have aligned the P -component of angular velocity with the translational velocity vector. Note that when $\alpha = \beta = 0$, the first equation reduces to the equation of motion of a particle. The gravity components are now more complex because there is an additional coordinate rotation involved.

It is now apparent from the wind-axes force equations that if linear dependence on $\dot{\alpha}$ is present in the lift coefficient, the alpha-dot terms can be collected on the left-hand side of the lift equations. Similarly, the side-force

equation can be made explicit in beta-dot if beta-dot dependence is present in the side-force coefficient. The derivatives of alpha and beta would then be available for use in the moment equations if dependence on these rates is present (e.g., if C_{m_a} is nonzero). Therefore, explicit state equations can be derived by using the wind-axes force equations when this was difficult or impossible with the body-axes state equations.

Wind-Axes Moment Equations

We now turn to the moment equations in (1.5-4) and transform them in the same way as the force equations, premultiplying by the rotation matrix S and using the cross-product matrix Ω_R [equation (2.4-11)]. The steps are

$$\begin{aligned} \mathbf{T}_B &= J\dot{\omega}_B + \boldsymbol{\omega}_B \times J\boldsymbol{\omega}_B \\ S\mathbf{T}_B &= SJ(\dot{\boldsymbol{\omega}}_W + S^T\dot{\boldsymbol{\omega}}_W) + (S\boldsymbol{\omega}_B) \times (SJS^T\boldsymbol{\omega}_W) \\ \mathbf{T}_W &= (SJS^T)(SS^T)\boldsymbol{\omega}_W + (SJS^T)\dot{\boldsymbol{\omega}}_W + \boldsymbol{\omega}_W \times (SJS^T)\boldsymbol{\omega}_W \\ \mathbf{T}_W &= J_W\Omega_R\boldsymbol{\omega}_W + J_W\dot{\boldsymbol{\omega}}_W + \boldsymbol{\omega}_W \times J_W\boldsymbol{\omega}_W. \end{aligned} \quad (2.4-15)$$

Equation (2.4-15) is the moment equation expressed in wind axes. Note once again that the angular rate vector, $\boldsymbol{\omega}_W$, represents the body-axes rates transformed into wind axes as opposed to the absolute wind-axes rates. The first two terms on the right-hand side are equivalent to $J_W(\boldsymbol{\omega}_R \times \boldsymbol{\omega}_W + \dot{\boldsymbol{\omega}}_W)$, and according to the theorem of Coriolis, the term in parentheses represents the derivative of the wind-axes rates with respect to the body axes. The inertia matrix has been expressed in wind axes by the congruence transformation

$$J_W = SJS^T = S_\beta S_\alpha JS_\alpha^T S_\beta^T. \quad (2.4-16)$$

The transformation of the inertia matrix with respect to alpha only is quite commonly used by stability and control engineers, so we will show the result of that transformation here. If the inertia matrix (1.3-19b) for the symmetric aircraft is used, then multiplying out the innermost transformation in (2.4-16) gives the stability-axes inertia matrix

$$J_S = \begin{bmatrix} J'_x & 0 & -J'_{xz} \\ 0 & J'_y & 0 \\ -J'_{xz} & 0 & J'_z \end{bmatrix} \quad (2.4-17)$$

where

$$\begin{aligned} J'_x &= J_x \cos^2 \alpha + J_z \sin^2 \alpha - J_{xz} \sin 2\alpha \\ J'_y &= J_y \\ J'_z &= J_x \sin^2 \alpha + J_z \cos^2 \alpha + J_{xz} \sin 2\alpha \\ J'_{xz} &= \frac{1}{2}(J_x - J_z) \sin 2\alpha + J_{xz} \cos 2\alpha. \end{aligned}$$

Performing the β -transformation leads to the final result:

$$J_W = \begin{bmatrix} J'_x \cos^2 \beta + J'_y \sin^2 \beta & \frac{1}{2}(J'_y - J'_x) \sin 2\beta & -J'_{xz} \cos \beta \\ \frac{1}{2}(J'_y - J'_x) \sin 2\beta & J'_x \sin^2 \beta + J'_y \cos^2 \beta & J'_{xz} \sin \beta \\ -J'_{xz} \cos \beta & J'_{xz} \sin \beta & J'_z \end{bmatrix}. \quad (2.4-18)$$

The inverse of the inertia matrix, (2.4-18), is given by

$$J_W^{-1} = (SJS^T)^{-1} = SJ^{-1}S^T,$$

which is the same as (2.4-16) with J^{-1} replacing J . Therefore, to obtain the inverse, we need only replace the elements of J in (2.4-17) with the corresponding elements of J^{-1} from (1.3-19b). If this is done, we find that because of symmetry, equations (2.4-17) can be used unchanged if instead we make some simple replacements in (2.4-18). Thus we replace J'_x by J'_z/Γ , J'_y by $1/J'_y$, J'_z by J'_x/Γ , and J'_{xz} by $-J'_{xz}/\Gamma$, where Γ was defined in (1.3-19b). The result is

$$J_W^{-1} = \frac{1}{\Gamma} \begin{bmatrix} J'_z \cos^2 \beta + \left(\frac{\Gamma}{J'_y}\right) \sin^2 \beta & \frac{1}{2} \left(\frac{\Gamma}{J'_y} - J'_z\right) \sin 2\beta & J'_{xz} \cos \beta \\ \frac{1}{2} \left(\frac{\Gamma}{J'_y} - J'_z\right) \sin 2\beta & J'_z \sin^2 \beta + \frac{\Gamma}{J'_y} \cos^2 \beta & -J'_{xz} \sin \beta \\ J'_{xz} \cos \beta & -J'_{xz} \sin \beta & J'_x \end{bmatrix}, \quad (2.4-19)$$

with the primed quantities given by (2.4-17).

The wind-axes moment equations (2.4-15) can now readily be solved for $\dot{\omega}_W$:

$$\dot{\omega}_W = -\Omega_R \omega_W - J_W^{-1}(\omega_W \times J_W \omega_W) + J_W^{-1} \mathbf{T}_W. \quad (2.4-20)$$

This equation is more complex than the original body-axes equation and

offers no advantages for use in a nonlinear model. Therefore, a nonlinear model will typically use the body-axes moment equations combined with force equations in either wind or body axes. However, (2.4-20) will be useful to us in Section 2.5 as the starting point in our derivation of a linear model.

Decoupling of the Nonlinear Equations

It is evident that the wind-axes force equations (2.4-13) will be considerably simplified if either or both of the bank and sideslip angles are zero. The relevant flight conditions are level, non-sideslipping flight so that $\phi = 0$ and $\beta = 0$, level skidding flight ($\phi = 0$, $\beta \neq 0$), and coordinated turning flight ($\phi \neq 0$, $\beta = 0$). The case $\phi = 0$, $\beta = 0$ is the most important because it is the normal flight condition and because it leads to *decoupling* of the flat-Earth equations of motion.

Decoupling means that the equations of motion separate into two independent sets. One set describes the *longitudinal motion* (pitching, and translation in the x - z plane), and the other set describes the *lateral-directional motion* (rolling, and sideslipping and yawing) of the aircraft. The decoupled equations are very much easier to handle in analytical studies. Decoupling also occurs in the linear small-perturbation equations that we derive later, and the reduced order of the two sets greatly simplifies the study of the "dynamic modes" of an aircraft.

Consider first the condition $\phi = 0$, the g -component equations, (2.4-14), become

$$\begin{aligned} g_1 &= -g'_0 \cos \beta \sin(\theta - \alpha) \\ g_2 &= g'_0 \sin \beta \sin(\theta - \alpha) \\ g_3 &= g'_0 \cos(\theta - \alpha). \end{aligned} \quad (2.4-21)$$

Next, apply the additional constraint $\beta = 0$. The pitch attitude angle and the angle of attack then lie in the same vertical plane above the north-east plane, and their difference is the *flight-path angle*, γ . That is,

$$\gamma|_{\phi=\beta=0} = \theta - \alpha \quad (2.4-22)$$

(the general expression for γ is given in (3.4-2)). Under these conditions the force equations (2.4-13) reduce to

$$\begin{aligned} m\dot{V}_T &= F_T \cos \alpha - D - mg'_0 \sin \gamma \\ m\dot{\beta}V_T &= Y - mV_T R_W \\ m\dot{\alpha}V_T &= -F_T \sin \alpha - L + mV_T Q_W + mg'_0 \cos \gamma. \end{aligned} \quad (2.4-23)$$

We can now see that the longitudinal (first and third) equations have become

independent of the lateral-directional variables (β, ϕ, ψ, P, R). These equations provide a commonly used model for longitudinal motion only.

It can be seen from the kinematic equations (2.4-3) that when the bank angle is zero,

$$\dot{\theta} = Q. \quad (2.4-24)$$

Then because $\beta = 0$, $Q_w = Q$, so the third equation of (2.4-23) is often written as

$$m\dot{y}V_T = F_T \sin \alpha + L - mg'_0 \cos \gamma. \quad (2.4-25)$$

Next consider the expanded body-axes moment equations (1.3-21). It is easy to see that if $P \equiv R \equiv 0$, the pitching moment equation is not coupled to the rolling and yawing moment equations. This yields the final equation for the decoupled longitudinal motion

$$J_y \dot{Q} = M. \quad (2.4-26)$$

The remaining moment equations are not particularly useful because integration of any roll and yaw accelerations will soon produce finite roll and yaw rates, and the equations will become coupled once more. Thus our main achievement in this subsection has been to derive a nonlinear model for longitudinal-only motion. In addition, the notion of decoupling has been introduced, and this will take on much greater importance in connection with linear models in the next section.

2.5 LINEAR MODELS AND THE STABILITY DERIVATIVES

When we perform a computer simulation to evaluate the performance of an aircraft with its control systems, we shall almost invariably use a nonlinear model. Also, the linear equations needed for control system design will mostly be derived by numerical methods from the nonlinear computer model. Because the nonlinear state models are difficult to handle without the use of a digital computer, most of the early progress in understanding the dynamics of aircraft and the stability of the motion came from studying linear small-perturbation equations. G. H. Bryan [Bryan, 1911] introduced the idea of perturbed forces and moments with respect to a *steady-state flight condition*, and this approach is still in use. The small-perturbation equations are linear equations derived algebraically from nonlinear equations like those of Section 2.4. In these equations the nonlinear aerodynamic coefficients are replaced by terms involving the stability derivatives described briefly in Section 2.4.

There are two good reasons, apart from their historical importance, for algebraically deriving the small perturbation equations. First, the stability

derivatives needed for the linear equations can be estimated relatively quickly [Hoak et al., 1970] before nonlinear aerodynamic data become available. Second, the algebraic small-perturbation equations provide a great deal of insight into the relative importance of the various stability derivatives under different flight conditions and their effect on the stability of the aircraft motion. In preparation for deriving the linear equations we now examine the concept of a steady-state flight condition.

Singular Points and Steady-State Flight

In the preceding section, when the body-axes force equations were used, alpha-dot or beta-dot force dependence created a difficulty in that the state equations become implicit in the derivatives of the states alpha and beta. This problem was solved in an ad hoc manner by using the wind-axes equations and collecting linear alpha-dot or beta-dot terms on one side of the equations. In this section where the goal is to derive linear equations algebraically, we take a more general approach, starting with *implicit state equations* in the general form

$$\mathbf{f}(\dot{\mathbf{X}}, \mathbf{X}, \mathbf{U}) = 0, \quad (2.5-1)$$

where \mathbf{f} is a vector of n scalar nonlinear functions f_i .

In the theory of nonlinear systems [Vidyasagar, 1978] the concept of a *singular point*, or *equilibrium point*, of an autonomous (no external control inputs) time-invariant system is introduced. The coordinates of the singular point(s) of the implicit nonlinear state equations are given by the solution vector, $\mathbf{X} = \mathbf{X}_e$, which satisfies

$$\mathbf{f}(\dot{\mathbf{X}}, \mathbf{X}, \mathbf{U}) = 0, \quad \text{with } \dot{\mathbf{X}} \equiv 0; \mathbf{U} \equiv 0 \text{ or constant.} \quad (2.5-2)$$

This idea has strong intuitive appeal; the system is “at rest” when all of the derivatives are identically zero, and then one may examine the behavior of the system near the singular point by slightly perturbing some of the variables. If, in the case of an aircraft model, the state trajectory departs rapidly from the singular point in response to a small perturbation in, say, pitch attitude, the human pilot is unlikely to be able to control this aircraft.

Steady-state aircraft flight can be defined as a condition in which all of the motion variables are constant or zero. That is, the linear and angular velocity components are constant (or zero), and all acceleration components are zero. This definition is very restrictive unless some simplifying assumptions are made; for example, it must be assumed that the aircraft mass remains constant. In the case of the round-Earth equations, because of the Earth’s angular velocity, only minor-circle (constant-latitude) flight around the Earth is a true steady-state condition. When the oblateness is also taken into

account, minor circles (including the major circle around the Equator) are the only trajectories along which gravity remains constant in magnitude.

Assuming that the flat-Earth equations are satisfactory for all of our control system design purposes, the definition allows steady wings-level flight and steady turning flight. Furthermore, if the change in atmospheric density with altitude is neglected, a wings-level climb and a climbing turn are permitted as steady-state flight conditions. In this case the NED position equations do not couple back into the equations of motion and need not be used in finding a steady-state condition. Therefore, the steady-state conditions that are important to us for control system design can be defined in terms of the remaining nine state variables of the flat-Earth equations as follows:

Steady-State Flight:

$$\dot{P}, \dot{Q}, \dot{R}, \dot{U}, \dot{V}, \dot{W} \text{ (or } \dot{V}_T, \dot{\beta}, \dot{\alpha}) \equiv 0, \quad U = \text{const.} \quad (2.5-3a)$$

with the following additional constraints according to the flight condition:

$$\text{Steady Wings-Level Flight: } \phi, \dot{\phi}, \dot{\theta}, \dot{\psi} \equiv 0 \quad (\therefore P, Q, R \equiv 0)$$

$$\text{Steady Turning Flight: } \dot{\phi}, \dot{\theta} \equiv 0, \quad \dot{\psi} \equiv \text{turn rate}$$

$$\text{Steady Pull-up: } \phi, \dot{\phi}, \dot{\psi} \equiv 0, \quad \dot{\theta} \equiv \text{pull-up rate}$$

$$\text{Steady Roll: } \dot{\theta}, \dot{\psi} \equiv 0, \quad \dot{\phi} \equiv \text{roll rate}$$

(2.5-3b)

The steady-state conditions $\dot{P}, \dot{Q}, \dot{R} \equiv 0$ require the angular rates to be zero or constant (as in steady turns), and therefore the aerodynamic and thrust moments must be zero or constant. The conditions $\dot{U}, \dot{V}, \dot{W} \equiv 0$ require the airspeed, angle of attack, and sideslip angle to be constant, and hence the aerodynamic forces must be zero or constant. Therefore, the steady-state pull-up (or push-over) and steady-state roll conditions can only exist instantaneously. However, it is useful to be able to linearize the aircraft dynamics in these flight conditions since the control systems must operate there.

While a pilot may not find it very difficult to put an aircraft into a steady-state flight condition, the mathematical model requires the solution of the simultaneous nonlinear equations (2.5-2). In general, because of the nonlinearity, a steady-state solution can only be found by using a numerical method on a digital computer. Multiple solutions may exist, and a feasible solution will emerge only when practical constraints are placed on the variables. We consider this problem in Chapter 3 and assume here that a solution $\mathbf{X}_e, \mathbf{U}_e$ is known for the desired flight condition.

Linearization

The implicit nonlinear equation will be written as

$$\begin{aligned} f_1(\dot{\mathbf{X}}, \mathbf{X}, \mathbf{U}) &= 0 \\ f_2(\dot{\mathbf{X}}, \mathbf{X}, \mathbf{U}) &= 0 \\ &\vdots \\ f_9(\dot{\mathbf{X}}, \mathbf{X}, \mathbf{U}) &= 0, \end{aligned} \quad (2.5-4)$$

and will be obtained, respectively, from the (nine) wind-axes force equations, kinematic equations, and moment equations by moving all nonzero terms to the right-hand side of the equations. The reduced state vector is

$$\mathbf{X}^T = [V_T \quad \beta \quad \alpha \quad \phi \quad \theta \quad \psi \quad P_W \quad Q_W \quad R_W]. \quad (2.5-5a)$$

The control vector was given in (2.4-7) and is repeated here:

$$\mathbf{U}^T = [thl \quad el \quad ail \quad rdr]. \quad (2.5-5b)$$

We now consider small perturbations from the steady-state condition $\mathbf{X}_e, \mathbf{U}_e$ and derive a set of linear constant-coefficient state equations. If we expand the nonlinear state equations (2.5-4) in a Taylor series about the equilibrium point $(\mathbf{X}_e, \mathbf{U}_e)$, and keep only the first-order terms, we find that the perturbations in the state, state derivative, and control vectors must satisfy

$$\begin{aligned} \nabla_{\dot{\mathbf{X}}} f_1 \delta \dot{\mathbf{X}} + \nabla_{\mathbf{X}} f_1 \delta \mathbf{X} + \nabla_U f_1 \delta \mathbf{U} &= 0 \\ &\vdots \\ \nabla_{\dot{\mathbf{X}}} f_9 \delta \dot{\mathbf{X}} + \nabla_{\mathbf{X}} f_9 \delta \mathbf{X} + \nabla_U f_9 \delta \mathbf{U} &= 0. \end{aligned} \quad (2.5-6)$$

In this equation ∇ (nabla) represents a row vector of first partial derivative operators, for example

$$\nabla_{\mathbf{X}} f_i = \left[\frac{\partial f_i}{\partial X_1} \quad \frac{\partial f_i}{\partial X_2} \quad \cdots \quad \frac{\partial f_i}{\partial X_n} \right].$$

Each term in (2.5-6) is a scalar product; thus $\nabla_{\mathbf{X}} f_1 \delta \mathbf{X}$ is the total differential of f_1 due to simultaneous perturbations in all the elements of the state vector.

Equations (2.5-6) can now be written in implicit linear state-variable form as

$$E \dot{\mathbf{x}} = A \mathbf{x} + B \mathbf{u}. \quad (2.5-7)$$

Lowercase notation has been used to indicate that \mathbf{x} and \mathbf{u} are perturbations

from the equilibrium values of the state and control vectors. The coefficient matrices

$$E = - \begin{bmatrix} \nabla_{\dot{X}} f_1 \\ \vdots \\ \nabla_{\dot{X}} f_9 \end{bmatrix}_{\substack{\mathbf{X} = \mathbf{X}_e \\ \mathbf{U} = \mathbf{U}_e}}, \quad A = \begin{bmatrix} \nabla_X f_1 \\ \vdots \\ \nabla_X f_9 \end{bmatrix}_{\substack{\mathbf{X} = \mathbf{X}_e \\ \mathbf{U} = \mathbf{U}_e}}, \quad B = \begin{bmatrix} \nabla_U f_1 \\ \vdots \\ \nabla_U f_9 \end{bmatrix}_{\substack{\mathbf{X} = \mathbf{X}_e \\ \mathbf{U} = \mathbf{U}_e}}, \quad (2.5-8)$$

are called *Jacobian matrices* and must be calculated at the equilibrium point. If E is nonsingular (2.5-7) can be rewritten as an explicit set of linear state equations, but we shall see later this is not necessarily the most convenient way to use the implicit state equations.

The Jacobian matrices E, A, B will be evaluated three rows at a time, corresponding, respectively, to the wind-axes force equations (f_1 to f_3), kinematic equations (f_4 to f_6), and moment equations (f_7 to f_9). The evaluation will be for the steady, level flight condition, with the additional constraint of no sideslip ($\beta = 0$). The latter condition greatly simplifies the algebra involved in the linearization and leads to "lat-long" decoupling. Therefore, the equilibrium (steady-state) conditions are

Steady-State Conditions: $\beta, \phi, P, Q, R \equiv 0$

$$\text{all derivatives} \equiv 0. \quad (2.5-9)$$

The algebra can be further reduced by taking advantage of some features of the equations. Thus, when differentiating products containing $\cos \beta$ or $\cos \phi$, all of the resulting $\sin \beta$ or $\sin \phi$ terms will disappear when we apply the $\beta = 0$ and $\phi = 0$ equilibrium conditions. Therefore, the $\cos \beta$ or $\cos \phi$ terms can be set to unity before differentiation. Similarly, a $\cos \beta$ or $\cos \phi$ in the denominator of a quotient term can be set to unity. Also, if two or more terms with equilibrium values of zero (e.g., $\sin \beta, \sin \phi$) occur in a product term, this product can be discarded before differentiation.

The Linearized Force Equations

The first three rows of the linear equations (2.5-7) will now be obtained by performing the gradient operations, shown in (2.5-8), on the nonlinear force equations (2.4-13). All of the terms in (2.4-13) will be moved to the right-hand side of the equations. First, we find the partial derivatives with respect to \dot{X} and use the steady-state condition (2.5-9); this gives

$$- \begin{bmatrix} \nabla_{\dot{X}} f_1 \\ \nabla_{\dot{X}} f_2 \\ \nabla_{\dot{X}} f_3 \end{bmatrix}_{\mathbf{X} = \mathbf{X}_e} = \begin{bmatrix} m \nabla_{\dot{X}} V_T + \nabla_{\dot{X}} D \\ m V_T \nabla_{\dot{X}} \dot{\beta} - \nabla_{\dot{X}} Y \\ m V_T \nabla_{\dot{X}} \dot{\alpha} + \nabla_{\dot{X}} L \end{bmatrix}. \quad (2.5-10)$$

A term such as $\nabla_X \dot{V}_T$ is simply a row vector with unity in the position corresponding to the \dot{V}_T state derivative, and zeros elsewhere. The other terms, such as $\nabla_X L$, are row vectors containing all of the partial derivatives of the forces with respect to the state derivatives. The thrust has been assumed to be independent of the state derivatives, and a number of other partial derivatives will also be neglected.

The partial derivatives of the aerodynamic forces and moments with respect to other variables are the so-called *aerodynamic derivatives*. Table 2.5-1 lists the derivatives that are normally significant in the force equations and defines their symbols. These derivatives are called the *dimensional derivatives*, and later we shall introduce a related set of derivatives that have been made dimensionless in the same way that the aerodynamic coefficients are made dimensionless. The dimensional derivatives are given the symbols X , Y , and Z , to indicate which force component is involved (the symbols D , Y , and L are also used). Their subscripts indicate the quantity with respect to which the derivative is taken (subscripts for the controls were defined in Equation (2.4-7)).

For the purpose of deriving the linear equations, only the derivatives shown in the table will be assumed to be nonzero. Therefore, the terms $\nabla_X D$ and $\nabla_X Y$ in (2.5-10) will now be dropped, and additional terms will be dropped later. Note that the forces involved in the partial derivatives are wind-axes forces, except for the engine thrust F_T . This force belongs naturally to the aircraft-body axes, and it only appears in wind-axes equations in conjunction with (resolved by) trigonometric functions of the aerodynamic angles.

TABLE 2.5-1. The Force Dimensional Derivatives (Wind Axes)

X-Axis	Y-Axis	Z-Axis
$X_V = -\frac{1}{m} \frac{\partial D}{\partial V_T}$	$Y_\beta = \frac{1}{m} \left(\frac{\partial Y}{\partial \beta} - D \right)$	$Z_V = -\frac{1}{m} \frac{\partial L}{\partial V_T}$
$X_{T_V} = \frac{1}{m} \frac{\partial F_T}{\partial V_T}$	$Y_p = \frac{1}{m} \frac{\partial Y}{\partial P_W}$	$Z_\alpha = -\frac{1}{m} \left(D + \frac{\partial L}{\partial \alpha} \right)$
$X_\alpha = \frac{1}{m} \left(L - \frac{\partial D}{\partial \alpha} \right)$	$Y_r = \frac{1}{m} \frac{\partial Y}{\partial R_W}$	$Z_\dot{\alpha} = -\frac{1}{m} \frac{\partial L}{\partial \dot{\alpha}}$
$X_{\delta e} = -\frac{1}{m} \frac{\partial D}{\partial \delta e}$	$Y_{\delta r} = \frac{1}{m} \frac{\partial Y}{\partial \delta r}$	$Z_q = -\frac{1}{m} \frac{\partial L}{\partial Q_W}$
$X_{\delta th} = \frac{1}{m} \frac{\partial F_T}{\partial \delta thl}$	$Y_{\delta a} = \frac{1}{m} \frac{\partial Y}{\partial \delta ail}$	$Z_{\delta e} = -\frac{1}{m} \frac{\partial L}{\partial \delta el}$

We shall interpret (2.5-10) in terms of the dimensional derivatives. The vector $\nabla_{\dot{X}} L$ contains only the derivative $Z_{\dot{\alpha}}$ (multiplied by m) in the $\dot{\alpha}$ position, so (2.5-10) can now be rewritten as

$$-\begin{bmatrix} \nabla_{\dot{X}} f_1 \\ \nabla_{\dot{X}} f_2 \\ \nabla_{\dot{X}} f_3 \end{bmatrix}_{\mathbf{X}=\mathbf{X}_e} = m \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & V_T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & V_T - Z_{\dot{\alpha}} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (2.5-11)$$

Next, using (2.4-13), form the partial derivatives with respect to \mathbf{X} and apply the steady-state conditions (2.5-9). The result is

$$\begin{aligned} & \begin{bmatrix} \nabla_X f_1 \\ \nabla_X f_2 \\ \nabla_X f_3 \end{bmatrix}_{\mathbf{X}=\mathbf{X}_e} \\ &= \begin{bmatrix} -F_T \sin \alpha \nabla_X \alpha + \cos \alpha \nabla_X F_T - \nabla_X D - mg'_0 \cos \gamma (\nabla_X \theta - \nabla_X \alpha) \\ -F_T \cos \alpha \nabla_X \beta + \nabla_X Y - mV_T \nabla_X R_W + mg'_0 (\sin \gamma \nabla_X \beta + \cos \theta \nabla_X \phi) \\ -F_T \cos \alpha \nabla_X \alpha - \sin \alpha \nabla_X F_T - \nabla_X L + mV_T \nabla_X Q_W + mg'_0 (\sin \gamma \nabla_X \alpha - \sin \gamma \nabla_X \theta) \end{bmatrix}. \end{aligned} \quad (2.5-12)$$

This result can be further reduced by using (2.4-23), in steady-state form [conditions (2.5-9) applied], to replace some groups of terms by the steady-state lift and drag forces. Thus the partial derivatives evaluated at the equilibrium point are

$$\begin{aligned} & \begin{bmatrix} \nabla_X f_1 \\ \nabla_X f_2 \\ \nabla_X f_3 \end{bmatrix}_{\mathbf{U}=\mathbf{U}_e} \\ &= \begin{bmatrix} \cos \alpha_e \nabla_X F_T + L \nabla_X \alpha - \nabla_X D - mg'_0 \cos \gamma_e \nabla_X \theta \\ \nabla_X Y - D \nabla_X \beta - mV_T \nabla_X R_W + mg'_0 \cos \theta_e \nabla_X \phi \\ -\sin \alpha_e \nabla_X F_T - \nabla_X L - D \nabla_X \alpha + mV_T \nabla_X Q_W - mg'_0 \sin \gamma_e \nabla_X \theta \end{bmatrix}, \end{aligned} \quad (2.5-13)$$

where α_e , θ_e , γ_e , and the lift, L , and drag, D , are steady-state values. Note that there is no steady-state sideforce. Now, if this expression is interpreted in terms of the derivatives from Table 2.5-1, we obtain for the right-hand side

$$m \begin{bmatrix} X_V + X_{T_V} \cos \alpha_e & 0 & X_{\alpha} & 0 & -g'_0 \cos \gamma_e & 0 & 0 & 0 & 0 \\ 0 & Y_{\beta} & 0 & g'_0 \cos \theta_e & 0 & 0 & Y_p & 0 & Y_r - V_T \\ Z_V - X_{T_V} \sin \alpha_e & 0 & Z_{\alpha} & 0 & -g'_0 \sin \gamma_e & 0 & 0 & V_T + Z_q & 0 \end{bmatrix}. \quad (2.5-14)$$

This matrix constitutes the top three rows of A in (2.5-7).

It only remains to obtain the partial derivatives of the force equations with respect to the control vector \mathbf{U} . The partial derivatives are

$$\begin{bmatrix} \nabla_U f_1 \\ \nabla_U f_2 \\ \nabla_U f_3 \end{bmatrix} = \begin{bmatrix} \cos \alpha \nabla_U F_T - \nabla_U D \\ \nabla_U Y \\ -\sin \alpha \nabla_U F_T - \nabla_U L \end{bmatrix}. \quad (2.5-15)$$

Now, inserting the relevant dimensional derivatives and the equilibrium values of the angles, we obtain

$$\begin{bmatrix} \nabla_U f_1 \\ \nabla_U f_2 \\ \nabla_U f_3 \end{bmatrix}_{\substack{\mathbf{X}=\mathbf{X}_e \\ \mathbf{U}=\mathbf{U}_e}} = m \begin{bmatrix} X_{\delta_{th}} \cos \alpha_e & X_{\delta_e} & 0 & 0 \\ 0 & 0 & Y_{\delta_a} & Y_{\delta_r} \\ -X_{\delta_{th}} \sin \alpha_e & Z_{\delta_e} & 0 & 0 \end{bmatrix}, \quad (2.5-16)$$

and these are the top three rows of B in (2.5-7).

This completes the linearization of the force equations. Note that the positions of the zero elements correspond to the beginnings of the anticipated decoupling in (2.5-7). One of the assumptions contributing to this decoupling is that the partial derivatives of drag with respect to the lateral/directional controls (ailerons and rudder) can be neglected. In practice aileron and rudder deflections do cause nonnegligible changes in drag, but this assumption does not have any significant consequences on the linearized dynamics.

The Linearized Kinematic Equations

We shall now determine the second block of three rows in (2.5-7). The nonlinear kinematic relationship between the Euler-angle rates and the wind-axes rates P_w , Q_w , and R_w is obtained from (1.4-15) or (2.4-3) and the transformation matrices S_α , S_β . Thus

$$\dot{\Phi} = \mathcal{E}(\Phi) S_\alpha^T S_\beta^T \omega_w. \quad (2.5-17)$$

There are no aerodynamic forces or moments involved in these equations, and it is easy to see that the contribution to the E matrix is given by

$$-\begin{bmatrix} \nabla_X f_4 \\ \nabla_X f_5 \\ \nabla_X f_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (2.5-18)$$

Next, we determine the contributions of the kinematic equations to the A matrix. Equations (2.5-17) are linear in P_w , Q_w , and R_w , so all partial derivatives of the coefficient matrix elements will be eliminated when we set

$P_W = Q_W = R_W = 0$. It only remains to evaluate the coefficient matrices under the steady-state conditions. The transformation S_β then reduces to the identity matrix, and the product of the two remaining matrices gives

$$\mathcal{E}(\Phi)S_\alpha^T = \begin{bmatrix} \cos \alpha + \tan \theta \cos \phi \sin \alpha & \tan \theta \sin \phi & -\sin \alpha + \tan \phi \cos \phi \cos \alpha \\ -\sin \phi \sin \alpha & \cos \phi & -\sin \phi \cos \alpha \\ \cos \phi \sin \alpha & \sin \phi & \cos \phi \cos \alpha \\ \hline \cos \theta & \cos \theta & \cos \theta \end{bmatrix}. \quad (2.5-19)$$

Inserting the steady-state conditions in this matrix and applying some trigonometric identities, we see that

$$\begin{bmatrix} \nabla_X f_4 \\ \nabla_X f_5 \\ \nabla_X f_6 \end{bmatrix}_{\substack{\mathbf{x}=\mathbf{x}_e \\ \mathbf{U}=\mathbf{U}_e}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{\cos \gamma_e}{\cos \theta_e} & 0 & \frac{\sin \gamma_e}{\cos \theta_e} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sin \alpha_e}{\cos \theta_e} & 0 & \frac{\cos \alpha_e}{\cos \theta_e} \end{bmatrix}. \quad (2.5-20)$$

The partial derivatives of the kinematic variables with respect to the control vector are all zero, so that this completes the linearization of the kinematic equations. Note that the force and moment equations (2.4-2) and (2.4-4) are independent of the heading angle ψ in the NED geographic frame, so the third kinematic equation is not really needed in the linear model.

The Linearized Moment Equations

Here we determine the last three rows of the linear state equations (2.5-7). The starting point for this linearization is the wind-axes moment equations (2.4-20), with all terms moved to the right-hand side. The moment partial derivatives that are normally considered important are contained in Table 2.5-2; the table defines the moment dimensional derivatives. These dimensional derivatives are given the symbols L , M , and N , to correspond, respectively, to rolling, pitching, and yawing moments, and their subscripts indicate the quantity with respect to which the derivative is taken. The derivatives with respect to beta-dot have been omitted from the table because they are usually unimportant and are difficult to measure. The effect of beta-dot on yawing moment may sometimes be important and the derivative can be estimated with methods given in the USAF DATCOM [Hoak et al., 1970].

We assumed in Section 2.4 that the engine thrust vector was parallel to the body x -axis, and contributed force and moment components F_T and M_T in

TABLE 2.5-2. The Moment Dimensional Derivatives (Wind Axes)

Roll	Pitch	Yaw
$L_\beta = \frac{1}{J'_X} \frac{\partial \bar{L}_{W_A}}{\partial \beta}$	$M_\nu = \frac{1}{J'_Y} \frac{\partial M_{W_A}}{\partial V_T}$	$N_\beta = \frac{1}{J'_Z} \frac{\partial N_{W_A}}{\partial \beta}$
$L_p = \frac{1}{J'_X} \frac{\partial \bar{L}_{W_A}}{\partial P_W}$	$M_\alpha = \frac{1}{J'_Y} \frac{\partial M_{W_A}}{\partial \alpha}$	$N_p = \frac{1}{J'_Z} \frac{\partial N_{W_A}}{\partial P_W}$
$L_r = \frac{1}{J'_X} \frac{\partial \bar{L}_{W_A}}{\partial R_W}$	$M_\dot{\alpha} = \frac{1}{J'_Y} \frac{\partial M_{W_A}}{\partial \dot{\alpha}}$	$N_r = \frac{1}{J'_Z} \frac{\partial N_{W_A}}{\partial R_W}$
$L_{\delta a} = \frac{1}{J'_X} \frac{\partial \bar{L}_{W_A}}{\partial a il}$	$M_q = \frac{1}{J'_Y} \frac{\partial M_{W_A}}{\partial Q_W}$	$N_{\delta a} = \frac{1}{J'_Z} \frac{\partial N_{W_A}}{\partial a il}$
$L_{\delta r} = \frac{1}{J'_X} \frac{\partial \bar{L}_{W_A}}{\partial r dr}$	$M_{\delta e} = \frac{1}{J'_Y} \frac{\partial M_{W_A}}{\partial e l}$	$N_{\delta r} = \frac{1}{J'_Z} \frac{\partial N_{W_A}}{\partial r dr}$
	$M_{T_\nu} = \frac{1}{J'_Y} \frac{\partial M_{W_T}}{\partial V_T}$	$N_{T_\beta} = \frac{1}{J'_Z} \frac{\partial N_{W_T}}{\partial \beta}$
	$M_{T_\alpha} = \frac{1}{J'_Y} \frac{\partial M_{W_T}}{\partial \alpha}$	
	$M_{\delta th} = \frac{1}{J'_Y} \frac{\partial M_{W_A}}{\partial thl}$	

body axes. Therefore, in the wind-axes moment equations, the thrust effects are given by (see Table 2.3-1)

$$\bar{L}_W = \bar{L}_{W_A} + M_T \sin \beta \quad (2.5-21)$$

$$M_W = M_{W_A} + M_T \cos \beta.$$

We shall consider only the variation of thrust with speed, and the derivative $\partial M_W / \partial V_T$ is broken down into two components, M_ν (aerodynamic only) and M_{T_ν} , as shown in Table 2.5-2. The table shows two other thrust derivatives (M_{T_α} and N_{T_β}) that are sometimes significant, but for simplicity, they will be excluded from our equations.

To find the block of E -matrix terms, all the moment-equation terms that involve state derivatives must be examined. These terms are $-\dot{\omega}_W$ and

$-\Omega_R \omega_W$, and the expansion of the latter term is

$$\Omega_R \omega_W = \begin{bmatrix} -\dot{\beta} Q_W - \dot{\alpha} R_W \cos \beta \\ \dot{\beta} P_W + \dot{\alpha} R_W \sin \beta \\ \dot{\alpha} P_W \cos \beta - \dot{\alpha} Q_W \sin \beta \end{bmatrix}. \quad (2.5-22)$$

In this term the coefficients of the state derivatives are the rates P_W , Q_W , and R_W , so there is no contribution to the linearized equations when the steady-state nonturning flight conditions $P_W = Q_W = R_W = 0$ are applied. The remaining term $\dot{\omega}_W$ contributes only an identity block to the E matrix in the P , Q , and R columns.

Next, we must consider the aerodynamic moments that have a partial derivative with respect to the state derivatives. As stated earlier, the beta-dot derivatives will be neglected, and only the dependence of M_W on alpha-dot will be considered. The moment equation is linear in the aerodynamic moments, and the coefficients of M_W are obtained from the middle column of J_W^{-1} [given by (2.4-19)]. The contribution to the E matrix is found simply by inserting the condition $\beta = 0$ into these coefficients; thus

$$\frac{1}{\Gamma} \begin{bmatrix} \frac{1}{2} \left(\frac{\Gamma}{J'_y} - J'_z \right) \sin 2\beta \\ J'_z \sin^2 \beta + \frac{\Gamma}{J'_y} \cos^2 \beta \\ -J'_{xz} \sin \beta \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ \frac{1}{J_y} \\ 0 \end{bmatrix}. \quad (2.5-23)$$

The term that this contributes to the E matrix is denoted by $M_{\dot{\alpha}}$ as shown in Table 2.5-2. Note that as shown by (2.4-17), J_y is identical to J'_y . Therefore, the last three rows of the E matrix are given by

$$-\begin{bmatrix} \nabla_{\dot{X}} f_7 \\ \nabla_{\dot{X}} f_8 \\ \nabla_{\dot{X}} f_9 \end{bmatrix}_{\mathbf{U}=\mathbf{U}_e}^{\mathbf{x}=\mathbf{x}_e} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -M_{\dot{\alpha}} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.5-24)$$

The contributions of the moment equations to the A and B matrices must now be found. The term $\Omega_R \omega_W$, given by (2.5-22), is of degree 2 or higher in the variables of the steady-state condition (2.5-9). Therefore, this term will contribute nothing to the A and B matrices. The next term in (2.4-20), $J_W^{-1}(\omega_W \times J_W \omega_W)$, is of degree 2 in the variables P_W , Q_W , R_W (see the previous expansions of the moment equations) and can also be discarded. The last term is $J_W^{-1} \mathbf{T}_W$ in which the aerodynamic and thrust moments occur (see Table 2.3-1). Note that all three components of \mathbf{T}_W must be zero in the

steady-state condition, so that terms obtained by differentiating J_W^{-1} will vanish in the steady state. Also, the term is linear with respect to the moments, so the coefficients of the moment derivatives are obtained by substituting the steady-state condition $\beta = 0$ in J_W^{-1} . Therefore, using (2.4-19), the last three rows of the A and B matrices are given by

$$\begin{bmatrix} \nabla_X f_7 \\ \nabla_X f_8 \\ \nabla_X f_9 \end{bmatrix}_{\beta=0} = \begin{bmatrix} \frac{J'_Z \nabla_X \bar{L}_W + J'_{XZ} \nabla_X N_W}{\Gamma} \\ \frac{\nabla_X M_W}{J'_Y} \\ \frac{J'_{XZ} \nabla_X \bar{L}_W + J'_X \nabla_X N_W}{\Gamma} \end{bmatrix} \quad (2.5-25)$$

and

$$\begin{bmatrix} \nabla_U f_7 \\ \nabla_U f_8 \\ \nabla_U f_9 \end{bmatrix}_{\beta=0} = \begin{bmatrix} \frac{J'_Z \nabla_U \bar{L}_W + J'_{XZ} \nabla_U N_W}{\Gamma} \\ \frac{\nabla_U M_W}{J'_Y} \\ \frac{J'_{XZ} \nabla_U \bar{L}_W + J'_X \nabla_U N_W}{\Gamma} \end{bmatrix}. \quad (2.5-26)$$

When the partial derivatives (2.5-25) and (2.5-26) are interpreted in terms of the dimensional derivatives in Table 2.5-2, we obtain the last three rows of the A matrix:

$$\begin{aligned} & \begin{bmatrix} \nabla_X f_7 \\ \nabla_X f_8 \\ \nabla_X f_9 \end{bmatrix}_{\substack{\mathbf{X}=\mathbf{X}_e \\ \mathbf{U}=\mathbf{U}_e}} \\ &= \begin{bmatrix} 0 & \mu L_\beta + \sigma N_\beta + \nu M_T & 0 & 0 & 0 & 0 & \mu L_p + \sigma N_p & 0 & \mu L_r + \sigma N_r \\ M_V + M_{T_V} & 0 & M_\alpha & 0 & 0 & 0 & 0 & M_q & 0 \\ 0 & \mu N_\beta + \sigma L_\beta & 0 & 0 & 0 & 0 & \mu N_p + \sigma L_p & 0 & \mu N_r + \sigma L_r \end{bmatrix}, \end{aligned} \quad (2.5-27)$$

and the last three rows of the B matrix,

$$\begin{bmatrix} \nabla_U f_7 \\ \nabla_U f_8 \\ \nabla_U f_9 \end{bmatrix}_{\substack{\mathbf{X}=\mathbf{X}_e \\ \mathbf{U}=\mathbf{U}_e}} = \begin{bmatrix} 0 & 0 & \mu L_{\delta a} + \sigma N_{\delta a} & \mu L_{\delta r} + \sigma N_{\delta r} \\ M_{\delta h} & M_{\delta e} & 0 & 0 \\ 0 & 0 & \mu N_{\delta a} + \sigma L_{\delta a} & \mu N_{\delta r} + \sigma L_{\delta r} \end{bmatrix}. \quad (2.5-28)$$

In these equations the constants μ , σ , and ν are given by

$$\mu = \frac{J'_Z J'_X}{\Gamma}, \quad \sigma = \frac{J'_Z J'_{XZ}}{\Gamma}, \quad \nu = \frac{J'_Z}{\Gamma}. \quad (2.5-29)$$

The value of the cross-product of inertia J'_{xz} , relative to J'_X , is usually such that the parameter μ is quite close to unity, and σ is much smaller. The parameter ν is not dimensionless and is approximately equal to $1/J'_X$. All of the terms in (2.5-27) are dimensional derivatives except for the thrust moment M_T (the limitations of the notation are apparent here), and this thrust moment has introduced some asymmetry and coupling into the equations.

Other components of thrust can be added to the nonlinear equations (2.4-13) if necessary, and additional thrust terms will appear in the linear equations. For example, if the effect of unbalanced power is to be analyzed, it is necessary to include a yawing moment due to engine thrust in the nonlinear equations. The resulting linear equations then become very cumbersome. We shall not include these additional thrust terms, because for design purposes, we shall obtain linear equations by numerical (rather than algebraic) linearization of the nonlinear 6-DOF equations (see Chapter 3). The algebraic linearization will be used mainly for providing insight into the aircraft dynamic modes.

The Decoupled Linear State Equations

All of the information for the coefficient matrices of the linear state equations (2.5-7) has now been obtained. An inspection of the coefficient blocks shows that the longitudinal and lateral-directional equations are decoupled (although the lateral-directional equations do depend on steady-state longitudinal quantities such as M_T , γ_e , and θ_e). Therefore, rather than attempt to assemble the complete equations, we shall collect the longitudinal- and lateral-directional equations separately. The thrust moment will be retained in the longitudinal equations, and its effect will be accounted for when we obtain linear equations for an aircraft model in Chapter 3. For simplicity the effect will be dropped from the lateral-directional equations.

The longitudinal states and controls are

$$\mathbf{x}^T = [v_T \quad \alpha \quad \theta \quad q], \quad \mathbf{u}^T = [\delta_{th} \quad \delta_e], \quad (2.5-30)$$

and these variables are perturbations from the steady-state values V_T , α_e , θ_e , Q , thl, and el. The longitudinal equations are obtained from the first and last rows of (2.5-11), (2.5-14), and (2.5-16) (divided through by m), the middle

rows of (2.5-18) and (2.5-20), and the middle rows of (2.5-24), (2.5-27), and (2.5-28). Thus the longitudinal coefficient matrices are given by

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & V_T - Z_{\dot{\alpha}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -M_{\dot{\alpha}} & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} X_{\delta_{th}} \cos \alpha_e & X_{\delta_e} \\ -X_{\delta_{th}} \sin \alpha_e & Z_{\delta_e} \\ 0 & 0 \\ M_{\delta_{th}} & M_{\delta_e} \end{bmatrix}$$

$$A = \begin{bmatrix} X_V + X_{T_V} \cos \alpha_e & X_\alpha & -g'_0 \cos \gamma_e & 0 \\ Z_V - X_{T_V} \sin \alpha_e & Z_\alpha & -g'_0 \sin \gamma_e & V_T + Z_q \\ 0 & 0 & 0 & 1 \\ M_V + M_{T_V} & M_\alpha & 0 & M_q \end{bmatrix}. \quad (2.5-31)$$

Note that $Z_{\dot{\alpha}}$ is normally small and V_T is considerably greater than zero for nonhovering flight, so the E matrix is always nonsingular.

The lateral-directional states and controls are

$$\mathbf{x}^T = [\beta \quad \phi \quad p_w \quad r_w], \quad \mathbf{u}^T = [\delta_a \quad \delta_r], \quad (2.5-32)$$

where the state ψ has been dropped. The state equations are obtained from the second rows of (2.5-11), (2.5-14), and (2.5-16), the first rows of (2.5-18) and (2.5-20), and the first and third rows of (2.5-24), (2.5-27), and (2.5-28). The resulting coefficient matrices are

$$E = \begin{bmatrix} V_T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ 0 & 0 \\ L'_{\delta_a} & L'_{\delta_r} \\ N'_{\delta_a} & N'_{\delta_r} \end{bmatrix}$$

$$A = \begin{bmatrix} Y_\beta & g'_0 \cos \theta_e & Y_p & Y_r - V_T \\ 0 & 0 & \frac{\cos \gamma_e}{\cos \theta_e} & \frac{\sin \gamma_e}{\cos \theta_e} \\ L'_\beta & 0 & L'_p & L'_r \\ N'_\beta & 0 & N'_p & N'_r \end{bmatrix}, \quad (2.5-33)$$

where primed moment derivatives have been defined, according to the common convention, by

$$\begin{aligned} L'_{\beta} &= \mu L_{\beta} + \sigma N_{\beta}, & L'_p &= \mu L_p + \sigma N_p, & L'_r &= \mu L_r + \sigma N_r, \\ N'_{\beta} &= \mu N_{\beta} + \sigma L_{\beta}, & N'_p &= \mu N_p + \sigma L_p, & N'_r &= \mu N_r + \sigma L_r, \end{aligned} \quad (2.5-34)$$

$$\begin{aligned} L'_{\delta a} &= \mu L_{\delta a} + \sigma N_{\delta a} & L'_{\delta r} &= \mu L_{\delta r} + \sigma N_{\delta r} \\ N'_{\delta a} &= \mu N_{\delta a} + \sigma L_{\delta a} & N'_{\delta r} &= \mu N_{\delta r} + \sigma L_{\delta r}. \end{aligned}$$

The E matrices for the two sets of equations are nonsingular (for other than hovering flight) and easily invertible in both cases. Therefore, although the original nonlinear equations were assumed implicit, the linear equations can be made explicit. The coefficient matrices depend on the steady-state angle of attack and pitch attitude in both cases. Although they nominally apply to small perturbations about a wings-level steady-state flight condition, the equations can be used satisfactorily for perturbed bank angles of several degrees.

The linear equations were derived in wind axes in order to obtain the beta-dot state derivative, and then beta was set to zero. Because the steady-state value of beta is zero, stability-axes aerodynamic derivatives can be used in the equations. Also, the moments of inertia used in the equations are the stability-axes values obtained from (2.4-18). The subscript W on the rates P , Q , and R could equally well be replaced by a subscript S . In the definitions of stability derivatives in the remainder of this chapter, the subscripts will be dropped altogether because the stability derivatives are normally specified in the stability axes.

In this chapter we shall be content with simply deriving the coefficient matrices for the linear state-equations; the equations will not be used until Chapter 3. The remainder of the chapter will be devoted to expressing the dimensional stability derivatives, used in the coefficient matrices, in terms of derivatives of the dimensionless aerodynamic coefficients defined in (2.3-7). The resulting "dimensionless derivatives" have the advantage that they are less dependent on the specific aircraft and flight condition, and more dependent on the geometrical configuration of an aircraft. Methods have been developed to estimate the dimensionless derivatives, and they can be used to compare and assess different design configurations.

The Dimensionless Stability and Control Derivatives

The dimensional aerodynamic derivatives are simply a convenient set of coefficients for the linear equations. We must now relate them to the

dimensionless stability derivatives used by stability and control engineers and found in most aerodynamic databases. The way in which the stability derivatives are made dimensionless depends on whether the independent variable for the differentiation is angle, angular rate, or velocity. This will be illustrated by example before we tabulate the derivatives.

Consider the derivative X_V in Table 2.5-1; this derivative is taken with respect to airspeed. The drag force depends on airspeed both through dynamic pressure and through the variation of the aerodynamic drag coefficient with airspeed. Therefore, using the definition of X_V and the drag equation from (2.3-7), we have

$$X_V = -\frac{1}{m} \left[\frac{\partial \bar{q}}{\partial V_T} S C_D + \bar{q} S \frac{\partial C_D}{\partial V_T} \right] = -\frac{\bar{q} S}{m V_T} (2 C_D + C_{D_V}),$$

where $C_{D_V} \equiv V_T (\partial C_D / \partial V_T)$ is the dimensionless speed damping derivative and C_D is the drag coefficient.

Next, consider a derivative that is taken with respect to angular rate, C_{m_q} . The dimensionless rate-dampling derivatives were actually defined in connection with Table 2.3-2 and can now be related to the dimensional derivatives. Making use of the definition of pitching moment coefficient in (2.3-7), we have

$$M_q = \frac{\bar{q} S \bar{c}}{J'_y} \frac{\partial C_M}{\partial Q} = \frac{\bar{q} S \bar{c}}{J'_y} \frac{\bar{c}}{2 V_T} C_{m_q}, \quad \text{where } C_{m_q} \equiv \frac{2 V_T}{\bar{c}} \frac{\partial C_M}{\partial Q}.$$

The “dimensionless” stability derivatives taken with respect to angle actually have dimensions of deg^{-1} when expressed in degrees rather than radians.

Tables 2.5-1 and 2.5-2 include six thrust derivatives (X_{T_V} , $X_{\delta_{th}}$, M_{T_V} , M_{T_α} , M_{T_β} , N_{T_β}). The corresponding dimensionless derivatives can be defined by expressing the thrust force and moment components in terms of dimensionless coefficients. For example, a pitching moment component due to thrust can be written as $M_T = \bar{q} S \bar{c} C_{m_T}$. Values for the thrust derivatives would be found by referring to the “installed thrust” data for the specific engine and determining the change in thrust due to a perturbation in the variable of interest. In the case of the derivatives with respect to V_T and throttle setting, it is probably most convenient to work directly with the dimensional derivatives (see Example 3.6-1). Determination of the thrust derivatives with respect to α and β is more complicated; a readable explanation is given in Roskam [1979].

Following the lines of the examples above, the longitudinal dimensionless stability and control derivatives and the lateral-directional dimensionless stability and control derivatives corresponding to Table 2.5-1 and 2.5-2 are given in Tables 2.5-3 and 2.5-4. The thrust derivatives have not been included in these tables for the reasons given above.

TABLE 2.5-3. Longitudinal Dimensional Versus Dimensionless Derivatives

$X_V = -\frac{\bar{q}S}{mV_T}(2C_D + C_{D_V}),$	$C_{D_V} \equiv V_T \frac{\partial C_D}{\partial V_T}$
$X_\alpha = \frac{\bar{q}S}{m}(C_L - C_{D_\alpha}),$	$C_{D_\alpha} \equiv \frac{\partial C_D}{\partial \alpha}$
$X_{\delta e} = -\frac{\bar{q}S}{m}C_{D_{\delta e}},$	$C_{D_{\delta e}} \equiv \frac{\partial C_D}{\partial \delta e}$
$Z_V = -\frac{\bar{q}S}{mV_T}(2C_L + C_{L_V}),$	$C_{L_V} \equiv V_T \frac{\partial C_L}{\partial V_T}$
$Z_\alpha = -\frac{\bar{q}S}{m}(C_D + C_{L_\alpha}),$	$C_{L_\alpha} \equiv \frac{\partial C_L}{\partial \alpha}$
$Z_{\dot{\alpha}} = -\frac{\bar{q}S\bar{c}}{2mV_T}C_{L_{\dot{\alpha}}},$	$C_{L_{\dot{\alpha}}} \equiv \frac{2V_T}{\bar{c}} \frac{\partial C_L}{\partial \dot{\alpha}}$
$Z_q = -\frac{\bar{q}S\bar{c}}{2mV_T}C_{L_q},$	$C_{L_q} \equiv \frac{2V_T}{\bar{c}} \frac{\partial C_L}{\partial Q}$
$Z_{\delta e} = -\frac{\bar{q}S}{m}C_{L_{\delta e}},$	$C_{L_{\delta e}} \equiv \frac{\partial C_L}{\partial \delta e}$
$M_v = \frac{\bar{q}S\bar{c}}{J_Y V_T}(2C_M + C_{m_V}),$	$C_{m_V} \equiv V_T \frac{\partial C_M}{\partial V_T}$
$M_\alpha = \frac{\bar{q}S\bar{c}}{J_Y}C_{m_\alpha},$	$C_{m_\alpha} \equiv \frac{\partial C_M}{\partial \alpha}$
$M_{\dot{\alpha}} = \frac{\bar{q}S\bar{c}}{J_Y} \frac{\bar{c}}{2V_T}C_{m_{\dot{\alpha}}},$	$C_{m_{\dot{\alpha}}} \equiv \frac{2V_T}{\bar{c}} \frac{\partial C_M}{\partial \dot{\alpha}}$
$M_q = \frac{\bar{q}S\bar{c}}{J_Y} \frac{\bar{c}}{2V_T}C_{m_q},$	$C_{m_q} \equiv \frac{2V_T}{\bar{c}} \frac{\partial C_M}{\partial Q}$
$M_{\delta e} = \frac{\bar{q}S\bar{c}}{J_Y}C_{m_{\delta e}},$	$C_{m_{\delta e}} \equiv \frac{\partial C_M}{\partial \delta e}$

TABLE 2.5-4. Lateral-Directional Dimensional Versus Dimensionless Derivatives

$Y_\beta = \frac{\bar{q}S}{m} C_{Y_\beta}$	$C_{Y_\beta} \equiv \frac{\partial C_Y}{\partial \beta}$
$Y_p = \frac{\bar{q}Sb}{2mV_T} C_{Y_p},$	$C_{Y_p} \equiv \frac{2V_T}{b} \frac{\partial C_Y}{\partial P}$
$Y_r = \frac{\bar{q}Sb}{2mV_T} C_{Y_r},$	$C_{Y_r} \equiv \frac{2V_T}{b} \frac{\partial C_Y}{\partial R}$
$Y_{\delta r} = \frac{\bar{q}S}{m} C_{Y_{\delta r}},$	$C_{Y_{\delta r}} \equiv \frac{\partial C_Y}{\partial r dr}$
$Y_{\delta a} = \frac{\bar{q}S}{m} C_{Y_{\delta a}},$	$C_{Y_{\delta a}} \equiv \frac{\partial C_Y}{\partial ail}$
$L_\beta = \frac{\bar{q}Sb}{J'_X} C_{l_\beta},$	$C_{l_\beta} \equiv \frac{\partial C_l}{\partial \beta}$
$L_p = \frac{\bar{q}Sb}{J'_X} \frac{b}{2V_T} C_{l_p},$	$C_{l_p} \equiv \frac{2V_T}{b} \frac{\partial C_l}{\partial P}$
$L_r = \frac{\bar{q}Sb}{J'_X} \frac{b}{2V_T} C_{l_r},$	$C_{l_r} \equiv \frac{2V_T}{b} \frac{\partial C_l}{\partial R}$
$L_{\delta a} = \frac{\bar{q}Sb}{J'_X} C_{l_{\delta a}},$	$C_{l_{\delta a}} \equiv \frac{\partial C_l}{\partial ail}$
$L_{\delta r} = \frac{\bar{q}Sb}{J'_X} C_{l_{\delta r}},$	$C_{l_{\delta r}} \equiv \frac{\partial C_l}{\partial r dr}$
$N_\beta = \frac{\bar{q}Sb}{J'_Z} C_{n_\beta},$	$C_{n_\beta} \equiv \frac{\partial C_n}{\partial \beta}$
$N_p = \frac{\bar{q}Sb}{J'_Z} \frac{b}{2V_T} C_{n_p},$	$C_{n_p} \equiv \frac{2V_T}{b} \frac{\partial C_n}{\partial P}$
$N_r = \frac{\bar{q}Sb}{J'_Z} \frac{b}{2V_T} C_{n_r},$	$C_{n_r} \equiv \frac{2V_T}{b} \frac{\partial C_n}{\partial R}$
$N_{\delta a} = \frac{\bar{q}Sb}{J'_Z} C_{n_{\delta a}},$	$C_{n_{\delta a}} \equiv \frac{\partial C_n}{\partial ail}$
$N_{\delta r} = \frac{\bar{q}Sb}{J'_Z} C_{n_{\delta r}},$	$C_{n_{\delta r}} \equiv \frac{\partial C_n}{\partial r dr}$

The dimensionless stability derivatives are in general very important to both the aircraft designer and the stability and control engineer. They provide information about the natural stability of an aircraft, about the effectiveness of the control surfaces, and about the maneuverability. They correlate with the geometrical features of the aircraft and thereby facilitate the preliminary design process. The typical variation of many of the stability derivatives with flight conditions (e.g., speed, angle of attack, sideslip angle) is known to the designer, and he or she can therefore anticipate the design problems in different parts of the flight envelope. Information on the importance of the stability derivatives, the accuracy with which they can be estimated, and their variation with flight conditions can be found in stability and control textbooks [Roskam, 1979; Etkin, 1972; Perkins and Hage, 1949] and in the USAF DATCOM [Hoak et al., 1970]. Stability derivatives at certain flight conditions, for a number of different aircraft, are also given in these books, in Blakelock [1965], in McRuer et al. [1973], and in various other texts.

The decoupled linear state equations can be expressed directly in terms of the dimensionless derivatives, with the object of more closely relating the dynamics to known coefficients. The resulting equations can be made dimensionless by introducing a dimensionless time variable [Etkin, 1972; Babister, 1961]. While these formulations are useful for limited algebraic studies, it will become apparent later that they are irrelevant in a computer-based design and simulation framework. Furthermore, in Section 4.2 we show that to understand the dynamic behavior, the linear equations can be simplified and approximated before the dimensionless derivatives are substituted.

In the next subsection we briefly describe the significance of various dimensionless derivatives and their variation with flight conditions. This information will be utilized in Chapter 4 when aircraft dynamic modes are analyzed.

Description of the Dimensionless Derivatives[†]

The most important longitudinal stability derivatives for conventional aircraft are $C_{L\alpha}$, $C_{m\alpha}$, C_{m_q} , $C_{m\nu}$, C_{m_a} and $C_{D\nu}$. The derivative $C_{L\alpha}$ is the *lift curve slope* (see Fig. 2.2-3), it is typically in the range 1 to 8 rad^{-1} for the linear region of the lift curve below stall. It increases with Mach number as the transonic range is approached, peaks, and then falls off rapidly at high subsonic speeds (cf. the airfoil characteristics in Section 2.2). The importance of this derivative lies in the fact that it relates changes in angle of attack, produced, for example, by wind gusts or maneuvers, to z -axis acceleration.

[†]The typical values cited in this section are quoted from Roskam [1979].

Therefore, it has a strong influence on the turbulence response and the maneuverability of the airplane.

In Section 2.3 the slope of the pitching moment curve, C_{m_α} , was shown to be the parameter that determines the pitch stiffness of the aircraft. A negative value of C_{m_α} is usually required for stability, but a pilot can still control a slightly unstable aircraft albeit with an increased workload. Its value is typically in the range from -3 rad^{-1} to small positive values. We shall see later that a stable aircraft responds to a pitch command with a response that tends to be a damped oscillation, whose natural frequency is determined by C_{m_α} and the pitch-axis moment of inertia. In fact, C_{m_α} is equivalent to the restoring torque constant in a simple spring/inertia mechanical oscillator. This derivative varies with Mach number and may change erratically in the transonic range.

The derivatives C_{m_q} and C_{m_α} are pitch damping derivatives and were introduced in Section 2.3. The damping of the aircraft pitch oscillation, referred to above, is dependent on these derivatives. C_{m_q} is usually of much greater magnitude than C_{m_α} and typically has a value in the range from zero to about -40 rad^{-1} . Their variation with Mach number tends to be relatively smooth and gradual.

The effect of speed variations on pitching moment is contained in the derivative C_{m_V} . If this derivative is positive, the aircraft will tend to pitch-up with an increase in speed. The resulting increase in drag and the greater component of gravity along the x -axis will then tend to slow the aircraft down. This is stable behavior. In practice, the effective center of pressure of an aircraft may move aft with increasing subsonic Mach number, thus creating a tendency to dive (negative C_{m_V}) with associated unstable speed behavior. Depending on the aircraft, this *tuck-under effect* may be benign because of a large accompanying transonic drag rise, or troublesome because of simultaneously decreasing control effectiveness with Mach number. This emphasizes the importance of the speed damping derivative C_{D_V} .

The most important lateral-directional stability derivatives for conventional aircraft are C_{y_β} , C_{l_β} , C_{n_β} , C_{l_p} , C_{n_p} , C_{l_r} , and C_{n_r} . The derivative C_{y_β} is known as the side-force derivative due to sideslip and typically lies in the range -0.1 to -2.0 rad^{-1} . The remaining important lateral-directional derivatives are moment derivatives.

The derivative C_{l_β} is the rolling moment due to sideslip and is often known as the *dihedral derivative*. It was discussed in connection with the aerodynamic rolling moment in Section 2.3. For positive roll stiffness this derivative must be negative. Its value is typically in the range -0.4 to $+0.1 \text{ rad}^{-1}$ and may change significantly with Mach number in the transonic range. It is important because it affects the stability of the aircraft dynamic behavior in what is known as the *spiral mode*.

The derivative C_{n_β} is the yawing moment due to sideslip, or *yaw-stiffness derivative*, and is associated with weathercock stability (see Section 2.2) and

the properties of the *dutch roll* mode of oscillation (see Section 4.2). It must be positive for positive stiffness in yaw; its value is typically in the range 0 to 0.4 rad^{-1} , tending to fall off and possibly even become negative at transonic to supersonic Mach numbers. It may also become negative at high angles of attack when the vertical tail becomes immersed in the wake from the wings and body. Achieving a suitable value of C_{n_β} is a consideration in the initial sizing of the vertical tail of an aircraft.

The derivatives C_{l_p} , C_{n_p} , C_{n_r} , and C_{l_r} are all damping moment derivatives and were introduced in Section 2.3. The roll-damping derivative C_{l_p} usually lies in the range -0.10 to -0.80 rad^{-1} ; it will typically increase in magnitude in the transonic region and then decrease again at higher Mach numbers. This derivative is important in determining the roll-rate performance of a highly maneuverable aircraft. The yaw-damping derivative is C_{n_r} ; it is always negative and is typically in the range from -1 rad^{-1} to near zero. It also may increase in magnitude in the transonic region.

The derivatives C_{n_p} and C_{l_r} are *cross-derivatives* that exist because of the coupling of the rolling and yawing motions of an aircraft. They tend to be difficult to measure and can change greatly with Mach number. C_{n_p} usually lies in the range from -0.5 to $+0.1 \text{ rad}^{-1}$ and can change dramatically from negative to positive values in the transonic region. C_{l_r} is typically between zero and 0.6 rad^{-1} , it may vary significantly with Mach number and is also a function of the lift coefficient.

In Chapter 4 we use the linear state equations and the dimensionless stability derivatives to derive some information about the dynamic modes of an aircraft.

2.6 SUMMARY

In this chapter we have described how the aerodynamic forces and moments acting on an aircraft are created, how they are modeled mathematically, and how the data for the models are gathered. We have related these forces and moments to the equations of motion of a rigid aircraft that were derived in Chapter 1. The transformation of the equations of motion into a different set of coordinates has been demonstrated, and also the derivation of a nonlinear model for longitudinal motion only. Steady-state flight conditions have been defined. It has been shown that the equations of motion can be linearized around a steady-state condition and that they can then be separated into two decoupled sets. One of these sets describes the longitudinal motion of an aircraft, and the other describes the lateral-directional motion. The linear equations have been expressed in terms of the aerodynamic derivatives, and the significance of these derivatives has been explained. In Chapter 3 we develop a number of powerful analytical and computational tools and use them in conjunction with the aircraft models developed here.

REFERENCES

- Anderson, F., *Northrop: An Aeronautical History*. Los Angeles: Northrop Corporation, 1976.
- Anderson, J. D., Jr., *Fundamentals of Aerodynamics*. New York: McGraw-Hill, 1984.
- Babister, A. W., *Aircraft Stability and Control*. Oxford: Pergamon Press, 1961.
- Blakelock, J. H., *Automatic Control of Aircraft and Missiles*. New York: Wiley, 1965.
- Bryan, G. H., *Stability in Aviation*. London: Macmillan, 1911.
- DeCamp, R. W., R. Hardy, and D. K. Gould, "Mission adaptive wing," *SAE Pap. 872419*, 1987.
- Dommasch, D. O., S. S. Sherby, and T. F. Connolly, *Airplane Aerodynamics*, 4th ed. New York: Pitman, 1967.
- Drendel, L., *SR-71 Blackbird in Action*. Carrollton, Tex.: Squadron/Signal Publications, 1982.
- Droste, C. S., and J. E. Walker, "The General Dynamics case study on the F-16 fly-by-wire flight control system," *AIAA Professional Case Studies*, no date.
- Duncan, W. J., *The Principles of the Control and Stability of Aircraft*. Cambridge: Cambridge University Press, 1952.
- Etkin, B., *Dynamics of Atmospheric Flight*. New York: Wiley, 1972.
- Hoak, D. E., et al., *USAF Stability and Control DATCOM*, Flight Control Division, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, September 1970.
- IMSL, *Library Contents Document*, 8th ed., International Mathematical and Statistical Libraries, Inc., 7500 Bellaire Blvd., Houston, TX 77036, 1980.
- Kandebo, S. W., "Second X-29 will execute high-angle-of-attack flights," and Scott, W. B., "X-29 proves viability of forward-swept wing," *Aviat. Week Space Technol.*, Oct. 31, 1988, pp. 36-42.
- Kuethe, A. M., and C. Y. Chow, *Foundations of Aerodynamics*, 4th ed. New York: Wiley, 1984.
- Maine, R. E., and K. W. Iliffe, "Formulation and implementation of a practical algorithm for parameter estimation with process and measurement noise," paper 80-1603, *AIAA Atmos. flight Mech. Conf.*, Aug. 11-13, 1980, pp. 397-411.
- McFarland, R. E., "A standard kinematic model for flight simulation at NASA Ames," *NASA CR-2497*, NASA, Washington, D.C., Jan. 1975.
- McRuer, D., I. Ashkenas, and D. Graham, *Aircraft Dynamics and Automatic Control*. Princeton, N.J.: Princeton University Press, 1973.
- Nguyen, L. T., et al., "Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability," *NASA Tech. Paper 1538*, NASA, Washington, D.C., Dec. 1979.
- Perkins, C. D., and R. E. Hage, *Airplane Performance Stability and Control*. New York: Wiley, 1949.
- Pope, A., *Wind Tunnel Testing*. New York: Wiley, 1954.
- Queijo, M. J., "Methods of obtaining stability derivatives," *Performance and Dynamics of Aerospace Vehicles*, by the staff of NASA-Langley Research Center, NASA, Washington, D.C., 1971.

- Rech, J., and C. S. Leyman, "A case study by Aerospatiale and British Aerospace on the Concorde," *AIAA Professional Case Studies*, no date.
- Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls*. Lawrence, Kans.: Roskam Aviation and Engineering Corp., 1979.
- Vidyasagar, M., *Nonlinear System Analysis*. Englewood Cliffs, N.J.: Prentice Hall, 1978.
- Yuan, S. W., *Foundations of Fluid Mechanics*. Englewood Cliffs, N.J.: Prentice Hall, 1967.

PROBLEMS

Section 2.3

- 2.3-1** Program the functions for the body-axes force coefficients CX and CZ, as given in Appendix A for the F-16 model. Write another program to use these data and plot a set of curves of lift coefficient as a function of alpha (for $-10^\circ \leq \alpha \leq 50^\circ$), with elevator deflection as a parameter (for $\text{el} = -25^\circ, 0^\circ, 25^\circ$). Determine the angle of attack at which maximum lift occurs.
- 2.3-2** Program the body-axes moment coefficient CM, as given in Appendix A for the F-16 model. Write another program to plot a set of curves of pitching moment as a function of alpha (for $-10^\circ \leq \alpha \leq 50^\circ$), with elevator deflection as a parameter (for $\text{el} = -25^\circ, 0^\circ, 25^\circ$). Comment on the pitch stiffness and on the elevator control power.
- 2.3-3** Program the F-16 engine thrust model, function THRUST, in Appendix A. Write a program to plot the thrust as a function of power setting (0 to 100%), with altitude as a parameter (for $h = 0, 25000$ ft, 50000 ft), at Mach 0.6. Also, plot thrust against Mach number, at 100% power, with altitude as a parameter (for $h = 0, 25000$ ft, 50000 ft). Comment on these characteristics of the jet engine.

Section 2.4

- 2.4-1** Expand the flat-Earth vector-form equations of motion, Equations (1.5-4), into 12 scalar equations. Check the results against (2.4-2) through (2.4-5).
- 2.4-2** Derive expressions for the derivatives of V_T , α , and β , in terms of U , V , and W and their derivatives. Check the results against (2.4-8).
- 2.4-3** Derive the cross-product matrix, Ω_R , for the rotation rate of the wind axes relative to the body axes. Compare the answer with (2.4-11).
- 2.4-4** Fill in all of the steps in the derivation of the scalar form of the flat-Earth rigid-body force equations, (2.4-13).

- 2.4-5 Derive the inverse of the inertia matrix, (2.4-18), and check the answer against (2.4-19).

Section 2.5

- 2.5-1 Fill in all of the steps in the derivation of the coefficient matrices for the linearized force equations [(2.5-11), (2.5-14), and (2.5-16)].
- 2.5-2 Fill in all of the steps in the derivation of the coefficient matrix, (2.5-20), for the linearized kinematic equations.
- 2.5-3 Fill in all of the steps in the derivation of the coefficient matrices for the linearized moment equations [(2.5-24), (2.5-27), and (2.5-28)].
- 2.5-4 Write a program to calculate (approximately) the derivative of a function of a single variable (assumed to be continuous), given discrete values of the function. Use the program with the lookup-table from Problem 2.3-2, to estimate the derivative $C_{m\alpha}$ at the values of $\alpha = 0^\circ, 10^\circ, 20^\circ$, and 30° (when $e_l = 0^\circ$). Hence determine whether the aircraft has positive pitch stiffness at these angles of attack.

CHAPTER 3

BASIC ANALYTICAL AND COMPUTATIONAL TOOLS

3.1 INTRODUCTION

We have placed a great deal of emphasis on the state-space formulation of the aircraft differential equations, and the advantages of this formulation will become increasingly evident as analytical tools and numerical algorithms are developed in this chapter. The numerical algorithms can readily be turned into software tools. The software tools will provide the capability to trim aircraft models for steady-state flight, perform digital flight simulation, extract linear state-space and transfer function descriptions of aircraft models, and perform operations on the linear equations. These linear equations allow us to design aircraft flight control systems in the later chapters. Figure 3.1-1 shows how the software tools fit together and emphasizes the central role of the nonlinear state-space model. We refer to this figure throughout the following sections as each algorithm is developed.

In Section 3.2 we complete the model building process by showing how various subsystems can be modeled for incorporation into the aircraft state-space model. In Section 3.3 two aircraft models are provided. The first model uses fixed aerodynamic derivatives and is strictly valid for only one flight condition; it will serve to illustrate how the aircraft dynamic modes depend on the stability derivatives. The second model is a much more realistic, nonlinear, six-degrees-of-freedom (6-DOF) aircraft model that can be used for simulation over a wide altitude/airspeed envelope. These models make use of the rigid-body equations that were developed in Chapter 1, the aerodynamic force and moment buildup equations described in Chapter 2, and in the case of the second model, tabular wind-tunnel data. They

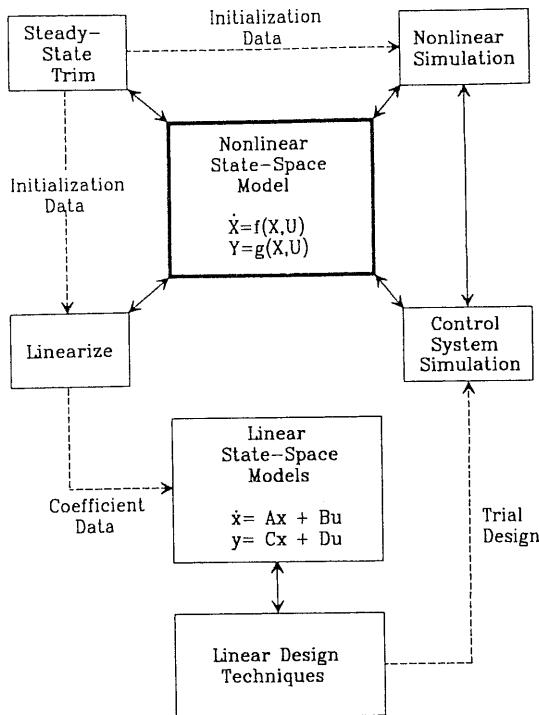


Fig. 3.1-1 State-space models and associated software.

illustrate many of the aerodynamic principles that were described in Chapter 2, are easily reproducible, and can be run on a personal computer that has the capability to run FORTRAN (or Pascal or C) programs.

Section 3.4 deals with the determination of steady-state flight conditions. A steady-state flight condition is a prerequisite for starting a flight simulation or beginning the process of control system design for the aircraft. Certain constraints must be imposed on the aircraft model according to the nature of the steady-state condition (e.g., straight and level, or a level turn). These constraints are derived, and a computer algorithm to solve the constrained nonlinear equations is described.

In Section 3.5 numerical algorithms for integrating the state equations are described. These algorithms make possible digital simulation of continuous dynamics and are our single most important tool. In contrast to most analytical methods for solving differential equations (e.g., the Laplace transform), they are not limited to linear systems, and input signals of arbitrary waveform can be applied to the dynamics. We develop enough theory to understand the limitations of the techniques and be able to choose the most appropriate algorithm for a given application. Numerical integration will then

be demonstrated with an aircraft simulation, and will subsequently be used to examine the performance of various aircraft control systems.

Numerical linearization of the nonlinear aircraft dynamics is described in Section 3.6. Unlike the algebraic linearization developed in Chapter 2, this technique will make the linear state equations for *any* steady-state flight condition immediately available to us. Therefore, we can, for example, derive the coupled linear equations that result from a high-*g* steady-state turn, or examine the dynamics in a rapid roll or pull-up. All of the linear equations used to illustrate control system design will be derived from the nonlinear aircraft model in this way.

In Section 3.7 a number of different analytical tools will be developed. These tools will provide a deeper understanding of the dynamic behavior of an aircraft and will also make possible the control system design techniques used in Chapter 4. Finally, in Section 3.8 we review the principles of feedback control and develop a state-space design framework that can be used for control design throughout the rest of the book.

3.2 MODELS OF SUBSYSTEMS

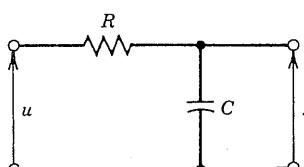
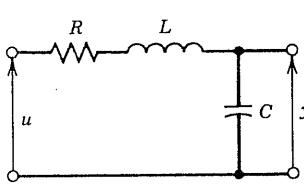
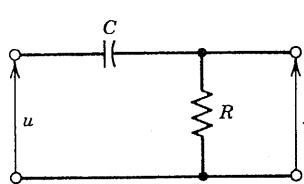
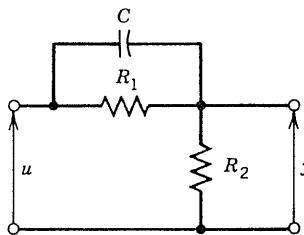
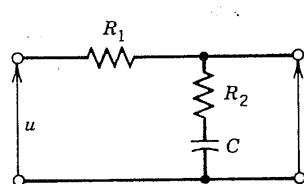
So far we have focused all of our modeling effort on the equations of motion of a rigid body and the aerodynamic forces and moments acting on an aircraft. This is not the full extent of the modeling problem; an aircraft carries a large number of subsystems, many of which affect the dynamics of the aircraft motion. These include the actuators that move the aerodynamic control surfaces, the sensors that measure airstream and inertial data, signal conditioning networks that modify these data, and compensation networks that are added to the control system to help meet performance specifications. We must therefore develop some familiarity with the methods of modeling these subsystems.

The mathematical models of the subsystems that are used to control the aircraft dynamics occur in the form of ordinary differential equations of various orders, transfer functions (in terms of the Laplace transform variable), and state equations. It is necessary to be able to derive the dynamic equations with whichever technique is most expedient, and to be able to convert freely from one representation to another. For linear systems all three techniques are used and formal methods of deriving the equations exist. For nonlinear systems transfer function methods cannot be used, and the methods of deriving differential equations or state equations are ad hoc.

Transfer Function Models

Table 3.2-1 shows a number of electrical networks that are analogs of many of the subsystem models that will be needed for a simulation of an aircraft and its flight control systems. The transfer functions have been derived by

TABLE 3.2-1. Network Transfer Functions and State Equations

Network	Transfer Function	State Equations
	$\frac{1}{1 + s\tau}, \quad \tau = CR$	$\dot{x} = \frac{u - x}{\tau}$ $y = x$
Simple lag		
	$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$ $\omega_n^2 = \frac{1}{LC}$ $\zeta = \frac{R}{2} \sqrt{\frac{C}{L}}$	$\dot{x}_1 = x_2$ $\dot{x}_2 = -\omega_n^2 x_1 - 2\zeta\omega_n x_2 + \omega_n^2 u$ $y = x_1$
Quadratic lag		
	$\frac{s\tau}{1 + s\tau}, \quad \tau = CR$	$\dot{x} = \frac{u - x}{\tau}$ $y = u - x$
Simple lead		
	$z = 1/\tau$ $\frac{s + z}{s + p}, \quad p = 1/(\alpha\tau)$ $\alpha = \frac{R_2}{R_1 + R_2}$ $\tau = CR_1$	$\dot{x} = u - px$ $y = u + (z - p)x$
Lead compensator		
	$z = 1/(\alpha\tau)$ $\alpha \left(\frac{s + z}{s + p} \right), \quad p = 1/\tau$ $\alpha = \frac{R_2}{R_1 + R_2}$ $\tau = C(R_1 + R_2)$	$\dot{x} = u - px$ $y = \alpha[u + (z - p)x]$
Lag compensator		

representing the network elements by their (Laplace) transform impedances (i.e., $1/(sC)$ for a capacitor) and applying simple circuit analysis (assuming no source and output loading effects). The convention of using lowercase symbols for a time function and uppercase for its Laplace transform will be followed. State equations are shown for the networks and can be derived from the transfer functions by the method given later in this section. All of these state equations (except the quadratic lag) are used in controller designs in Chapter 4.

Network 1 is a *simple lag*; it is described by a first-order differential equation and corresponds to the simplest model of a system whose speed of response is limited (because of the time needed to change its stored energy). It is commonly used as a model of an actuator, or a complete servomechanism, that moves an aerodynamic control surface or adjusts the pitch of propeller blades. We will also use it to model the throttle response of a turbofan jet engine in the next section. The second network in Table 3.2-1 is a *quadratic lag* (i.e., a second-order lag). It is used for the same purposes as the first network when the exponential time response of that network is not a good fit to the actual system response. A second-order model is needed when the actual time response includes overshooting (and undershooting) behavior to a step input.

The third network is a first-order high-pass filter that is used to provide approximate differentiation; it or its discrete-time (digital) equivalent are commonly used on the yaw-rate feedback signal in a flight control system. Its output is zero in the steady state, and it responds most strongly to rapidly changing input signals. The fourth and fifth networks are, respectively, a phase-lead and a phase-lag network, both of which we will use as compensation networks when we undertake classical control system design.

Model Parameter Identification

When a mathematical model is postulated for a physical process the subsequent problem is to identify the parameters of the model (e.g., transfer function or state equation coefficients). *Parameter identification* techniques are widely used; the most common approach is to use a set of discrete measurements and form a scalar function (performance index, or PI) of the difference between the observed response and the model response. A mathematical algorithm is then used to minimize the PI by adjusting the model parameters.

In *online* identification the processing is performed as the data are being measured, and the parameter estimates may continue to improve as more data are gathered. In *offline* processing the data collection is completed, and then the data are processed in a batch. Either frequency-response data or time-response data may be used for identification. Relevant examples of parameter identification include the NASA–Dryden MMLE3 program used

to estimate aircraft stability and control derivatives [Maine and Iliffe, 1980], the online parameter identification algorithms used in adaptive control [Harris and Billings, 1981], and various industry programs used to fit frequency-response data for assessment of flying qualities [Hodgkinson, 1979]. Background information on the identification problem can be found in various texts [Chen and Haas, 1968; Eykhoff, 1974; Goodwin and Sin, 1984].

State Models from Transfer Functions and ODEs

When a transfer function or ODE model with known parameters has been obtained, a method of converting this to a state model will be needed. There are a variety of ways in which this can be done [Kuo, 1987]; we shall present one technique that will be generally adequate for all of our needs. Consider the differential equation

$$\ddot{y} + a_1\dot{y} + a_0y = b_2\ddot{u} + b_1\dot{u} + b_0u, \quad (3.2-1)$$

where u and y are, respectively, the input and output of a system or network. The differential equation has an input derivative of the same order as the highest derivative of the output, so the relative degree of the transfer function (denominator degree – numerator degree) will be zero. This will lead to a direct-feed term ($D \neq 0$) in the state equations. When no input derivatives are present in the differential equation, the task of finding a state representation is trivial. State variables can simply be assigned to the output, y , and its derivatives up to order $(n - 1)$; n state equations can then be written down immediately. We shall now describe a general method of deriving state equations when input derivatives are present.

First transform (3.2-1) and, using an auxiliary variable $V(s)$, separate the numerator and denominator of the transfer function as follows:

$$\frac{Y(s)}{b_2s^2 + b_1s + b_0} = \frac{U(s)}{s^2 + a_1s + a_0} \equiv V(s). \quad (3.2-2)$$

Now draw a series of cascaded integrators as shown in Fig. 3.2-1, with $V(s)$ as the output of the last integrator and the number of integrators equal to the degree of the transfer function denominator polynomial. The operation of integration corresponds to a transfer function $1/s$ and so, working backward from the last integrator, we can identify the integrator inputs as sV , s^2V , and so on. To find out how the input $U(s)$ should be connected in this diagram, rewrite the right half of (3.2-2) as

$$s^2V = U - a_1sV - a_0V. \quad (3.2-3)$$

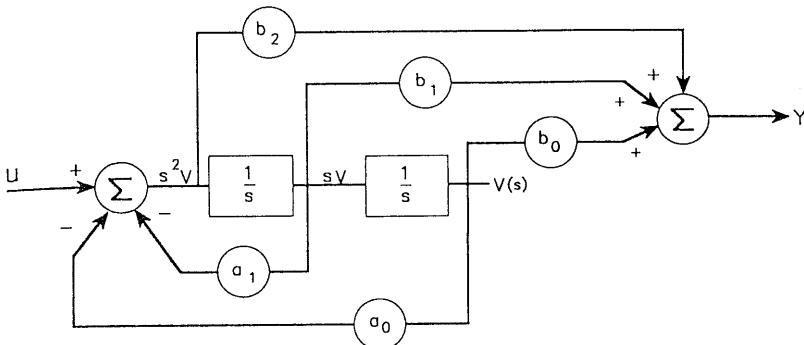


Fig. 3.2-1 Analog computer simulation diagram.

This equation relates the input of the first integrator to the transfer function input, U , and the feedback signals from later integrators. The appropriate connections are shown in the figure. Next, to establish the output connections, rewrite the left half of (3.2-2) as

$$Y = b_2 s^2 V + b_1 s V + b_0 V, \quad (3.2-4)$$

which leads to the output connections shown in the figure.

Figure 3.2-1 is a representation of a transfer function or ODE that is commonly referred to as a *simulation diagram*. If the electronic hardware to perform integration and summation of signals is available, this diagram can be used to construct an electronic analog of the transfer function (i.e., an analog computer simulation). We shall use the simulation diagram simply as an intermediate step in constructing state equations from transfer functions or ODEs. The integrators in the simulation diagram are the energy storage devices, and their output variables can be chosen as the state variables. Therefore, assigning state variables to integrator outputs beginning with the rightmost integrator, the following state equations can readily be obtained:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (3.2-5)$$

$$y = [b_0 - b_2 a_0 \quad b_1 - b_2 a_1] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b_2 u.$$

The state equations derived by this approach are characterized by a companion-form A matrix (all elements zero except for superdiagonal 1's, and a nonzero bottom row). The simplicity of the companion form matrix makes it useful for theoretical derivations. It does not have good numerical computa-

tion properties, but this is not a problem when dealing with systems of low order.

The technique used to derive the simulation diagram dates from the era when all simulation was done with analog computers; this approach has the very useful feature that it can be extended to differential equations with time-varying coefficients [Laning and Battin, 1956; DeRusso et al., 1965]. Many other methods of deriving state equations from transfer functions are known. For example, the transfer function may be expanded as a sum of partial fractions, and each partial-fraction term can be turned into a separate state equation.

Nonlinear State Models

Nonlinearities abound in the dynamics of aircraft and their associated systems. In many cases the nonlinear behavior occurs when large-amplitude signals are present in the system, for example, large angles of attack and large control surface deflections or rates. In some cases nonlinear behavior is associated with small-amplitude signals, as in the case of dead-zone or Coulomb friction in mechanical subsystems. Nonlinear systems present formidable analytical difficulties, but computer simulation of their behavior often presents no particular problems. We now give an example of a hydraulic actuator, as used to operate aircraft control surfaces. This involves dead-zone and square-root nonlinearities.

Example 3.2-1: A Nonlinear Model of a Hydraulic Actuator. Figure 3.2-2 shows a *spool valve* and hydraulic ram that together comprise a hydraulic actuator used to move aircraft control surfaces, landing gear, and other

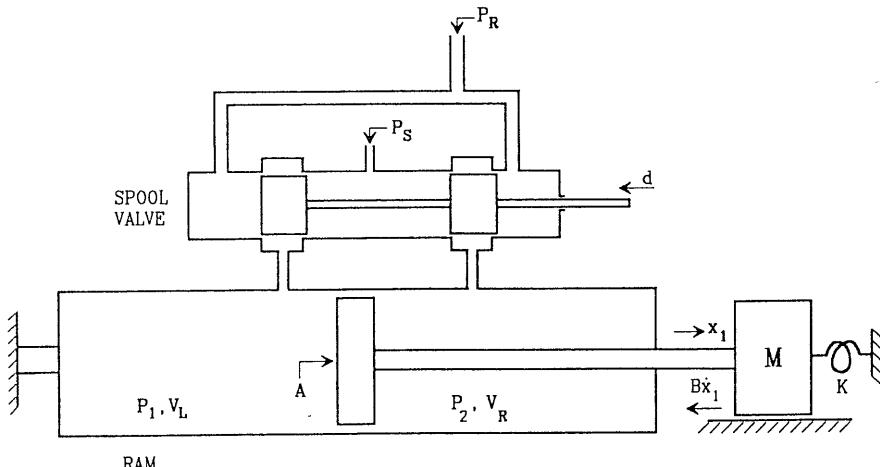


Fig. 3.2-2 Diagram of a hydraulic actuator.

devices. The load on the ram is represented as a simple mass-spring model. When the spool valve is in its central position, the valve *lands* cover the ports through which hydraulic fluid flows. When the spool valve is deflected, high-pressure (p_S) fluid from a hydraulic pump can flow to one side of the ram, and displaced fluid from the other side of the ram can return to the low-pressure (p_R) side of the pump. The spool valve is usually slightly overlapped; that is, the lands are wider than the ports in order to obtain a fluid seal. Therefore, there is a dead zone in the movement of the spool within which no actuator response occurs. Another important nonlinear effect is that for small openings, the flow rate through the orifices of the spool valve is proportional to the square root of the pressure drop across the orifice. Therefore, the equation relating flow rate, q , to pressure drop, Δp , is

$$q = k(d)\sqrt{\Delta p},$$

where the constant, k , is a function of the size and geometry of the orifice (and therefore of the spool-valve displacement d).

Another equation that will be needed is that relating rate of change of pressure, \dot{p} , to the rate at which a volume, v , of a fluid is compressed (i.e., a flow rate q). This is

$$\dot{p} = \frac{\beta}{v}q,$$

where β is the bulk modulus of the hydraulic fluid (2×10^5 psi for oil). Note that yet another source of nonlinearity is that the ram volume constitutes a variable coefficient in this equation for the rate of change of pressure.

With these basic equations we can construct a simulation diagram for Fig. 3.2-2, as shown in Fig. 3.2-3. In this simulation diagram the first box on the left represents the action of the spool valve. This switches the supply and return pressures to the dummy pressure variables p_A and p_B used to compute the pressure drops across the two orifices. The flow rates q_1 and q_2 can be computed from the pressure drops, and when these are summed with the ram displacement flows q_L and q_R , the "compressibility" flows are obtained. The ram-chamber volumes v_L and v_R , and the bulk modulus β , then yield the rates of change of pressure. These rates must be integrated to obtain the pressures p_1 and p_2 , needed for the orifice calculations. It is evident that these pressures, and the ram speed and displacement, must be assigned as state variables. The various quantities will then be available when they are needed in the sequence of calculations. Note that a viscous friction force, $B\dot{x}_1$, has been assumed at the load.

The hydraulic supply pressure is commonly 3000 or 4000 psi for high-performance aircraft systems, the return pressure is a small fraction of this, and for simplicity they are often assumed constant. Auxiliary devices such as

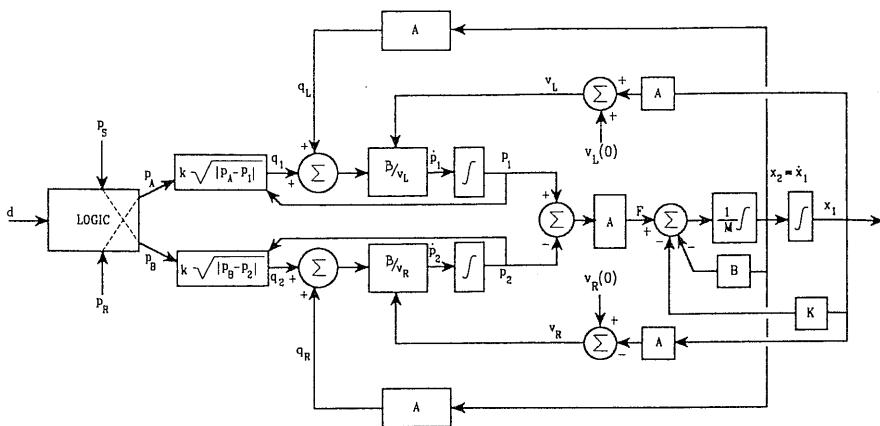


Fig. 3.2-3 Simulation of a hydraulic actuator.

hydraulic servo-compensation schemes have been omitted. In the case of a control surface actuator, the dynamics of the load depend on the way in which the actuator is mounted, the moment of inertia of the control surface, and the aerodynamic "hinge moments" and damping forces.

Figure 3.2-4 shows how the hydraulic actuator can be used as a *follow-up servo* when controlling the position of a control surface. The position of the ram body must follow the position of the spool valve; otherwise, a correcting flow will occur in the spool valve. The spool valve can be deflected electrically

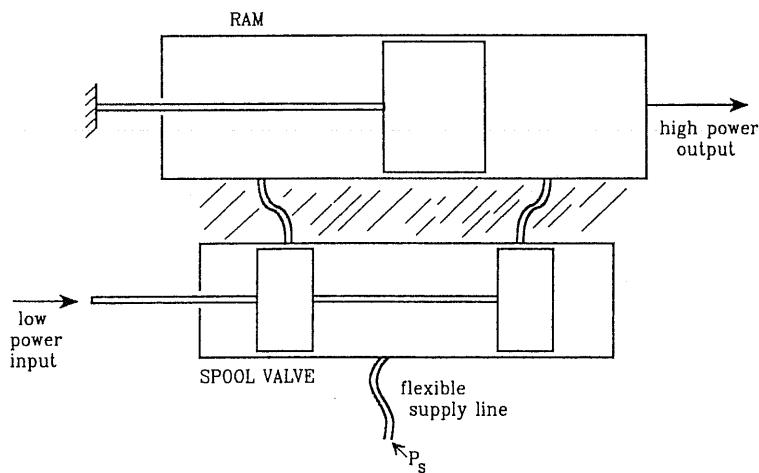


Fig. 3.2-4 A follow-up hydraulic servo.

or mechanically. The layout of a typical hydraulically powered flight-control system will be illustrated in Chapter 4.

3.3 AIRCRAFT MODELS FOR SIMULATION

A Simple Longitudinal Model[†]

This model has only three degrees of freedom (i.e., translation and pitching motion, in the vertical plane), it has fixed aerodynamic coefficients and is representative of a medium-sized transport aircraft at a low-speed flight condition. Data are also provided for the effects of extending landing gear and flaps. The aircraft weighs 162,000 lb (one-half fuel, partial cargo), and it has two turboprop engines each developing 30,000 lb of static thrust at sea level. The wing area is 2170 ft², wing span 140 ft, $\bar{c} = 17.5$ ft, length 90 ft, pitch-axis inertia 4.1×10^6 slug-ft², and reference cg position $0.25\bar{c}$.

The model is programmed as a FORTRAN subroutine named F, and the source code is reproduced in Fig. 3.3-1 (it has been condensed by multiple statements on a line). The formal arguments of the subroutine are the state vector (X) as an input, and the state derivative vector (XD) as an output. The other formal argument, Time, is unused in this particular model. The x -axis position of the cg is a parameter supplied through the common block PARAM, and control inputs are applied through the common block CONTROLS. Miscellaneous outputs are available through the common block OUTPUT, and the output A_n is "normal acceleration" (measured in g-units at the aircraft cg in this case). Normal acceleration is the acceleration measured along the negative z -axis; it is important because it is an indication of the load on the aircraft structure and on the pilot.

The aerodynamic derivatives are in stability axes and have "per degree" units except for the pitch damping coefficients (C_{M_q}, C_{M_a}), which are per radian/per second. There is provision for a C_{L_a} derivative, but it is zero in this case. Lift is calculated from a linear lift curve and the stall is not modeled, while drag is calculated from the nonlinear drag polar. The elevator deflection is in degrees, and the throttle input is in the range zero to 1. Atmospheric density (and hence dynamic pressure) is calculated in the subroutine ADC (air data computer, see Appendix A) from the temperature variation of the standard atmosphere [Yuan, 1967]. The engine thrust is modeled as decreasing linearly with airspeed, to approximate the characteristics of a propeller-driven aircraft. The thrust vector does not pass through the cg (the perpendicular distance from the vector to the cg is Z_E), and therefore throttle changes will tend to cause pitching motion of the aircraft. Other parts of the model are either self-evident or can be understood by referring

[†]Data for this model were given to the author by C. P. Moore III.

```

SUBROUTINE F(TIME,X,XD)
DIMENSION X(*),XD(*)
LOGICAL LAND
COMMON/PARAM/XCG,LAND
COMMON/CONTROLS/THTL,ELEV
COMMON/OUTPUT/AN,DUM2,DUM3,QBAR,AMACH,VT,ALPHA,THETAD,QD
DATA S,CBAR,AM,AIYY,G /2170.0, 17.5, 5.0E3, 4.1E6, 32.17/
DATA TSTAT,DTDV,ZE / 6.0E4, -38.0, 2.0/
DATA CLA,CLADOT,CDCLS,CMA /.085, 0.0, .042, -.022/
DATA CMQ,CMADOT, CMDE, RTOD/ -16.0, -6.0, -.016, 57.29578 /
C
      VT =X(1)                                ! T.A.S. (ft/s)
      ALPHA=X(2)*RTOD                         ! A.O.A. (deg)
      THETA=X(3)                               ! PITCH ATTITUDE (rad)
      Q   =X(4)                                ! PITCH RATE (rad/s)
      H   =X(5)                                ! ALTITUDE (ft)
C
      THETAD= RTOD*THETA;        QD= RTOD*Q
      CALL ADC(VT,H,AMACH,QBAR)
      QS=QBAR*S
      SALP=SIN(X(2));          CALP=COS(X(2))
      STH=SIN(THETA);          CTH=COS(THETA)
      GAM= THETA - X(2)
C
      IF (LAND) THEN                           ! landing gear and flaps
          CL0= 1.0; CD0= .08; DCDG=.02; DCMG=-.05
      ELSE                                     ! clean
          CL0=.20; CD0=.016; CM0=.05; DCDG=0.0; DCMG=0.0
      END IF
C
      THR=(TSTAT+VT*DTDV)*THTL                ! THRUST
      CL=CL0+CLA*ALPHA                         ! LIFT COEFF
      CM=DCMG+CM0+CMA*ALPHA+CMDE*ELEV+CL*(XCG-.25) ! PITCH. MOM.
      CD= DCDG+CD0+CDCLS*CL*CL                ! DRAG POLAR
C
      XD(1) = (THR*CALP - QS*CD)/AM - G*SIN(GAM) ! VT RATE
      TEMP = -THR*SALP - QS*CL + AM*(VT*Q+G*COS(GAM))
      XD(2) = TEMP/(AM*VT+QS*CLADOT)           ! ALPHA RATE
      XD(3) = Q                                  ! PITCH RATE
      D   = .5*CBAR*(CMQ*Q+CMADOT*XD(2))/VT    ! DAMPING TERMS
      XD(4) = (QS*CBAR*(CM + D) + THR*ZE)/AIYY   ! PITCH ACCLN.
      XD(5) = VT*(CALP*STH - SALP*CTH)           ! UPWARD SPEED
      XD(6) = VT*(CALP*CTH + SALP*STH)           ! HORIZONTAL SPEED
      AN   = QS*(CL*CALP+CD*SALP)/(G*AM)         ! NORMAL ACCEL.
      RETURN
END

```

Fig. 3.3-1 Computer model of a transport aircraft.

to the descriptions of aerodynamic effects in Chapter 2. This model will be used later for illustrative examples.

A Six-Degrees-of-Freedom Nonlinear Aircraft Model

The mathematical model given here uses the wind-tunnel data from NASA-Langley wind-tunnel tests on a scale model of an F-16 airplane [Nguyen et al., 1979]. The data apply to the speed range up to about $M = 0.6$, and were used in a NASA-piloted simulation to study the maneuvering and stall/post-stall characteristics of a relaxed static-stability airplane.

Because of the application, and the ease of automated data collection, the data cover a very wide range of angle of attack (-20° to 90°), and of sideslip angle (-30° to 30°). However, the present state of the art does not allow accurate dynamic modeling in the poststall region, and in addition the aircraft has insufficient pitching moment control for maneuvering at angles of attack beyond about 25° . Therefore, for use here, we have reduced the range of the data to $-10^\circ \leq \alpha \leq 45^\circ$, and approximated the beta dependence in some cases.

The F-16 has a leading-edge flap that is automatically controlled as a function of alpha and Mach and responds rapidly as alpha changes during maneuvering. In the speed range for which the data are valid, the Mach-dependent variation of the flap is small, so we have eliminated this dependence. Then, neglecting the dynamics of the flap actuator and assuming that the flap is dependent on alpha only, we have merged all of the independent flap data tables into the rest of the tabular aerodynamic data. The effect of the flap deflection limits (but not the rate limits) is still present in the reduced data. These steps have greatly reduced the size of the database and made it feasible to present the data here (Appendix A). The approximate model constructed from these data exhibits steady-state flight trim conditions, and corresponding dynamic modes, that are close to those of the full (50-lookup-table) model.

The F-16 model has been programmed as a FORTRAN subroutine in the same form as the previous model. The code is shown in Fig. 3.3-2; all subroutines and functions called by the model are included in Appendix A. The constants for the geometrical and mass properties are in the DATA statements and correspond to the values given in the appendix. Note that English units have been used here rather than SI units. The constants c_i were defined in equations (2.4-6), and the quantities RM, XCGR, and HE are, respectively, the reciprocal of the aircraft mass, x -coordinate of the reference cg position, and engine angular momentum (assumed constant at 160 slug-ft²/s). The equations are in body-axes form with separate conversions to and from the variables V_T , α , β that have been used instead of U , V , and W for the first three state variables. Subroutine ADC is the same routine as used in the previous model for calculating atmospheric density and hence dynamic pressure.

The aerodynamic force and moment component buildup follows the outline presented in Section 2.3 except that body axes are used. For example, CX(alpha, el) is a function subprogram that computes the nondimensional force coefficient for the body x -axis and is a function of angle of attack and elevator deflection. The total force coefficients for the three axes are CXT, CYT, and CZT. As shown in the appendix, the component functions typically contain a two-dimensional data lookup table and a linear interpolation routine. We have used as much commonality as possible in the data tables and interpolation routines and have provided an interpolator that will also

```

SUBROUTINE F(TIME,X,XD)
REAL X(*), XD(*), D(9)
COMMON/PARAM/XCG
COMMON/CONTROLS/THTL,EL,AIL,RDR
COMMON/OUTPUT/AN,ALAT,AX,QBAR,AMACH,Q,ALPHA
DATA S,B,CBAR,RM,XGCR,HE/300,30,11.32,1.57E-3,0.35,160.0/
DATA C1,C2,C3,C4,C5,C6,C7,C8,C9/- .770, .02755, 1.055E-4,
& 1.642E-6, .9604, 1.759E-2, 1.792E-5, -.7336, 1.587E-5/
DATA RTOD,G / 57.29578, 32.17/

C
C Assign state & control variables
C
VT= X(1); ALPHA= X(2)*RTOD; BETA= X(3)*RTOD
PHI=X(4); THETA= X(5); PSI= X(6)
P= X(7); Q= X(8); R= X(9); ALT= X(12); POW= X(13)

C Air data computer and engine model
C
CALL ADC(VT,ALT,AMACH,QBAR); CPOW= TGEAR(THTL)
XD(13) = PDOT(POW,CPOW); T= THRUST(POW,ALT,AMACH)

C Look-up tables and component buildup
C
CXT = CX (ALPHA,EL)
CYT = CY (BETA,AIL,RDR)
CZT = CZ (ALPHA,BETA,EL)
DAIL= AIL/20.0; DRDR= RDR/30.0
CLT = CL(ALPHA,BETA) + DLDA(ALPHA,BETA)*DAIL
& + DLDR(ALPHA,BETA)*DRDR
CMT = CM(ALPHA,EL)
CNT = CN(ALPHA,BETA) + DNDA(ALPHA,BETA)*DAIL
& + DNDR(ALPHA,BETA)*DRDR

C Add damping derivatives :
C
TVT= 0.5/VT; B2V= B*TVT; CQ= CBAR*Q*TVT
CALL DAMP(ALPHA,D)
CXT= CXT + CQ * D(1)
CYT= CYT + B2V * ( D(2)*R + D(3)*P )
CZT= CZT + CQ * D(4)
CLT= CLT + B2V * ( D(5)*R + D(6)*P )
CMT= CMT + CQ * D(7) + CZT * (XGCR-XCG)
CNT= CNT + B2V*(D(8)*R + D(9)*P) - CYT*(XGCR-XCG) * CBAR/B

C Get ready for state equations
C
CBTA = COS(X(3)); U= VT * COS(X(2)) * CBTA
V= VT * SIN(X(3)); W= VT * SIN(X(2)) * CBTA
STH= SIN(THETA); CTH= COS(THETA); SPH= SIN(PHI)
CPH= COS(PHI); SPSI= SIN(PSI); CPSI= COS(PSI)
QS = QBAR * S; QSB= QS * B; RMQS= RM * QS
GCTH = G * CTH; QSPH= Q * SPH
AY = RMQS*CYT; AZ= RMQS * CZT

C Force equations
C
UDOT = R*V - Q*W - G*STH + RM * (QS * CXT + T)
VDOT = P*W - R*U + GCTH * SPH + AY
WDOT = Q*U - P*V + GCTH * CPH + AZ
DUM = (U*U + W*W)
XD(1) = (U*UDOT + V*VDOT + W*WDOT)/VT
XD(2) = (U*WDOT - W*UDOT) / DUM
XD(3) = (VT*VDOT- V*XD(1)) * CBTA / DUM


---


C Kinematics
C
XD(4) = P + (STH/CTH)*(QSPH + R*CPH)
XD(5) = Q*CPH - R*SPH
XD(6) = (QSPH + R*CPH)/CTH

```

Fig. 3.3-2 Computer model of an F-16 aircraft.

```

C
C Moments
C
XD(7) = (C2*P + C1*R + C4*HE)*Q + QSB*(C3*CLT + C4*CNT)
XD(8) = (C5*P - C7*HE)*R + C6*(R*R-P*P) +QS*CBAR*C7*CMT
XD(9) = (C8*P-C2*R+C9*HE)*Q + QSB*(C4*CLT + C9*CNT)

C
C Navigation
C
T1= SPH * CPSI; T2= CPH * STH; T3= SPH * SPSI
S1= CTH * CPSI; S2= CTH * SPSI; S3= T1 * STH - CPH * SPSI
S4= T3 * STH + CPH * CPSI; S5= SPH * CTH; S6= T2*CPSI + T3
S7= T2 * SPSI - T1; S8= CPH * CTH

C
XD(10) = U * S1 + V * S3 + W * S6 ! North speed
XD(11) = U * S2 + V * S4 + W * S7 ! East speed
XD(12) = U * STH -V * S5 - W * S8 ! Vertical speed

C
C Outputs
C
AN= -AZ/G; ALAT= AY/G
RETURN
END

```

Fig. 3.3-2 (Continued)

extrapolate beyond the limits of the tables. Therefore, a simulation may recover without loss of all data despite exceeding the limits of a lookup table.

Engine Model

The NASA data include a model of the F-16 afterburning turbofan engine, in which the thrust response is modeled with a first-order lag, and the lag time constant is a function of the actual engine power level (POW) and the commanded power (CPOW). This time constant is calculated in the function PDOT, whose value is the rate of change of power, while the state variable X_{13} represents the actual power level. The function TGEAR (throttle gearing) relates the commanded power level to the throttle position (0 to 1.0) and is a linear relationship apart from a change of slope when the military power level is reached at 0.77 throttle setting. The variation of engine thrust with power level, altitude, and Mach number is contained in the function THRUST.

Sign Convention for Control Surfaces

The sign conventions used in the model follow a common industry convention and are given in Table 3.3-1.

Testing the Model

When constructing this model a simple program should be written to exercise each of the aerodynamic lookup tables individually, and plot the data, before

TABLE 3.3-1. Aircraft Control-Surface Sign Conventions

	Deflection	Sense	Effect
Elevator	Trailing edge down	Positive	Negative pitching moment
Rudder	Trailing edge left	Positive	Negative yawing moment, positive rolling moment
Ailerons	Right-wing trailing edge down	Positive	Negative rolling moment

the tables are used with the model. The range of the independent variables should be chosen to ensure that extrapolation is performed correctly, as well as interpolation. A simple check on the complete model can be obtained by writing another program to set the parameter, input, and state vectors to the arbitrarily chosen values given in Table 3.3-2. The resulting values of the derivative vector should then agree with those given in the table.

The next thing we must do with this model is to bring it under control by finding a combination of values of the state and control variables that correspond to a steady-state flight condition. This is quite difficult for us, unlike a real pilot who is constantly receiving visual and other cues, and will be the subject of the next section. In the next section steady-state trim data

TABLE 3.3-2. F-16 Model Test Case

Element	X_{CG}	$U(i)$	$X(i)$	$\dot{X}(i)$
1	0.4	0.9	500	-75.23724
2		20	0.5	-0.8813491
3		-15	-0.2	-0.4759990
4		-20	-1	2.505734
5			1	0.3250820
6			-1	2.145926
7			0.7	12.62679
8			-0.8	0.9649671
9			0.9	0.5809759
10			1,000	342.4439
11			900	-266.7707
12			10,000	248.1241
13			90	-58.68999

will be given for both wings-level, non-sideslipping flight and for turning flight. Therefore, the longitudinal equations can be tested alone before all the equations are brought into play.

3.4 STEADY-STATE TRIMMED FLIGHT

Steady-state flight was defined in Section 2.5 and was shown to require the solution of a set of nonlinear simultaneous equations derived from the state model. Now we are faced with the problem of actually calculating the values of the state and control vectors that satisfy these equations. This cannot be done analytically because of the very complex functional dependence of the aerodynamic data. Instead, it must be done with a numerical algorithm which iteratively adjusts the independent variables until some solution criterion is met. The solution will be approximate but can be made arbitrarily close to the exact solution by tightening up the criterion. Also, the solution may not be unique; for example, steady-state level flight at a given engine power level can in general correspond to two different airspeeds and angles of attack. Our knowledge of aircraft behavior will allow us to specify the required steady-state condition so that the trim algorithm converges on an appropriate, if not unique, solution.

Our goals in this section are to solve the algorithmic and computational problems and to be able to determine steady-state conditions for the two aircraft models in Section 3.3. These flat-Earth models have less restrictive steady-state flight conditions than a round-Earth model and in this sense are more interesting. However, the algorithms that we shall develop can easily be specialized to determine steady-state conditions for round-the-Earth flight. We shall also work with some self-imposed restrictions and thus deal with the state-space model only through its proper inputs and outputs. That is, we shall not work within the model to balance forces and moments separately as is often done within the industry. This is not a significant computational restriction, and it has the important advantage of separating the model from the supporting software, and promoting modular software with standard mathematical algorithms. Figure 3.1-1 illustrates this idea; the generic trim program links to any nonlinear model and produces a file containing the steady-state values of the control and state vectors for use by the time-history and linearization programs.

One of the first things that must be decided is how to specify the steady-state condition, how many of the state and control variables may be chosen independently, and what constraints exist on the remaining variables. A computer program can then be written so that the specification variables are entered from the keyboard, and the independent variables are adjusted by the numerical algorithm that solves the nonlinear equations, while the remaining variables are determined from the constraint equations.

For steady-state flight we expect to be able to specify the altitude and the velocity vector (i.e., speed and climb angle) within the limits imposed by engine power. Then, assuming that the aircraft configuration (i.e., flap settings, landing gear up or down, speed brake deployed, etc.) is prespecified, for a conventional aircraft we expect that a unique combination of the control inputs and the remaining state variables will exist. All of the control variables (THTL, EL, AIL, RDR) enter the model only through tabular aerodynamic data, and we cannot, in general, determine any analytical constraints on these control inputs. Therefore, these four control inputs must be adjusted by our numerical algorithm. This is not the case for the state variables.

Since only the altitude component of the NED position vector is relevant and can be prespecified, we can temporarily eliminate the three position states from consideration. Consider first steady translational flight. The state variables ϕ, P, Q, R are all identically zero, and the orientation ψ can be specified freely; this only leaves V_T, α, β , and θ to be considered. These variables determine the component of the velocity vector in the vertical plane of the NED frame. The sideslip angle cannot be specified freely; it must be adjusted by our trim algorithm to zero out any sideforce. This leaves the variables V_T, α , and θ ; the first two are interrelated through the amount of lift needed to support the weight of the aircraft; therefore, only two may be specified independently (θ , and either V_T or α). We usually wish to impose a flight-path angle (γ) constraint on the steady-state condition, so we shall finally choose to specify V_T and γ .

A nonzero flight-path angle corresponds to an "instantaneous" steady-state condition only, because the atmospheric density changes as the altitude changes. Nevertheless, it is a useful condition since it fixes the rate of climb (ROC); for example, it could be used to determine steady-state rate of climb as a function of other parameters. We shall therefore derive a general rate of climb constraint; this constraint will allow a nonzero bank angle so that it can also be applied to steady-state turning flight.

Steady-state turning flight must now be considered; the variables ϕ, P, Q , and R will no longer be set to zero. The turn can be specified by the Euler angle rate $\dot{\psi}$; this is the rate at which the aircraft's heading changes (the initial heading can still be freely specified). Then, given values of the attitude angles ϕ and θ , the state variables P, Q , and R can be determined from the kinematic equation (1.4-15). The required value of θ can be obtained from the ROC constraint if the value of ϕ is known, and we next consider the determination of ϕ .

The bank angle (ϕ) for the steady-state turn can be freely specified, but then, in general, there will be a significant sideslip angle and the turn will be a "skidding" turn. The pilot will feel a force pushing him or her against the side of the cockpit, the passengers' drinks will spill, and the radius of the turn will be unnecessarily large. In a "coordinated" turn the aircraft is banked at an angle such that there is no component of aerodynamic force along the body y -axis. This condition is used as the basis of the turn coordination

constraint derived below. The turn coordination constraint will be found to involve both θ and ϕ ; therefore, it must be solved simultaneously with the ROC constraint.

The Rate-of-Climb Constraint

In the flat-Earth equations the rate of climb is simply $V_T \sin \gamma$, and this is the negative- z component of the velocity in the NED frame. We can use the coordinate transformation from wind axes to NED axes to obtain the required constraint; thus

$$\begin{bmatrix} * \\ * \\ -V_T \sin \gamma \end{bmatrix} = B_\psi^T B_\theta^T B_\phi^T S^T \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix}. \quad (3.4-1)$$

The asterisks indicate “don’t care” components, and if this equation is expanded and then arranged to solve for θ (Problem 3.4-3), the results are

$$\sin \gamma = a \sin \theta - b \cos \theta, \quad (3.4-2)$$

where

$$a = \cos \alpha \cos \beta, \quad b = \sin \phi \sin \beta + \cos \phi \sin \alpha \cos \beta.$$

Now, solving for θ , we find

$$\tan \theta = \frac{ab + \sin \gamma \sqrt{a^2 - \sin^2 \gamma + b^2}}{a^2 - \sin^2 \gamma}, \quad \theta \neq \pm \frac{\pi}{2}. \quad (3.4-3)$$

The sign of the radical was determined by examining the conditions under which (3.4-3) reduces to the condition $\theta = \alpha + \gamma$ [equation (2.4-22)].

The Turn-Coordination Constraint

In the flat-Earth NED frame the velocity vector is tangential to the turning circle, so the centripetal acceleration, in g 's, is

$$G = \frac{\dot{\psi} V_T}{g'_0}.$$

If we take the lateral (\dot{V}) equation of the nonlinear force equations (2.4-2) and impose the steady-state condition $\dot{V} = 0$ and the coordination condition

$F_y = 0$, we obtain

$$0 = -RU + PW + g'_0 \sin \phi \cos \theta.$$

Now use (1.4-15) to substitute for the body-axes rates in terms of the Euler rates, with $\dot{\phi}, \dot{\theta} = 0$ and $\dot{\psi}$ = turn rate, and use (2.3-5) to substitute for the velocity components U, V , and W . The result is

$$\sin \phi = G \cos \beta (\sin \alpha \tan \theta + \cos \alpha \cos \phi). \quad (3.4-4)$$

This is the required coordination constraint; it can be used in conjunction with (3.4-3) to trim the aircraft for turning flight with a specified rate of climb. If we can now solve (3.4-3) and (3.4-4) simultaneously for the state variables ϕ and θ , our numerical trim algorithm need only vary the states α and β (in addition to the four controls). The simultaneous solution is quite cumbersome but can be shown to be[†]

$$\tan \phi = G \frac{\cos \beta}{\cos \alpha} \frac{(a - b^2) + b \tan \alpha \sqrt{c(1 - b^2) + G^2 \sin^2 \beta}}{a^2 - b^2(1 + c \tan^2 \alpha)}, \quad (3.4-5)$$

where

$$a = 1 - G \tan \alpha \sin \beta, \quad b = \frac{\sin \gamma}{\cos \beta}, \quad c = 1 + G^2 \cos^2 \beta.$$

The value of ϕ given by (3.4-5) can now be used to solve (3.4-3) for θ . Note that when the flight-path angle, γ , is zero, (3.4-5) reduces to

$$\tan \phi = \frac{G \cos \beta}{\cos \alpha - G \sin \alpha \sin \beta}, \quad (3.4-6)$$

and when β is small, this reduces to

$$\tan \phi = \frac{G}{\cos \alpha}. \quad (3.4-7)$$

Equation (3.4-7) is also a good approximation for (3.4-5) in general. This completes the description of the flight-path constraints; a complete trim algorithm using the two constraints will be described next.

The Steady-State Trim Algorithm

The steady-state flight conditions are determined by solving the nonlinear state equations for the state and control vectors that make the state deriva-

[†]This result was derived by A. H. Haddad.

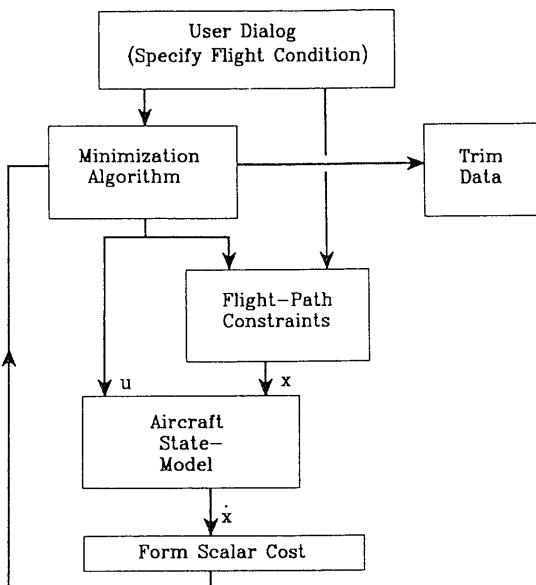


Fig. 3.4-1 Trim algorithm.

tives \dot{V}_T , $\dot{\alpha}$, $\dot{\beta}$, and \dot{P} , \dot{Q} , \dot{R} , identically zero. A convenient way to do this, with a readily available numerical algorithm, is to form a *cost function* from the sum of the squares of the derivatives above. A multivariable numerical optimization algorithm can then be used to adjust the control variables, and appropriate state variables, to minimize this scalar cost. Examples of suitable algorithms are the IMSL routine ZXMD [IMSL, 1980], and the SIMPLEX algorithm [Press et al., 1986]. The simplex algorithm given in Appendix B [Nelder and Mead, 1964] has always performed well on this problem (and many other problems, over a number of years).

Figure 3.4-1 illustrates how the complete trim algorithm may be organized, and Appendix B gives a code nucleus for writing a trim program. The main program contains the dialog to select the required steady-state flight condition, enter the specifications from the keyboard, and file the answers ready for use by other software. The minimization algorithm is a subroutine called from the main program and it requires the scalar cost function as one of its arguments. The cost function connects the optimization algorithm outputs to the four aircraft control variables and two state variables, and then calls the constraint subroutine followed by the aircraft model subroutine. The constraint subroutine computes elements of the state vector using the ROC and coordination constraints.

The trim program is relatively simple to construct depending on the minimization algorithm chosen, and the whole program can be tested with

the constraint algorithm only partially complete (e.g., ROC constraint only, and skidding turns). The constraint routine can also be simplified by eliminating (3.4-5) and allowing the minimization algorithm to tune the bank angle (i.e., seven minimization variables). The bank angle can alternatively be fixed at the approximate value given by (3.4-7). A program TRIM has been written to use the simplex algorithm, and this program will be used to obtain the results needed throughout the rest of the book. We now give a simple trim example using the transport aircraft model.

Example 3.4-1: Longitudinal Trim for the Transport Aircraft Model. In this example we shall obtain a level-flight ($\gamma = 0$) trim for the transport aircraft longitudinal model in Fig. 3.3-1. The constraint equations are therefore not needed. It is only necessary to choose the speed and altitude, fix the pitch rate at zero, and adjust the throttle and elevator controls and the angle of attack. The pitch attitude is set equal to the angle of attack. A suitable cost function is given by

```
FUNCTION COST(S)
PARAMETER (NN=20)
REAL S(*)
COMMON/STATE/X(NN), XD(NN)
COMMON/CONTROLS/THTL, ELEV
THTL = S(1)
ELEV = S(2)
X(2) = S(3)
X(3) = X(2)
CALL F(TIME, X, XD)
COST = XD(1)**2 + 100.0*(XD(2)**2) + 10.0*(XD(4)**2)
RETURN
END
```

The vector S contains the (simplex) variables coming from the minimization algorithm, and the controls and the angle-of-attack states are assigned to these. The fixed states come from the main program through the STATE common block. The call to the aircraft model (F) sets its states and controls and returns the derivative vector, which is used to compute the scalar cost function. Note that different weightings have been placed on the elements of the scalar cost; these are not necessarily optimal and are not at all critical. The main program can be equally simple; it must assign the cg position and the speed, pitch-rate, and altitude states, and call the minimization routine with this cost function. The results obtained with 300 iterations (function calls) of a simplex algorithm, with $x_{cg} = 0.25\bar{c}$, and flaps and landing gear retracted, are shown in Table 3.4-1. Additional iterations will further reduce the final cost function, but negligible changes will occur in the states and controls.

TABLE 3.4-1. Trim Data for the Transport Aircraft Model

Altitude (ft)	Speed (ft/s)	Initial cost	Final cost	Throttle	Elevator (deg)	Alpha (deg)
0	170	28.9	7.4E-15	0.297	-25.7	22.1
0	500	3.54	3.0E-15	0.293	2.46	0.580
30,000	500	10.8	5.4E-15	0.204	-4.10	5.43

The trim program for Example 3.4-1 executes very quickly on a small computer because of the simple aircraft model involved; it is therefore convenient for other experiments, such as trimming for a specific alpha by varying the airspeed (Problem 3.4-4). We next consider the more difficult problem of trimming a 6-DOF model, with additional dynamics such as the engine model that must also be put into a steady-state condition. This will be illustrated with the F-16 model.

Figure 3.4-2 shows a cost function subprogram for the F-16 model and illustrates an important point. This cost function is specific to the F-16 model because of the assignment statement for X_{13} . An examination of the F-16 model will show that this statement sets the derivative \dot{X}_{13} to zero and hence puts the engine dynamics into the steady state. Any other dynamics in the aircraft model besides the rigid-body dynamics must be put into the steady-state condition in this way. In our original, large, F-16 model this was done for the leading-edge flap actuator and its phase-lead network. The cost function routine is the only part of the trim software that needs to be tailored to the specific aircraft.

In this cost function, unlike the previous case, the state variables X_4 through X_9 (excluding X_6) are continually assigned new values in the constraint routine CONSTR. This routine implements the rate-of-climb and turn coordination constraints that were derived earlier. In the cost the

```

FUNCTION SF16(S)
PARAMETER (NN=20)
REAL S(*)
COMMON/STATE/X(NN),XD(NN)
COMMON/CONTROLS/THTL,EL,AIL,RDR
THTL= S(1)
EL = S(2)
X(2)= S(3)
AIL = S(4)
RDR = S(5)
X(3)= S(6)
X(13)= TGEAR(THTL)
CALL CONSTR(X)
CALL F(TIME,X,XD)
SF16= XD(1)**2 + 100*(XD(2)**2 + XD(3)**2) +
& 10*(XD(7)**2 + XD(8)**2 + XD(9)**2)
RETURN
END

```

Fig. 3.4-2 A cost function for trimming the F-16 model.

aerodynamic-angle rates $\dot{\alpha}$ and $\dot{\beta}$ have been weighted the most heavily, the angular rate derivatives \dot{P} , \dot{Q} , \dot{R} have medium weights, and the derivative \dot{V}_T has the least weight. Again, the weights are uncritical. The TRIM program can now be used on the F-16 model; two examples are given and then some trim conditions are derived for use later.

Example 3.4-2: The F-16 Trimmed Power Curve. In this example the variation of the angle of attack, the elevator deflection, and in particular the throttle setting, will be examined in trimmed level flight conditions over a range of speed. The F-16 is balanced to minimize trim drag, and for straight and level flight across the speed range of our model, the change in the trimmed elevator deflection is very small and varies erratically. The required throttle setting increases as transonic speeds are approached because of the increasing drag.

At very low speeds (and low dynamic pressure) a high value of the lift coefficient is needed to support the aircraft weight. This causes high induced drag, and because of the large angle of attack, the engine thrust must support a large component of the aircraft weight. Therefore, the throttle setting must increase again at low speeds, and the throttle setting versus speed curve must pass through a minimum.

Data for trimmed level flight at sea level, with the nominal cg position, are given in Table 3.4-2. As the speed is lowered, the angle of attack increases, the leading-edge flap reaches its limit (at about $\alpha = 18^\circ$, although no longer visible in the data), and the trimmed throttle setting begins to increase from its very low value. The model can be trimmed until alpha reaches about 45° , when a rapid increase in trimmed elevator deflection occurs, quickly reaching the deflection limit.

Figure 3.4-3 shows throttle setting plotted against airspeed. This curve is not the same as the airplane "power-required" curve because the engine characteristics are also included. Nevertheless, we shall loosely refer to it as the *power curve*. It illustrates the fact that there is a speed at which the throttle setting passes through a minimum. For a propeller-driven plane this is the condition for *best endurance* (but not best range) at the given altitude. For a jet plane the fuel consumption is more strongly related to thrust than

TABLE 3.4-2. Trim Data for the F-16 Model

Speed	130	140	150	170	200	260	300	350	400
Thrtl.	0.816	0.736	0.619	0.464	0.287	0.148	0.122	0.107	0.108
AOA	45.6	40.3	34.6	27.2	19.7	11.6	8.49	5.87	4.16
Elev.	20.1	-1.36	0.173	0.621	0.723	-0.090	-0.591	-0.539	-0.591
Speed	440	500	540	600	640	700	800	ft/s	
Thrtl.	0.113	0.137	0.160	0.200	0.230	0.282	0.378	per unit	
AOA	3.19	2.14	1.63	1.04	0.742	0.382	-0.045	degrees	
Elev.	-0.671	-0.756	-0.798	-0.846	-0.871	-0.900	-0.943	degrees	

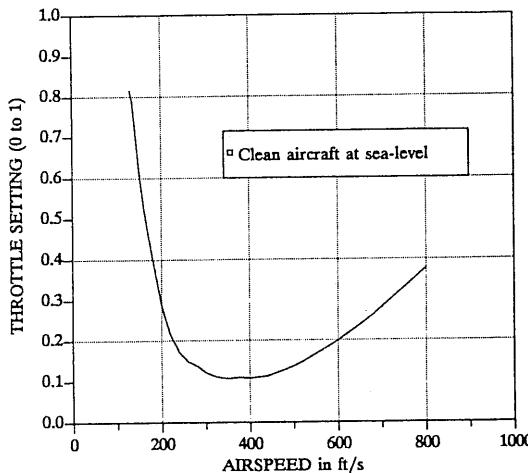


Fig. 3.4-3 Power curve for the F-16 model.

power, so this is no longer true. For more details on the static performance information that can be derived from a power curve, see Dommasch et al. [1967].

The region to the left of the minimum of the power-required curve is known as the *back side of the power curve*. If the aircraft is operating on the back side of the power curve, opening the throttle produces an increase in altitude, *not* an increase in speed. The speed is then controlled by the elevator. This region of operation may be encountered in the landing phase of flight (e.g., carrier landings). ■

Example 3.4-3: Trim Conditions for a Coordinated Turn. This example illustrates the use of the trim algorithm to determine the steady-state conditions for a coordinated turn, performed with the F-16 model. The cg location of the model is at $0.35\bar{c}$, and the aircraft dynamics are unstable in pitch at the chosen flight condition. The turn would stress a pilot quite severely since it involves a sustained normal acceleration of $4.5 g's$.

The trim program dialogue and keyboard inputs are shown in Figure 3.4-4 as they would appear on a terminal display. Note that entering a "/" in response to a FORTRAN read statement causes the program to use the last values assigned to the variable. This allows the minimization to be picked up from where it was left off if the final cost function was not low enough. In the run shown, the cost function was reduced by almost 10 orders of magnitude after 1000 function calls. Execution is very fast, and this is a reasonable number of calls.

The cost function can always be reduced to 1×10^{-10} or less; lower values are useful simply for checking consistency of results. The most effective way

```

? Altitude (ft) : 0
? Air Speed (ft/s) and Climb Angle (deg) : 502,0
? Roll, Pull-Up, and Turn rates (rad/s) : 0,0,.3

Turn Radius (ft) = 1.6733E+03 Approx Bank Angle (deg) = 77.94
? Coordinated Turn (def. = Y) : /
? Required No. of Trim Iterations (def. =1000) : /

? Guess : Throttle, Elevator, Ailerons, Rudder : /
Computed : 8.35E-01 -1.48E+00 9.54E-02 -4.11E-01

Angle of Attack 1.37E+01 Sideslip Angle 2.92E-02
Pitch Angle 2.87E+00 Bank Angle 7.83E+01
Normal Acceleration 4.65E+00 Lateral Accln. -5.02E-06
Dynamic Pressure 3.00E+02 Mach Number 4.50E-01

Initial Cost Function 1.85E+01, Final Cost Fn. 3.98E-09
? More Iterations (def = Y) : N

? Enter "M" to modify this trim
    "R" to restart
    "/" to file data/quit : /

? Name of Output File (def= None) : /

? Enter "M" for Menu
    "/" to quit : /

```

Fig. 3.4-4 Input and output data of the trim program.

to use the simplex algorithm is to perform 500 to 1000 iterations, and if the cost is not acceptable, reinitialize the step size of the minimization algorithm before each new set of iterations. More trim iterations were later performed on this example and the cost function reached a lower limit of 5.52×10^{-13} (the trim program and model use only single-precision arithmetic); no significant changes occurred in the numerical values given above. The final state and control vectors placed in the output file were

$$\begin{aligned}
X_1 &= 5.020000E + 02, & X_2 &= 2.392628E - 01, & X_3 &= 5.061803E - 04, \\
X_4 &= 1.366289E + 00, & X_5 &= 5.000808E - 02, & X_6 &= 2.340769E - 01, \\
X_7 &= -1.499617E - 02, & X_8 &= 2.933811E - 01, & X_9 &= 6.084932E - 02, \\
X_{10} &= 0.000000E + 00, & X_{11} &= 0.000000E + 00, & X_{12} &= 0.000000E + 00, \\
X_{13} &= 6.412363E + 01,
\end{aligned}$$

$$\begin{aligned}
U_1 &= 8.349601E - 01, & U_2 &= -1.481766E + 00, & U_3 &= 9.553108E - 02, \\
U_4 &= -4.118124E - 01.
\end{aligned}$$

This trim will be used for a flight simulation example in the next section and in Section 3.6 to illustrate coupling effects in the aircraft dynamics. ■

Trimmed Conditions for Studying Aircraft Dynamics

The steady-state performance of an airplane can be investigated very thoroughly from a set of trimmed flight conditions. The specific fuel consumption,

TABLE 3.4-3. Trimmed Flight Conditions for the F-16 Model^a

Variable	Nominal	Condition			
		$x_{cg} = 0.3\bar{c}$	$x_{cg} = 0.38\bar{c}$	$\dot{\psi} = 0.3 \text{ rad/s}$	$\dot{\theta} = 0.3 \text{ rad/s}$
V_T (ft/s)	502.0	502.0	502.0	502.0	502.0
α (rad)	0.03691	0.03936	0.03544	0.2485	0.3006
β (rad)	-4.0E - 9	4.1E - 9	3.1E - 8	4.8E - 4	4.1E - 5
ϕ (rad)	0	0	0	1.367	0
θ (rad)	0.03691	0.03936	0.03544	0.05185	0.3006
P (rad/s)	0	0	0	-0.01555	0
Q (rad/s)	0	0	0	0.2934	0.3000
R (rad/s)	0	0	0	0.06071	0
Thl (0-1)	0.1385	0.1485	0.1325	0.8499	1.023
El (deg)	-0.7588	-1.931	-0.05590	-6.256	-7.082
Ail (deg)	-1.2E - 7	-7.0E - 8	-5.1E - 7	0.09891	-6.2E - 4
Rdr (deg)	6.2E - 7	8.3E - 7	4.3E - 6	-0.4218	0.01655

^aNominal condition: $h = 0$ ft, $\bar{q} = 300$ psf, $x_{cg} = 0.35\bar{c}$, $\dot{\phi} = \dot{\theta} = \dot{\psi} = \gamma = 0$.

rate of climb, various critical speeds for takeoff and landing, radius of turn, and so on, can all be determined for a number of different flight conditions. We have not provided enough modeling detail for all of these investigations, but the model and the trim program could be further developed if required.

Table 3.4-3 presents a set of trimmed conditions for the F-16 model; these will be used for the simulation examples in this chapter, and for controller design in subsequent chapters. The F-16 model aerodynamic data were referenced to the $0.35\bar{c}$ cg x -position, and this is the “nominal” position for the cg. The nominal speed and altitude were chosen to give a representative flight condition suitable for later examples and designs. The table contains data for the nominal condition, a forward-cg condition, an aft-cg condition, and steady-state turn and pull-up conditions with a forward cg. The forward and aft-cg cases have been included to demonstrate later the effect of cg position on stability. A forward cg-location has been used for the two maneuvering cases so that the effects of the maneuver can be illustrated without the additional complication of unstable dynamics.

3.5 NUMERICAL SOLUTION OF THE STATE EQUATIONS

To simulate a flight using the aircraft model, we must solve the differential equations that govern the motion. The equations are nonlinear, depend on experimental data, and are subjected to arbitrary input signals. Therefore, an analytical solution is out of the question, and numerical methods must be used to calculate an aircraft trajectory. The term *numerical integration* is

often reserved for numerical methods of evaluating a definite integral (e.g., Simpson's rule). However, for want of an equally concise name, we will also use it to mean numerical solution of ordinary differential equations (ODEs).

In general, a *trajectory* will mean the motion of the tip of the state vector in the state coordinate system as time evolves; in a more limited sense it also means the motion of the aircraft in some three-dimensional coordinate frame as time evolves. The state trajectory will change in a continuous manner because the state variables describe the energy stored in a physical system, and an instantaneous change in energy would require infinite power. This continuous variation of the state makes numerical integration possible, although state derivatives can often change discontinuously, and this may cause difficulties with a particular integration algorithm.

Numerical evaluation of the continuous trajectory implies calculating discrete sequential values of the state

$$X(t_0 + kT), \quad k = 0, 1, 2, \dots \quad (3.5-1a)$$

that satisfy the state equations

$$\dot{X}(t) = f(X(t), U(t)), \quad (3.5-1b)$$

where t_0 is the initial time and T is the *time step*. It is convenient to use a fixed basic time step for a given simulation, but the integration algorithm may internally subdivide the basic time step as it attempts to obtain an accurate solution.

Next, we must consider the fact that the state equations are not autonomous since the control input is an external input. In order to use standard numerical integration algorithms, it is necessary to assume that the time step is so small that the control input remains constant within the *sample interval* kT to $(k + 1)T$. This parallels the practical situation of an aircraft with a digital flight control computer, where the pilot's control inputs are sampled and held constant during the sample period by a *zero-order-hold circuit*, in order that they may be digitized. Later we shall show that for linear state equations we can calculate values of the state vector using higher-order data holds that assume a general polynomial type of input signal variation during the sample interval.

Numerical integration of the continuous state equations is an initial-value problem; the initial value $X(t_0)$ is given and the "future" values given by (3.5-1a) must be calculated. This can also be called "prediction" of future states based on current values of the first derivative. It is therefore intimately related to the Taylor series, which links prediction to knowledge of the derivatives of all orders, and we will use the Taylor series to derive some basic algorithms. One class of algorithms that will be needed, the *Runge-Kutta algorithms*, directly solve the initial-value problem. They use the set of continuous state equations to predict the state derivatives within the sample interval, at time instants beyond the current time. Another class,

the *linear multistep methods*, require past values of the state. They are therefore not self-starting and do not directly solve the initial-value problem. However, they do not require values of the state derivatives within the sample interval and will therefore be useful for integrating the discrete-time signals in combined continuous-discrete simulations. Numerical integration is not limited to linear systems, as is the Laplace transform, and it is our single most important tool. Therefore, we will briefly review the theory and derive some algorithms, before applying them to aircraft simulation.

Numerical Integration Techniques

To understand some of the techniques of numerical integration, we shall consider a scalar state equation (only one state variable); the algorithms can easily be extended to the vector case. Therefore, consider the simplest ODE initial-value problem, of a single first-order autonomous differential equation with a specified boundary condition

$$\frac{dx}{dt} = f(x, t), \quad x(t_0) = x_0. \quad (3.5-2)$$

Ideally, we would like to have a function that satisfies this differential equation and the boundary condition, that is, a *solution function*. This would provide both numerical values for $x(t)$ and insight into its general behavior. In general, it will be impossible to determine solution functions explicitly for the aircraft state equations, and we must be content with numerical values.

The problem of finding the discrete solution values for (3.5-2) has an obvious connection to the Taylor series

$$x(t_0 + T) = x(t_0) + T\dot{x}(t_0) + \frac{T^2}{2!}\ddot{x}(t_0) + \dots \quad (3.5-3)$$

Computation of the Taylor series derivatives (if they exist) up to appropriate orders, for arbitrary cases, is not an efficient approach to the problem. The Runge-Kutta (RK) methods of numerical integration can be related to the Taylor series but have the advantage of only requiring evaluation of the first derivative, as given by (3.5-2). The simplest RK method is *Euler integration*, which merely truncates the Taylor series after the first derivative. The Euler formula applied to (3.5-2) is therefore

$$x_E(t_0 + T) \approx x(t_0) + Tf(x(t_0), t_0). \quad (3.5-4)$$

This formula is obviously not very accurate, and furthermore, it can easily be improved upon.

A simple method of improving on the Euler scheme is to use *trapezoidal integration*. In this method an estimate of the function derivative at the end

of the time-step is obtained from the Euler formula; then the average of the derivatives at the beginning and end of the time step is used to make a more accurate Euler step. The connection with the formula for the area of a trapezium (not a trapezoid!) will be obvious from the following equations. In these equations the subscripts E and T are used to indicate, respectively, Euler and trapezoidal integration, and the time t can be any one of the discrete-time instants in (3.5-1a). The equations are

$$\begin{aligned}x_E(t+T) &= x(t) + Tf(x(t), t) \\ \dot{x}_E(t+T) &= f(x_E(t+T), t+T) \\ x_T(t+T) &= x(t) + \frac{T}{2}[\dot{x}(t) + \dot{x}_E(t+T)].\end{aligned}\tag{3.5-5}$$

Now for reasons that will soon become clear, these equations are commonly written as

$$\begin{aligned}k_1 &= Tf(x, t) \\ k_2 &= Tf(x + k_1, t + T) \\ x_T(t+T) &= x(t) + \frac{1}{2}(k_1 + k_2).\end{aligned}\tag{3.5-6}$$

This algorithm can be shown to agree with the first three Taylor series terms, that is, up to and including the second derivative term. Therefore, this trapezoidal integration formula is said to be of order two and gives an improvement in accuracy over the Euler first-order method.

Runge-Kutta Algorithms

RK algorithms are an extension of (3.5-6) to higher orders, and the general form is

$$\begin{aligned}k_1 &= Tf(x, t) \\ k_2 &= Tf(x + \beta_1 k_1, t + \alpha_1 T) \\ k_3 &= Tf(x + \beta_2 k_1 + \beta_3 k_2, t + \alpha_2 T) \\ k_4 &= Tf(x + \beta_4 k_1 + \beta_5 k_2 + \beta_6 k_3, t + \alpha_3 T) \\ &\dots \\ x_{RK}(t+T) &= x(t) + \gamma_1 k_1 + \gamma_2 k_2 + \gamma_3 k_3 + \dots.\end{aligned}\tag{3.5-7}$$

Implicit RK algorithms also exist, wherein a coefficient k_i occurs on both sides of one of the equations above. The constants α_i , β_i , and γ_i are chosen so that a particular RK scheme agrees with the Taylor series to as high an order as possible. A great deal of algebraic effort is needed to derive higher-order (greater than four) RK algorithms, and the constants are not

unique for a given order. An algorithm that dates from the end of the nineteenth century, and is still popular, is "Runge's fourth-order rule," which uses the constants

$$\begin{aligned}\alpha_1 &= \alpha_2 = \beta_1 = \beta_3 = \frac{1}{2} \\ \alpha_3 &= \beta_6 = 1 \\ \beta_2 &= \beta_4 = \beta_5 = 0 \\ \gamma_1 &= \gamma_4 = \frac{1}{6}, \quad \gamma_2 = \gamma_3 = \frac{1}{3}.\end{aligned}\tag{3.5-8}$$

In this case only one previous k value appears in each of the k -equations in (3.5-7), thus making a simpler algorithm. This algorithm has been used for many of our examples, and FORTRAN code for the general case of n simultaneous state equations in vector formulation is given in Appendix B.

An important feature of the RK methods is that the only value of the state vector that is needed is the value at the beginning of the time step; this makes them well suited to the ODE initial value problem. The amount of computation involved is governed by the number of derivative evaluations using the state equations, performed during each time step. The number of derivative evaluations depends on the order chosen. For example, a fourth-order RK algorithm cannot be achieved with fewer than four derivative evaluations. For a given overall accuracy in a time response calculation, there is a trade-off between many small steps with a low-order method, and fewer setups but more derivative evaluations with a higher-order method. This led mathematicians to consider the problem of estimating the error in the computed solution function at each time step. Such an error estimate can be used to control the step size automatically in order to meet a specified accuracy. Algorithms that combine RK integration with error estimation include Runge-Kutta-Merson (RKM), Runge-Kutta-England, and Runge-Kutta-Gill; computer codes are commonly available. In terms of (3.5-7) the coefficients for the RKM scheme, for example, are

$$\begin{aligned}\alpha_1 &= \beta_1 = \frac{1}{3} \\ \alpha_2 &= \frac{1}{3}, \quad \beta_2 = \beta_3 = \frac{1}{6} \\ \alpha_3 &= \frac{1}{2}, \quad \beta_4 = \frac{1}{8}, \quad \beta_5 = 0, \quad \beta_6 = \frac{3}{8} \\ \alpha_4 &= 1, \quad \beta_7 = \frac{1}{2}, \quad \beta_8 = 0, \quad \beta_9 = -\frac{3}{2}, \quad \beta_{10} = 2 \\ \gamma_1 &= \frac{1}{6}, \quad \gamma_2 = \gamma_3 = 0, \quad \gamma_4 = \frac{2}{3}, \quad \gamma_5 = \frac{1}{6},\end{aligned}\tag{3.5-9}$$

and the estimated error is

$$E \approx \frac{1}{30}[2k_1 - 9k_3 + 8k_4 - k_5].$$

Linear Multistep Methods

Besides the RK methods, there is a large and important class of integration algorithms in which the computed value of the solution function is a linear combination of past values of the function and its derivatives. These are called *linear multistep methods* (LMMs) and they fit the linear difference equation

$$x(n+1) = \sum_{r=0}^n \alpha_r x(n-r) + T \sum_{r=-1}^n \beta_r \dot{x}(n-r), \quad (3.5-10)$$

where $x(i)$ indicates the value of x at time iT , with i an integer. If β_{-1} is nonzero, the algorithm is an implicit algorithm because the solution $x(n+1)$ is needed to evaluate $\dot{x}(n+1)$ on the right-hand side. Otherwise, the algorithm is explicit. The implicit equation must be solved at each time step. LMMs can be designed to require less computation than RK methods because a number of past values can be kept in storage as the computation proceeds. Because of the requirements for past values, the LMMs are not self-starting, and an RK method, for example, could be used to generate the starting values.

The LMM algorithms may be created in a number of different ways. For instance, if the scalar state equation (3.5-2) is written as an integral equation over the time interval nT to $(n+k)T$, the result is

$$x(n+k) = x(n) + \int_{nT}^{(n+k)T} f(x, t) dt. \quad (3.5-11)$$

There are many finite-difference formulae for evaluating a definite integral, and this approach leads to the *Newton-Coates integration formulae* [Isaacson and Keller, 1966; Ralston, 1965]. Two examples are

$$x(n+1) = x(n-1) + 2T\dot{x}(n) \quad (3.5-12a)$$

$$x(n+1) = x(n-1) + \frac{T}{3}[\dot{x}(n+1) + 4\dot{x}(n) + \dot{x}(n-1)]. \quad (3.5-12b)$$

The first formula uses the *midpoint rule* for the area represented by the integral and is explicit, while the second uses *Simpson's rule* and is implicit. Implicit and explicit formulae can be used together in a *predictor-corrector algorithm* [Hamming, 1962]. The explicit formula is the predictor, used to obtain an approximate value of the solution; and the implicit formula is the corrector equation, which is solved (by iteration) to obtain a more accurate solution.

LMMs of any order can be derived directly from (3.5-10). When $\alpha_r \equiv 0$ for $r > 0$, the *Adams-Basforth-Moulton* (ABM) formulae are obtained, and we

now give two examples. Assume that (3.5-10) has the terms

$$x(n+1) = \alpha_0 x(n) + T[\beta_0 \dot{x}(n) + \beta_1 \dot{x}(n-1)]. \quad (3.5-13)$$

Now write Taylor series expansions for the terms that are not taken at time nT :

$$\begin{aligned} x(n+1) &= x(n) + T\dot{x}(n) + \frac{T^2}{2!}\ddot{x}(n) + \dots \\ \dot{x}(n-1) &= \dot{x}(n) - T\ddot{x}(n) + \frac{T^2}{2!}\dddot{x}(n) - \dots \end{aligned}$$

Substitute these expressions in (3.5-13) and equate powers of T on both sides of the resulting equation; this gives

$$\begin{aligned} T^0: \quad 1 &= \alpha_0 \\ T^1: \quad 1 &= \beta_0 + \beta_1 \\ T^2: \quad \frac{1}{2} &= -\beta_1. \end{aligned}$$

Therefore, (3.5-13) yields the second-order ABM formula

$$x(n+1) = x(n) + \frac{T}{2}[3\dot{x}(n) - \dot{x}(n-1)]. \quad (3.5-14)$$

This requires only one state equation evaluation per time step. The higher-order methods also require only one derivative evaluation per time step, and the third-order ABM is

$$x(n+1) = x(n) + \frac{T}{12}[23\dot{x}(n) - 16\dot{x}(n-1) + 5\dot{x}(n-2)]. \quad (3.5-15)$$

The implicit formulae may be derived in the same way; they give improved accuracy and can also provide an error estimate. They are commonly used in the predictor-corrector form, and this requires two derivative evaluations per step.

Stability, Accuracy, and Stiff Systems

In developing numerical algorithms it is always necessary to consider how computational errors are magnified. If, in pursuit of greater accuracy, one blindly attempts to create higher-order LMM formulae, it is quite possible that the algorithm will be unstable and errors will grow with time. Stability can be determined by analyzing a finite-difference equation associated with

the integration algorithm. This analysis is beyond our scope [Shampine and Gordon, 1975] and we simply note that the specific algorithms described above are stable.

The RK stability properties are different from those of the LMMs. In the case of the RK algorithms, a reduction in step size will eventually eliminate an instability, although the required step size may be unrealistically small. Example 3.5-1 (following) is an example of a reduction in step size eliminating an instability. When a set of state equations is being integrated, the required step size will be determined by the smallest time constant (i.e., the fastest component) of the solution function. A system with a very wide spread of time constants is known as a *stiff system*, and a very large number of RK steps may be necessary to yield only a small part of the complete solution. Other techniques are required for stiff systems (see below).

Choice of Integration Algorithm

The most important feature of the RK methods is that they directly solve the initial value problem. That is, no past values are needed to start the integration. This, of course, exactly matches the philosophy of the state-space formulation in which all of the information describing the "state" of the system is contained in the state vector at any given time instant. The full significance of these facts can only be appreciated when a simulation containing discrete events is considered. This is a common practical engineering situation. For instance, at a given time a new subsystem may be activated, or at a certain value of some variable, the equations of motion may change because limiting or saturation behavior occurs. Consequently, previous states are less relevant; the information they carry may now apply to only a part of the complete system. This favors the RK methods over the multistep methods, and we shall return to these points later. The disadvantages of the RK methods are that the error expressions are complex, they are inefficient when dealing with stiff systems, and more derivative evaluations are required for a given order than is the case with LMMs. The tremendous increases in computing power in recent years have made these disadvantages much less significant for small to medium-sized simulations. Such simulations are commonly run with a fixed time step that has been found (by trial and error) to be adequate for the required accuracy and is also determined by other discrete event considerations.

The important features of LMMs are that higher-order methods are obtained for a given number of derivative evaluations, and an accurate expression for the integration error can usually be obtained. These methods come into their own on very large systems of equations, large stiff systems, and when there is no hard-limiting behavior or topological changes due to switching. The software package ODEPACK [Hindmarsh, 1982] is available

for large and stiff problems, and it handles equations in standard explicit form or in linearly implicit form. For nonstiff problems it uses the implicit ABM methods, and for stiff problems it uses a backward difference formula and improves on the Gear algorithms [Gear, 1971] that have long been used for stiff systems. These algorithms have been used on atmosphere models with more than 10,000 simultaneous ODEs; the spread of time constants in the problem ranged from milliseconds to years, thus making the equations extremely stiff.

Flight Simulation

We now illustrate how non-real-time simulation can be performed, and we apply the technique to the aircraft models to obtain the time history of a trimmed steady-state flight. The steady-state trim condition from the preceding subsection will be used, and the integration algorithm will be the fourth-order Runge-Kutta algorithm given in Appendix B3.

Figure 3.5-1 shows how a non-real-time simulation program may be organized. Two separate subroutines are needed for the dynamic models. One subroutine, F, contains the continuous-time state equations, and another subroutine, D, contains discrete-time commands and any discrete-time algorithms used for digital control. Simulation time is controlled by a DO-loop, and time progresses in increments of size DT. The *time step* or *sampling interval*, DT, is usually chosen to match the highest sampling rate in the digital controller. The numerical integration routine may use this same time step or, if insufficient accuracy is achieved, it may be arranged to subdivide the sampling interval into smaller steps.

```

READ(*)          ! initial conditions and time scales
READ(*)          ! variables to be recorded
N=               ! calculate no. of time steps
NS=              ! calculate interval for recording data
FLAG=.TRUE.      ! flag for initialization
DO I=0,N         ! start simulation
TIME=I*dt + TZERO
CALL D(FLAG,    ) ! initialization & discrete time algorithms
IF(MOD(I,NS)=0)WRITE ! write to terminal or output array
CALL RK(F,     ) ! numerical integration of F(X,U)
END DO
OPEN(*)          ! open files for plot data
WRITE(*)        ! dump output array to disc file
END

```

Fig. 3.5-1 Organization of a time-history program.

If we let the continuous and discrete-time state vectors be denoted, respectively, by X and Z , then when subroutine D is called at $t = 0$ it must establish the initial conditions X_0 , Z_0 , and the initial command input R_0 . The numerical integration algorithm (shown as "RK") is called next, with subroutine F as one of its arguments. Using these initial conditions it determines X_1 (i.e., $X(T)$), the value of the continuous-time state vector at the end of the first time step. During the next pass through the DO-loop the initialization process is bypassed (by setting a software flag) and subroutine D calculates the discrete state Z_1 from Z_0 , X_1 , and R_1 . Thus subroutine D may be used to solve *difference equations* of the form

$$Z_i = g(Z_{i-1}, Z_{i-2}, \dots, Z_{i-p}, X_i, X_{i-1}, \dots, X_{i-n}, R_i, R_{i-1}, \dots, R_{i-m})$$

where g is, in general, a nonlinear function of the states and commands.

In subroutine D some provision must be made to save values of Z , X , and R that are more than one time step old (values from the last time step are available until overwritten), depending on the values of p , n , and m used by the above difference equation. Now consider the subroutine F again; at any time, iT , the state X_i is available from the last sample time, and a control input $h(Z_i, R_i)$ is available from D. Therefore, the derivative $\dot{X}_i = f(X_i, h(Z_i, R_i))$ can be evaluated in subroutine F, and the next state X_{i+1} can be calculated by the numerical integration routine.

In general it is not easy to set up a simulation so that the timing of the computations and the age of the data match the behavior of real hardware. In this simulation scheme the order in which the discrete and continuous state equations are executed, time is updated, and data are recorded allows a simple simulation program that will accommodate a wide range of problems. For example, at time iT , the current discrete state, Z_i , is allowed to depend on the current value of the continuous state vector, X_i , in addition to the past values of Z and X . Note also that a digital controller with multiple sample rates can be accommodated by using modulo counters in D, provided that a common factor can be found for the various sampling periods. Pure delays associated with digital computations can be accommodated by making them multiples of the basic time step.

A simple simulation scheme, in which the integration time step is constant and the discrete-state changes are synchronized with the numerical integration, is adequate for virtually all non-real-time dynamic simulations. However, if discrete-state changes occur aperiodically, it is possible to arrange for the numerical integration algorithms to determine the continuous state vector at the time of a discrete event and to integrate forward from that point to the next discrete event. This *event-driven* simulation is not difficult to implement, but is outside our present requirements. A simulation program, TRESP, was written using the structure shown in Fig. 3.5-1 and some

examples of flight simulation will now be given using this program in the manner shown in Fig. 3.1-1.

Example 3.5-1: Simulation of a Coordinated Turn. This example is a time-history simulation of a steady-state coordinated turn, using the trim data from Example 3.4-3. In Figure 3.5-2 the simulation data are presented as they would appear on the terminal display. The aircraft is turning at 0.3 rad/s, and therefore turns through 54 rad or about 8.6 revolutions in the 180-s simulation. Figure 3.5-3 shows the ground track of the aircraft and shows that the eight circles fall exactly over each other. In Section 3.7 we shall see that the aircraft dynamics have quite a wide spread of time constants, and in this flight condition, there is an unstable mode with a time constant of about 1.7 s. Unless the integration time step is reduced below about 0.02 s, the fourth-order Runge-Kutta routine eventually diverges when integrating this example.

```

Initial condition filename ? (def.= none): EX343

Number of states and outputs to be recorded ?
# States (def.= 0) : 2
Which ones ? : 10 11
# Outputs (def.= 0) : /
Variable (V) or fixed (F) step integration ? (def= F) : /
Length of run (sec) ? : 180
Print time-interval (on screen) ? : 10
Plotting time-interval ? : .5
Sample period (integration step) ? : .01

      TIME          X-10          X-11
0.00E+00  0.00E+00  0.00E+00
1.00E+01  2.36E+02  3.33E+03
2.00E+01 -4.68E+02  6.65E+01
3.00E+01  6.90E+02  3.20E+03
4.00E+01 -8.97E+02  2.61E+02
5.00E+01  1.09E+03  2.94E+03
6.00E+01 -1.26E+03  5.68E+02
7.00E+01  1.40E+03  2.59E+03
8.00E+01 -1.51E+03  9.62E+02
9.00E+01  1.60E+03  2.16E+03
1.00E+02 -1.65E+03  1.41E+03
1.10E+02  1.67E+03  1.70E+03
1.20E+02 -1.66E+03  1.89E+03
1.30E+02  1.61E+03  1.22E+03
1.40E+02 -1.53E+03  2.35E+03
1.50E+02  1.42E+03  7.87E+02
1.60E+02 -1.28E+03  2.76E+03
1.70E+02  1.11E+03  4.22E+02
1.80E+02 -9.21E+02  3.07E+03

Enter  0 to file data           1 to quit
      2 to restart             3 to pick other states
      4 to change integration   5 to change run time  :

```

Fig. 3.5-2 Simulation of a coordinated turn.

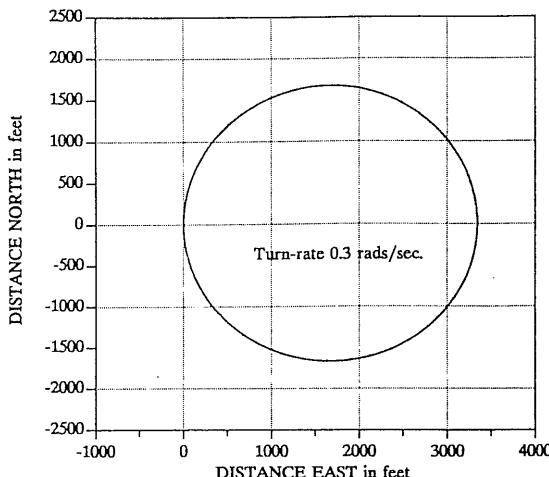


Fig. 3.5-3 Ground track from the coordinated-turn simulation.

Example 3.5-2: Simulated Response to an Elevator Pulse. This example is a time-history simulation of the F-16 longitudinal response to elevator deflection. The trim conditions are the second set of conditions given in Table 3.4-3 (straight and level, with $\bar{q} = 300$ psf and $x_{cg} = 0.3\bar{c}$), and the time-history program was run with an integration step of 0.01 s. The elevator input was a doublet pulse, that is, a pulse that is symmetric about the reference level (the elevator trim setting). It was applied using the discrete-time routine D, and the pertinent D is shown below.

```

SUBROUTINE D(TIME, TS, X, XD, DFLAG)
DIMENSION X(*), XD(*)
COMMON/CONTROLS/THTL, EL, AIL, RDR
LOGICAL DFLAG
C
IF (DFLAG) THEN
  SAVE= EL
  RETURN
END IF
IF (TIME .LT. 1.0) THEN
  RETURN
ELSE IF (TIME .LT. 1.5) THEN
  EL= SAVE + 2.0
ELSE IF (TIME .LT. 2.0) THEN
  EL= SAVE - 2.0
ELSE
  EL= SAVE
END IF
RETURN
END

```

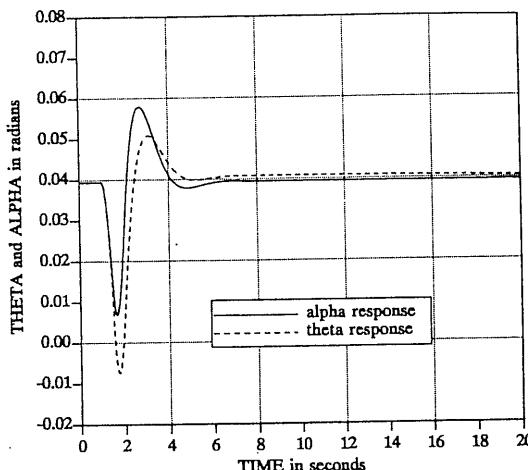


Fig. 3.5-4 F-16 model response to a $\pm 2^\circ$ elevator doublet.

The logical variable DFLAG is “true” only during initialization; the common block corresponds exactly to that in the aircraft model.

This input does not affect the lateral-directional states, and Figure 3.5-4 shows the longitudinal response (pitch rate is also affected but not shown). The pitch responses do not resemble (in shape or duration) the elevator disturbance that caused them. Instead, the responses are characteristic of the aircraft and represent a *natural mode* of the aircraft dynamics, which is coupled to the alpha and theta variables. This particular mode is known as the *short-period mode*. If we inspect the other longitudinal variables, we will find that airspeed remains essentially constant, and only alpha, theta, and pitch rate vary. The figure shows that theta and alpha vary in unison, so that there is only a small change in flight-path angle γ . The elevator disturbance was symmetric, but because of the nonlinearity of the dynamics, the aircraft does not settle back into the same steady-state condition. Thus the figure shows that a small positive flight-path angle remains after the disturbance has settled down. ■

Example 3.5-3: Simulated Response to a Throttle Pulse. In this example we shall use the same trim conditions as Example 3.5-2 and superimpose a doublet pulse on the steady-state throttle setting. The doublet will have the value 0.1 from 1 to 4 s and -0.1 from 4 to 7 s. The D subroutine is otherwise the same as in Example 3.5-2. Again, the lateral/directional dynamics are unaffected, and Figure 3.5-5 shows the longitudinal response. The angle of attack is barely affected, but the pitch attitude exhibits a very lightly damped oscillation with a long period. Therefore, the flight-path angle will change in phase with theta, and the airspeed will also be found to vary. This is the *phugoid mode*; it involves an interchange of potential and kinetic energy as

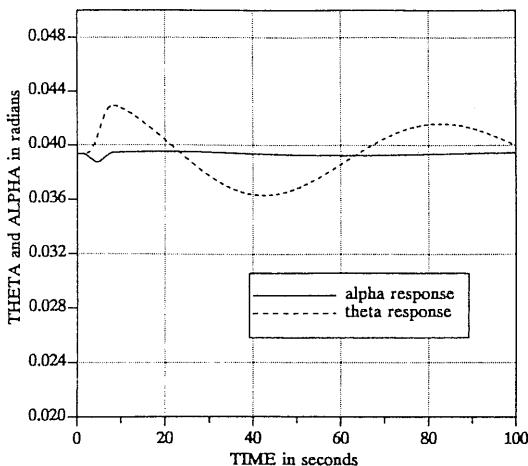


Fig. 3.5-5 F-16 model response to a ± 0.1 throttle doublet.

the aircraft climbs and slows down, and then dives and speeds up. In the case of the transport aircraft model, where the thrust vector does not pass through the cg, we can expect a throttle pulse to excite the short-period mode also. In the following sections we provide more insight into the modes, and tools to analyze them. ■

The foregoing examples have illustrated digital simulation using continuous-time dynamic equations, with control inputs applied in discrete time (i.e., changing only at the sampling instants) through the subroutine D. In Section 3.7 combined continuous-discrete simulation is illustrated, with synchronous discrete-time calculations performed in D to simulate onboard inertial navigation.

3.6 LINEARIZATION

The linear state equations were introduced in Chapter 1 as Equations (1.1-3), and in Section 2.5 the \dot{x} coefficient matrix, E , was introduced to accommodate implicit equations. Thus we made use of the state equations in the general linear form

$$\dot{E}x = Ax + Bu, \quad x \in R^n, \quad u \in R^m \quad (3.6-1a)$$

$$y = Cx + Du, \quad y \in R^p. \quad (3.6-1b)$$

The coupling matrix, D , representing a direct feed from input to output, will usually be null in our applications. In Chapter 2 the E matrix was derived algebraically and shown to be nonsingular for normal flight conditions. For a given flight condition the coefficient matrices are time invariant, and the resulting state equations are referred to as LTI (linear-time-invariant) state equations. Virtually all control system analysis and design techniques require LTI equations in some form, and we must now consider how we shall mechanize the derivation of LTI equations from our aircraft models.

In Section 2.5 the coefficient matrices were expressed in terms of stability derivatives at a specific steady-state flight condition. The algebraic linearization was limited to one type of steady-state flight condition: wings-level, non-sideslipping flight. This is a considerable inconvenience because other steady-state flight conditions are also important, and the dynamics may then be significantly different from the wings-level non-sideslipping case. Another disadvantage of the algebraic linearization is that although the calculation of coefficient matrices from the stability derivatives may be automated, one must be constantly aware of any approximations and assumptions about the various derivatives. Therefore, we shall develop an alternative method of deriving LTI state equations.

The stability derivatives can be derived from the same nonlinear wind-tunnel data that are used to build the nonlinear simulation model. This entails calculating various gradients. However, since we have cast the nonlinear simulation model in state-space form also, it is convenient to bypass the stability derivatives and calculate the Jacobian matrices for the LTI equations directly from the nonlinear model. This is done by perturbing the state and control variables from the steady-state condition, and numerically evaluating the partial derivatives in the Jacobian matrices. The Jacobian matrices may therefore be determined for any steady-state flight condition. Figure 3.1-1 illustrated how this idea fits into the software environment. Because the nonlinear equations in our state models are explicit, the linearization program need only be written to determine the equivalent A , B , C , and D matrices. We now describe a program to implement the technique, and provide some examples of the results.

Numerical Linearization

The basic method of obtaining approximations to the partial derivatives of discrete numerical data is to fit a polynomial curve through a number of adjacent data points. The derivative of this curve will be a known function that can then be evaluated at the midpoint of the range. The curve may be obtained from a least-squares fit to the data points, or from a spline with some degree of smoothing, such as that given by the IMSL routine ICCSU [IMSL, 1980]. This kind of processing is appropriate for the raw wind-tunnel data and for obtaining the stability derivatives. The controls design problem is slightly different. The designer will be supplied with a database that may have already been preprocessed (e.g., smoothed), and the data will appear continuous because an interpolation algorithm will be included in the database.

The 6-DOF rigid-body equations will be linked to the database, and the controls designer's problem is to generate the partial derivatives in the LTI Jacobian matrices. This must be done for arbitrary values of the independent variable, as demanded by the trim conditions. The problem is then to generate some adjacent data points that are close enough to give a good approximation to the slope at the trim value. If the points are too closely spaced, the finite precision of the computer arithmetic will cause errors in

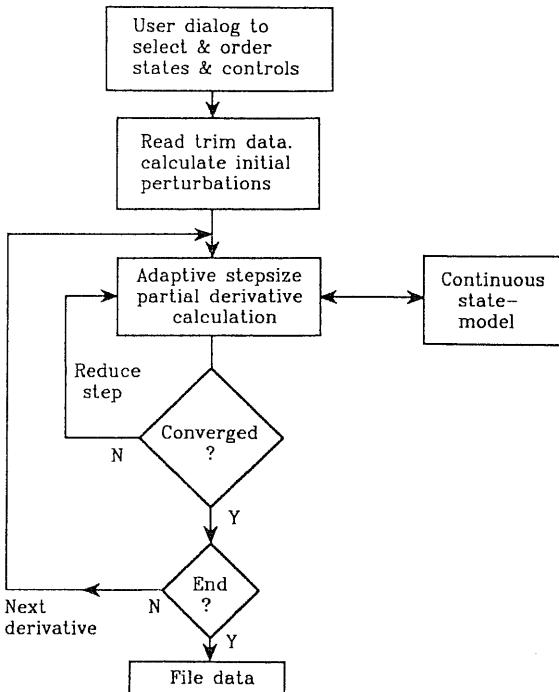


Fig. 3.6-1 Algorithm used for numerical linearization.

the calculation. A quadratic polynomial fitted through two data points equally spaced on either side of the equilibrium value is perfectly satisfactory for determining the gradient. In fact, the midpoint gradient calculated from a quadratic fit to three equally spaced data points is the same as that obtained from a straight line drawn between the two endpoints. Therefore, a quadratic-fit algorithm is very simple to implement, and this will be our choice.

A program, JACOB, was written to perform numerical linearization, and Figure 3.6-1 shows a simplified flowchart for the algorithm. Some source code is given in Appendix B. The algorithm uses an adaptive scheme in which perturbations of decreasing size are applied to the independent variable. As the perturbations are reduced, the calculated approximations to the partial derivative are compared, and the procedure is stopped when the change is within a certain tolerance of the calculated value.

A number of precautions against nonconvergence are built into the algorithm. The convergence tolerance is changed adaptively, and the operator is warned if the tolerance was not sufficiently small for a particular partial derivative. The final tolerance is 0.0001, and the output values are limited to a precision that is consistent with this tolerance. This avoids problems created by Jacobian elements that should be identical but differ in the last one or two digits (this can cause spurious transfer function zeros). The

algorithm will give a false answer if, in the nonlinear model, it encounters a limiter that sets a derivative to zero when the corresponding state variable reaches a saturation value. This problem is avoided by disabling rate limiters during linearization (alternatively, the algorithm can be designed to compare the results of both positive and negative perturbations about the equilibrium value). The nonlinear models of Section 3.3 will now be used to provide illustrative examples of numerical linearization.

Example 3.6-1: Linearization of the F-16 Model. In Chapter 2 it was shown that under the conditions of small perturbations from steady-state, wings-level, non-sideslipping flight, the rigid-aircraft equations of motion could be split into two uncoupled sets. These were the longitudinal equations that involve the variables speed, alpha, pitch attitude, and pitch rate, and the lateral-directional equations that involve beta, bank angle, and roll and yaw rates. The program JACOB makes it easy to demonstrate this decoupling and to show that coupling occurs when the sideslip and bank angles are nonzero. A good example is provided by two steady-state flight conditions that differ only in terms of bank angle. In Table 3.4-3, the wings-level pull-up with $\dot{\theta} = 0.3 \text{ rad/s}$ and the coordinated turn at 0.3 rad/s are both at 300 psf dynamic pressure with zero sideslip and similar angles of attack (and almost identical normal acceleration). The Jacobian matrices for these two steady-state conditions will now be compared.

When the Jacobian program is applied to the two steady-state flight cases cited, it will immediately be found that the geographic states north and east, and the orientational state ψ , have no coupling to the other states. Also, the altitude state only enters the aircraft equations through the atmosphere model and dynamic pressure, and in this case it has negligible coupling to the other states. These states are needed for navigation purposes, and their rates of change are of course coupled to the other states, but they can be omitted from consideration when we examine the basic aircraft dynamics. It is convenient to use the program to reorder the remaining nine states into longitudinal states v_T, α, θ, q , and the engine power state (POW), followed by lateral directional states β, ϕ, p, r . The inputs were ordered as $\delta_{th}, \delta_e, \delta_a, \delta_r$, and the outputs as a_n, q , and α . The Jacobian program converges for all of the partial derivatives in the A , B , C , and D matrices, and the results for the steady-state pull-up are shown in Figure 3.6-2.

These results are rounded to three significant digits, except for numbers less than 0.001, which are rounded to only one significant digit. The A and B matrices have been partitioned to separate the longitudinal and lateral states and controls, and it is evident that the expected decoupling does indeed exist. The decoupled relationship $\dot{\theta} = q$ is evident on the third row of the A matrix. There is a small amount of coupling in the A matrix between the q and r states, from alpha to p and r , and from ϕ to v_T .

The D matrix has a nonzero entry, corresponding to the elevator to normal acceleration transfer function, because accelerations are directly

$$\begin{array}{ccccccccc}
 v_t & \alpha & \theta & q & \text{pow} & b & \phi & p & r \\
 \hline
 A = & \left[\begin{array}{ccccccccc}
 -.127 & -235 & -32.2 & -9.51 & .314 & :-.0028 & .00126 & 5E-5 & 2E-4 \\
 -7E-4 & -.969 & 0 & .908 & -2E-4 & : 1.5E-5 & 0 & -4E-5 & -1E-5 \\
 0 & 0 & 0 & 1 & 0 & : 0 & 0 & 0 & 0 \\
 9E-4 & -4.56 & 0 & -1.58 & 0 & : 9.2E-5 & 0 & 0 & -.00287 \\
 0 & 0 & 0 & 0 & -5.00 & : 0 & 0 & 0 & 0 \\
 \hline
 1E-8 & 2E-5 & 3E-6 & 8E-7 & -3E-8 & : -.322 & .0612 & .298 & -.948 \\
 0 & 0 & 0 & 0 & 0 & : 0 & .0930 & 1.00 & .310 \\
 -3E-7 & -.00248 & 0 & 3E-4 & 0 & : -62.5 & 0 & -3.00 & 1.99 \\
 -3E-6 & -.00188 & 0 & .00254 & 0 & : 7.67 & 0 & -.262 & -.629
 \end{array} \right] \\
 \\
 \delta_{th} & \delta_e & \delta_a & \delta_r \\
 \hline
 B = & \left[\begin{array}{cccc}
 0 & -.244 & : 6E-6 & 2E-5 \\
 0 & -.00209 & : 0 & 0 \\
 0 & 0 & : 0 & 0 \\
 0 & -.199 & : 0 & 0 \\
 \hline
 1087 & 0 & : 0 & 0 \\
 0 & 2E-8 & : 3E-4 & 8E-4 \\
 0 & 0 & : 0 & 0 \\
 0 & 0 & : -.645 & .126 \\
 0 & 0 & : -.0180 & -.0657
 \end{array} \right] & D = & \left[\begin{array}{ccc}
 0 & .0333 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 0
 \end{array} \right]
 \end{array}$$

$$C = \left[\begin{array}{ccccccccc}
 .0208 & 15.2 & 0 & 1.45 & 0 & -4.5E-4 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1.00 & 0 & 0 & 0 & 0 & 0 \\
 0 & 57.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{array} \right] \begin{matrix} a_n \\ q \\ \alpha \end{matrix}$$

Fig. 3.6-2 Jacobian matrices for F-16 in a steady-state pull-up.

coupled to forces (and therefore to control surface deflection). Other expected features are the reciprocal of the engine time constant (at full power) as the only nonzero entry on the fifth row of the A matrix, and the value of g appearing in the 1,3 position. The reader should compare these results with the state equations derived in Chapter 2 and determine the significance of the numerical values.

If we now use the Jacobian program to determine the A matrix for the $4.5g$ coordinated turn, we obtain the matrix shown in Figure 3.6-3. A comparison of this matrix with the previous A matrix shows that several strong coupling terms have now appeared in the upper right block. Less pronounced coupling has appeared in the lower left block. At this stage we can only conclude that the effect of a large bank angle, with no change in dynamic pressure and little change in angle of attack, is to change the dynamic equations significantly. In the next section we develop the tools to

$$A = \left[\begin{array}{ccccccccc}
 -.090 & -169 & -31.2 & -7.75 & .318 & 31.4 & -7.73 & 5E-4 & 2E-3 \\
 -5E-4 & -1.05 & .0151 & .903 & -2E-4 & 3E-4 & -.0607 & -5E-4 & -1E-4 \\
 0 & 0 & 0 & .203 & 0 & 0 & -.300 & 0 & -.979 \\
 1E-3 & 1.26 & 0 & -1.66 & 0 & 1E-3 & 0 & .0589 & -.0157 \\
 0 & 0 & 0 & 0 & -5.00 & 0 & 0 & 0 & 0 \\
 -1E-4 & 1.4E-4 & -.0032 & 7E-6 & -3E-7 & -.322 & .0130 & .248 & -.961 \\
 0 & 0 & .300 & .0508 & 0 & 0 & 0 & 1.00 & .0105 \\
 -3E-4 & .0578 & 0 & -.0469 & 0 & -59.4 & 0 & -3.19 & 1.64 \\
 5E-5 & -.0617 & 0 & .0123 & 0 & 8.88 & 0 & -.299 & -.564
 \end{array} \right]$$

determine how the dynamic behavior is changed by this coupling, and in Chapter 4 we consider the implications of these changes in the dynamics. ■

Example 3.6-2: Comparison of Algebraic and Numerical Linearization. This example uses the transport-aircraft longitudinal model of Section 3.3. The model contains an alpha-dot contribution to the pitching moment, the thrust vector is offset from the cg by the amount Z_E , and the engine thrust varies with speed. It therefore provides a good check on the results of the algebraic linearization in Section 2.5.

A short program was written to evaluate the longitudinal state-equations coefficient matrices (2.5-31), using the formulae in Tables 2.5-1 to 2.5-3. The program contains the dimensionless stability derivatives given in the transport aircraft model and reads the steady-state trim data from a data file. It calculates the A and B matrices in (2.5-31) and then premultiplies them by the inverse of the E matrix. The new A and B matrices were printed out for comparison with numerical linearization results. It is instructive for the student to write a program like this (Problem 3.6-2). Close attention must be paid to the thrust derivatives, and this exercise leads to a better appreciation of the utility of numerical linearization.

The model was trimmed in the clean condition at a large angle of attack, and in climbing flight, so that $\sin \alpha$ and $\sin \gamma$ terms contributed significantly to the results. The trim condition was $cg = 0.25\bar{c}$, $h = 0$ ft, $V_T = 200$ ft/s, and $\gamma = 15^\circ$. This condition required an angle of attack of 13.9° and a throttle setting of 1.01 (i.e., slightly beyond maximum power!). The algebraic linearization program gave

$$E^{-1}A = \begin{bmatrix} v_T & \alpha & \theta & q \\ -2.7337E-02 & 1.6853E+01 & -3.1074E+01 & 0.0000E+00 \\ -1.4167E-03 & -5.1234E-01 & -4.1631E-02 & 1.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ -1.1415E-04 & -4.9581E-01 & 4.8119E-03 & -4.2381E-01 \end{bmatrix}$$

$$E^{-1}B = \begin{bmatrix} \delta_{th} & \delta_e \\ 1.0173E+01 & 0.0000E+00 \\ -1.2596E-02 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 \\ 2.7017E-02 & -7.0452E-03 \end{bmatrix}.$$

The algebraic linearization did not account for an altitude state and the consequent coupling of the equations through the atmosphere model. Therefore, only the first four states were selected when the numerical linearization was performed. The numerical linearization produced the following results:

$$A = \begin{bmatrix} -2.7337E-02 & 1.6852E+01 & -3.1073E+01 & 0.0000E+00 \\ -1.4168E-03 & -5.1232E-01 & -4.1630E-02 & 1.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ -1.1415E-04 & -4.9583E-01 & 4.8118E-03 & -4.2381E-01 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.0173E+01 & 0.0000E+00 \\ -1.2596E-02 & 0.0000E+00 \\ 0.0000E+00 & 0.0000E+00 \end{bmatrix}.$$

Note that the linearization program produces explicit equations directly ($E \equiv I$). These results are in very close agreement with the algebraic linearization results; the largest discrepancy is a difference of 2 in the fifth digit.

3.7 COMPUTATIONS FOR LINEAR TIME-INVARIANT STATE EQUATIONS

Time-Domain Solution of the State Equations

When the state equations are linear and time invariant, as in (3.6-1a), they can be solved analytically. Because of our powerful numerical integration tools, the solution will be less important to us as a practical technique than as a theoretical technique with important practical consequences. We consider first the time-domain solution of (3.6-1a), and later the Laplace transform solution. It will be assumed here that $E = I$, or that the equations have been solved for \dot{x} explicitly. The time-domain solution is easy to derive and can be found in standard control texts [Kailath, 1980]; it is

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^t e^{A(t-\tau)}Bu(\tau) d\tau. \quad (3.7-1)$$

The first component of this solution is the *homogeneous component*, which is the response to the initial conditions x_0 . The second component is the *forced component*, which is the response to the input $u(t)$. The matrix exponential e^{At} is known as the *continuous-time transition matrix* and is defined by the matrix series

$$e^{At} = I + At + \frac{A^2 t^2}{2!} + \frac{A^3 t^3}{3!} + \dots \quad (3.7-2)$$

The integral on the right-hand side of (3.7-1) is a convolution integral, the time-domain equivalent of transform multiplication in the frequency domain. The solution is of little computational value to us because of the difficulties of finding analytical expressions for the transition matrix for systems of all but the lowest order. Also, the convolution integral is inconvenient to evaluate for any but the simplest input functions.

Although the continuous-time solution of the LTI state equation is not very useful, it does lead to a discrete-time recursion formula that is sometimes useful. As in the case of the numerical integration of the nonlinear state equation, a solution becomes practical when we consider time intervals over which the input can be approximated by simple functions. Therefore, proceeding as in Section 3.5, we look for a discrete-time formula by consider-

ing a time interval from $t = kT$ to $t = (k + 1)T$. Equation (3.7-1) becomes

$$x(k + 1) = e^{AT}x(k) + e^{A(k+1)T} \int_{kT}^{(k+1)T} e^{-A\tau} Bu(\tau) d\tau. \quad (3.7-3)$$

The integral in (3.7-3) can be evaluated by a variety of methods: for example, the trapezoidal rule or Simpson's rule. We will take the simple zero-order hold (ZOH) approach [i.e., $u(\tau)$ held constant during the sampling interval, as mentioned in Section 3.5]; $u(kT)$ can then be taken out of the integrand. The remaining integral can be evaluated analytically, and the result is

$$x(k + 1) = e^{AT}x(k) + Q(T)Bu(k), \quad (3.7-4)$$

where

$$Q(T) = T \left[I - \frac{AT}{2!} + \frac{A^2 T^2}{3!} - \frac{A^3 T^3}{4!} + \dots \right].$$

Equation (3.7-4) is a discrete-time recursion formula. It can be used as an alternative to numerical integration of the state equations when the equations are linear or are such that the computation of e^{AT} is easy. The matrix exponential e^{AT} is the *discrete-time transition matrix*, also defined by (3.7-2). The transition matrix and the matrix $Q(T)$ can be computed simultaneously by means of a series approximation. A subroutine to compute this series approximation is given in Appendix B. Other methods of computing the transition matrix are available in the literature [Healey, 1973; Moler and Van Loan, 1978]; a superior technique is to use rational approximants [Zakian, 1970], but this is considerably more complex.

Example 3.7-1: Discrete-Time Quaternion Propagation. As an example of discretization of a continuous-time linear state equation and the use of discrete-time recursion, we will apply (3.7-4) to the quaternion differential equation (1.4-28). This equation must be solved by an onboard digital computer in strapdown inertial navigation systems, and the solution is needed in "real time."

The quaternion differential equation is time-varying because its coefficient matrix, Ω_q , contains the continuously changing body-rate components P , Q , and R . If the body rates are measured frequently enough by the digital computer, they can be assumed constant over each sample period. The discrete-time recursion formula for the quaternion differential equation, when Ω_q is a constant matrix, is

$$q(k + 1) = e^{-(1/2)\Omega_q T} q(k).$$

Powers of the Ω_q matrix have the properties given by (1.4-30), and in this

particular application the discrete-time transition matrix reduces to

$$e^{-(1/2)\Omega_q T} = I \cos\left(\|\omega_B\| \frac{T}{2}\right) - \Omega_q T \frac{\sin(\|\omega_B\| T/2)}{\|\omega_B\| T/2}.$$

This solution is satisfying analytically, but it is not very suitable for numerical computation. It is valid only when $\|\omega_B\|T/2$ is small (i.e., Ω_q constant over T), and since sine and cosine functions are normally computed from series approximations, it may be more efficient to use a series approximation to compute the transition matrix directly.

In practice, even with modest body rates, the quaternion must be updated at a very high rate to provide the attitude accuracy required for many applications. Currently used methods for quaternion propagation are based on a Taylor series approximation, and past values of the angular rates, to approximate the change in Ω_q over the sample interval. They are derived in a way similar to our derivations of the LMM integration formulae and can be traced back to the work of McKern [1968]; a routine is given in Appendix B. Results analogous to the transition matrix solution of the quaternion differential equation can be derived for the strapdown differential equation (1.4-12) (Problem 3.7-2). ■

Example 3.7-2: Discrete-Time Inertial Navigation Calculations. This example illustrates how combined continuous, discrete simulation may be performed. We will use the discrete-time quaternion propagation algorithm given in Appendix B, and an Adams–Bashforth integration routine, based on (3.5-14), to simulate inertial navigation onboard our aircraft model. Figure 3.7-1 illustrates the principle of strapdown inertial navigation; three rate gyros measure absolute angular rates around the three body axes, and three

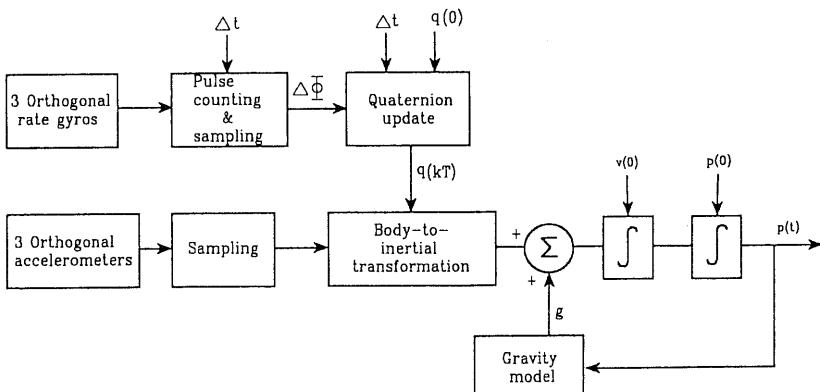


Fig. 3.7-1 Illustration of the principle of inertial navigation.

accelerometers measure components of acceleration along these axes. The measured body rates are integrated so that the angular orientation of the body (i.e., vehicle attitude) can be determined relative to its initial orientation (with respect to the reference frame). Using this attitude information, the accelerations along the body axes can be resolved into the reference frame and integrated twice to determine the vehicle position in the reference frame.

Figure 3.7-2 presents a discrete-time subroutine that can be called by the time-history simulation program at each sample instant to perform the inertial navigation calculations. Referring to the figure, on the first time through D the subroutine INIT is called to initialize the quaternion (S) according to (1.4-25). The initialization requires the Euler angles (X_4, X_5, X_6) from the aircraft state vector. The velocity (V) and acceleration (W) vectors are also initialized. On subsequent passes through D the routine QUAT (Appendix B) is used to update the quaternion. This requires the angular rates P, Q , and R and the sample period TS . The subroutine RESOLVE then constructs the body-to-inertial matrix (B^T) and resolves aircraft body-axes accelerations (outputs 7, 8, and 9) into the inertial reference frame (defined by the initialization). Subroutine ABM (Appendix B) is used twice to perform the integrations to determine position in the inertial frame. A simple simulation may use the NED frame as an inertial reference frame and

```

SUBROUTINE D2(TIME, TS, X, XD, DFLAG)
PARAMETER (MM=10, LL=20)
DIMENSION X(*), XD(*), S(4), A(3), V(3), W(3)
COMMON/CONTROLS/U(MM)
COMMON/OUTPUT/OP(LL)
LOGICAL DFLAG
C
IF (DFLAG) THEN
  CALL INIT(X(4), S)
  V(1)=OP(4)
  V(2)=OP(5)
  V(3)=OP(6)
  W(1)=0.0
  W(2)=0.0
  W(3)=0.0
ELSE
  CALL QUAT(P, Q, R, S, TS)
END IF
C
P= X(7)
Q= X(8)
R= X(9)
CALL RESOLVE (S, OP(7), A)
CALL ABM1(A, V, TS, 3, DFLAG)
CALL ABM2(V, W, TS, 3, DFLAG)
OP(1)= W(1)
OP(2)= W(2)
OP(3)= -W(3)
C
RETURN
END

```

Fig. 3.7-2 Subroutine for inertial navigation.

compare the results of computing position in the continuous-time model with position computed from subroutine *D*. This is left for the reader to perform.

Modal Decomposition

In Section 1.2 the modal coordinates were introduced to show the connection between eigenvalues and the characteristic modes of a dynamic system. In Examples 3.5-2 and 3.5-3 two different characteristic modes of the F-16 model dynamics were excited separately by applying different inputs. One of the variables (θ) observed in the simulation was found to be involved in both modes; the other observed variable (α) was involved in only one. It is possible to use the additional information contained in the eigenvectors to determine what physical variables are involved in a given mode and what inputs will excite the mode. The time-domain solution of the LTI state equation, (3.7-1), can be used for this purpose.

The continuous-time transition matrix can be expressed in terms of eigenvalues and eigenvectors in the following way. The similarity between the A matrix and (in general) a Jordan form matrix can be used to express an arbitrary power of A as

$$A^k = (MJM^{-1})(MJM^{-1}) \cdots = MJ^k M^{-1}.$$

When this is done for every term in the matrix exponential series, and J is assumed to be diagonal (distinct eigenvalues), the result is

$$e^{At} = Me^{Jt}M^{-1} = [v_1 \quad v_2 \quad \cdots \quad v_n] \begin{bmatrix} e^{\lambda_1 t} & & & \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ & & & e^{\lambda_n t} \end{bmatrix} \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_n^T \end{bmatrix}, \quad (3.7-5)$$

where v_i is the i th column of M (the i th eigenvector), and w_i^T is the i th row of M^{-1} . By definition, vectors w_i are orthonormal with the eigenvectors, that is,

$$w_i^T v_j = \begin{cases} 0, & i \neq j \\ 1, & i = j. \end{cases} \quad (3.7-6)$$

It is also easy to show that the vectors w_i are actually the *left eigenvectors* of A , that is, the right eigenvectors of A^T . If (3.7-5) is postmultiplied by the initial-condition vector x_0 , the homogeneous part of the solution of the

continuous-time state equation is obtained:

$$e^{At}x_0 = [v_1 \quad v_2 \quad \cdots \quad v_n] \begin{bmatrix} e^{\lambda_1 t}(w_1^T x_0) \\ e^{\lambda_2 t}(w_2^T x_0) \\ \vdots \\ \vdots \\ e^{\lambda_n t}(w_n^T x_0) \end{bmatrix}, \quad (3.7-7)$$

where the terms $(w_i^T x_0)$ are scalar products. Equation (3.7-7) can be rewritten as

$$e^{At}x_0 = \sum_{i=1}^n (w_i^T x_0) e^{\lambda_i t} v_i.$$

If this same procedure is followed, but with (3.7-5) postmultiplied by the vector $Bu(\tau)$, the forced component of the response is obtained. The complete response is therefore given by

$$x(t) = \sum_{i=1}^n (w_i^T x_0) v_i e^{\lambda_i t} + \sum_{i=1}^n v_i \int_0^t e^{\lambda_i(t-\tau)} (w_i^T Bu(\tau)) d\tau. \quad (3.7-8)$$

Equation (3.7-8) in effect uses the n linearly independent eigenvectors as a basis for the n -dimensional space, associates a characteristic mode with each basis vector, and shows the scalar component of $x(t)$ in each direction. If, for example, the initial condition vector lies in a direction such that a scalar product $(w_i^T x_0)$ is zero, the mode $e^{\lambda_i t}$ will not appear in the homogeneous response. Similarly, if the scalar product $[w_i^T Bu(\tau)]$ is zero, the mode $e^{\lambda_i t}$ will not contribute to the forced response. According to (3.7-6), this will occur if the initial-condition vector lies along any eigenvector other than the i th. Furthermore, if we examine the i th eigenvector, its nonzero elements will indicate which state variables participate in the i th mode. The relative involvement of the different variables will not be obvious because the elements of the state vector and the eigenvectors can, in general, all have different units.

Modal Decomposition Applied to Aircraft Dynamics

In Section 3.5 the classical phugoid and short-period aircraft modes were illustrated by nonlinear simulation, using the F-16 model with a stable cg position. The complete set of modes of a conventional aircraft will now be illustrated by modal decomposition using the linear F-16 model. The second set of trim conditions in Table 3.4-3 will be used, that is, stable straight and level flight. A Jacobian A matrix must first be found for this flight condition, and not all of the 13 states in the full A matrix will be needed.

Some states in the linear flat-Earth model do not couple back into the dynamics. The north and east displacement states and the yaw state ψ do not

affect the dynamics; the altitude state also has a negligible effect on the dynamics in the flight condition described above. These states correspond to the integrals of linear combinations of other states, and if retained in the A matrix, will produce zero eigenvalues (poles at the origin). The engine power state couples into the dynamics (through V_T) but is not influenced by any other states. Left in the A matrix, it will produce an eigenvalue of -1.0 , corresponding to the reciprocal of the 1-s engine time constant. All of these states can be omitted without affecting the basic aircraft modes. There is also clear decoupling of the lateral and longitudinal dynamics in this flight condition. Therefore, the modal decomposition will be demonstrated using two separate, reduced Jacobian matrices. Note that the method of deriving the A matrix by perturbing the state variables assumes that the control inputs are constant. Therefore, the modes derived in the analysis are "stick-fixed" modes; that is, the control surfaces are implicitly assumed to be locked in position. This assumption will hold most accurately for fully powered (as opposed to power-boosted or unpowered) control surfaces; these control systems are called *irreversible*.

Example 3.7-3: F-16 Longitudinal Modes. The IMSL eigenvalue subroutine EIGRF [IMSL, 1980], with double precision, was used to produce the following results. Other sources of eigenvalue/eigenvector routines are readily available (e.g., Press et al. [1986]). A simple driver program was written, and each pair of eigenvectors was normalized by dividing all elements by the complex number corresponding to the element of greatest complex magnitude. The longitudinal-dynamics Jacobian matrix for the F-16 model in straight and level flight at 502.0 ft/s with a cg position of $0.3\bar{c}$ is given by

$$A = \begin{bmatrix} v_T & \alpha & \theta & q \\ -2.0244E-02 & 7.8761E+00 & -3.2169E+01 & -6.5020E-01 \\ -2.5373E-04 & -1.0189E+00 & 0.0000E+00 & 9.0484E-01 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ 7.9472E-11 & -2.4982E+00 & 0.0000E+00 & -1.3861E+00 \end{bmatrix}.$$

The four states give rise to two complex-conjugate pairs of eigenvalues, which correspond to two stable oscillatory modes. The eigenvalues are

$$\begin{aligned} -1.2039 \pm j1.4922 & \quad (\text{short-period mode: } T = 4.21 \text{ s}, \zeta = 0.628) \\ -0.0087297 \pm j0.073966 & \quad (\text{phugoid mode: } T = 84.9 \text{ s}, \zeta = 0.117) \end{aligned}$$

The periods of these modes are separated by more than an order of magnitude, so they are easily identifiable as the short period and phugoid modes. The periods are also in reasonable agreement with the results of the nonlinear simulation in Examples 3.5-2 and 3.5-3. The phugoid mode is very lightly damped ($\zeta = 0.117$), but its period is so long that a pilot would have no difficulty in damping out a phugoid oscillation. The short-period mode is

reasonably well damped in this particular flight condition, and the aircraft response to elevator commands would be acceptable to the pilot.

The corresponding eigenvectors are given by

<i>Short period</i>	<i>Phugoid</i>	
1.0E + 00 $\pm j0.0E + 00$	1.0E + 00 $\pm j0.0E + 00$	v_T
9.0E - 02 $\pm j1.7E - 02$	-9.6E - 05 $\pm j5.0E - 07$	α
5.9E - 02 $\pm j5.4E - 02$	-3.8E - 04 $\pm j2.3E - 03$	θ
9.2E - 03 $\pm j1.5E - 01$	1.7E - 04 $\pm j8.4E - 06$	q

Both pairs of eigenvectors are dominated by the element corresponding to airspeed and the relative involvement of the other variables is difficult to assess. Nevertheless, the results show that the variables α and q are involved relatively weakly in the phugoid mode as compared to the short period. This agrees with the conclusions drawn from the nonlinear simulation examples.

The relative involvement of different variables in the dynamic modes can be determined more precisely if the dynamic equations are made dimensionless, so that the eigenvectors are also dimensionless. This was mentioned in Section 2.5 and requires the introduction of time scaling. Additional information can be extracted from the eigenvectors if they are plotted in the complex plane so that their phase relationship can be observed [Etkin, 1972]. ■

Example 3.7-4: F-16 Lateral-Directional Modes. The Jacobian matrix for the lateral/directional dynamics of the F-16 model, in straight and level flight at 502.0 ft/s, with a cg position of $0.3\bar{c}$, is given by

$$A = \begin{bmatrix} \beta & \phi & p & r \\ -3.2200E - 01 & 6.4032E - 02 & 3.8904E - 02 & -9.9156E - 01 \\ 0.0000E + 00 & 0.0000E + 00 & 1.0000E + 00 & 3.9385E - 02 \\ -3.0919E + 01 & 0.0000E + 00 & -3.6730E + 00 & 6.7425E - 01 \\ 9.4724E + 00 & 0.0000E + 00 & -2.6358E - 02 & -4.9849E - 01 \end{bmatrix}.$$

This time there are two real eigenvalues and a complex-conjugate pair; they are

$$\begin{aligned} -0.4399 &\pm j3.220 && \text{(dutch roll mode, } T = 1.95 \text{ s, } \zeta = 0.135) \\ -3.601 & && \text{(roll subsidence mode, } \tau = 0.28 \text{ s)} \\ -0.0128 & && \text{(spiral mode, } \tau = 77.9 \text{ s).} \end{aligned}$$

The eigenvectors are:

<i>Dutch Roll Mode</i>	<i>Roll Mode</i>	<i>Spiral Mode</i>	
-1.1E - 01 $\pm j9.7E - 02$	-2.0E - 03	3.2E - 03	β
-3.7E - 02 $\pm j3.0E - 01$	-2.8E - 01	1.0E + 00	ϕ
1.0E + 00 $\pm j0.0E + 00$	1.0E + 00	-1.5E - 02	p
-2.9E - 01 $\pm j3.3E - 01$	1.5E - 02	6.3E - 02	r

The oscillatory mode involves the roll and yaw rates and produces some bank and sideslip angle effects. This rolling, yawing motion, like the motion of a skater, is called the *dutch roll mode*. The aircraft rudder produces both rolling and yawing moments, and simulation and further analysis could be used to show that this mode can readily be excited by a rudder pulse. The eigenvalues show that the dutch roll period is quite short ($T = 1.95$ s) and the oscillation is very lightly damped ($\zeta = 0.135$). This would make landing in gusty wind conditions difficult for the pilot, and in a passenger aircraft, passengers sitting near the tail would be very uncomfortable in turbulent conditions.

The second mode is simply a stable exponential mode and clearly involves mostly roll rate; it is known as the *roll subsidence mode*. The aircraft bank angle response to lateral control inputs is an important part of the handling qualities requirements. This mode, derived from the linear model, will not allow the maximum roll rate to be calculated but does give a good idea of how quickly the aircraft will start to roll. In this case the time constant of 0.28 s indicates a fast roll response.

The third mode is also a stable exponential mode but is distinguished by a much longer time constant (78 s). It involves more bank angle and yaw rate than the roll mode and is known as the *spiral mode*. The spiral mode may be unstable in some aircraft, and stability can be built into a design by using wing dihedral (see Chapter 2). An unstable spiral mode can cause an aircraft to get into an ever-steepener, coordinated, spiral dive, and this has in the past claimed the lives of pilots not fully alert on long nighttime flights. ■

Laplace Transform Solutions and Transfer Functions

The Laplace transform solution technique for ODEs produces algebraic equations in terms of the transform variable s , which is also known as the complex frequency variable and has the dimensions of inverse time. In principle it is easy to solve the algebraic transform equations. Scalar equations can be solved using partial-fraction expansions and tables of inverse transforms, while the matrix state equations require, in addition, the inversion of a polynomial matrix ($sI - A$). However, these solutions are not well suited to machine computation, and hand computation involves a prohibitive amount of labor for other than low-order dynamic systems. Therefore, the Laplace transform solutions are mainly of interest as a theoretical tool, as we shall now see.

Since the matrix E will always be nonsingular in our applications, we can assume that the LTI state equation (3.6-1a) has been premultiplied by E^{-1} , and thereafter neglect E . If (3.6-1a) (with $E = I$) is Laplace transformed, the result is

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

Therefore,

$$X(s) = (sI - A)^{-1} [x(0) + BU(s)], \quad (3.7-9)$$

and the output is given by

$$Y(s) = C(sI - A)^{-1} [x(0) + BU(s)] + DU(s). \quad (3.7-10)$$

The matrix that relates the system outputs to the inputs, with zero initial conditions, is the *transfer function matrix*, $G(s)$. We shall also refer to the transfer function matrix as the MIMO (multi-input, multi-output) transfer function. Unlike the formal Laplace transform solution of the state equations, transfer functions are very important to us. Equation (3.7-10) shows that $G(s)$ is the $(p \times m)$ matrix given by

$$G(s) = C(sI - A)^{-1}B + D. \quad (3.7-11)$$

It is easy to show that a transfer function matrix is unchanged by a state-space transformation. Equations (1.2-21) represent a general state-space transformation, and if the coefficient matrices from those equations are substituted into (3.7-11), the result is

$$\begin{aligned} G(s) &= CM(sI - M^{-1}AM)^{-1}M^{-1}B + D \\ &= C[M(sI - M^{-1}AM)M^{-1}]^{-1}B + D, \end{aligned}$$

which reduces to (3.7-11) again when M and M^{-1} are taken inside the inner parentheses.

We shall now review some other important properties of transfer functions. A matrix inverse can be expressed in terms of the adjoint matrix with its elements divided by a determinant, so $G(s)$ can be written as

$$G(s) = \frac{C \operatorname{adj}(sI - A)B + D|sI - A|}{|sI - A|}. \quad (3.7-12)$$

The transfer function from the j th input to the i th output is the ij th element of $G(s)$, and this is the SISO (single-input, single-output) transfer function $g_{ij}(s)$. A SISO transfer function can therefore be written as

$$g_{ij}(s) = \frac{c_i \operatorname{adj}(sI - A)b_j + d_{ij}|sI - A|}{|sI - A|}, \quad (3.7-13)$$

where c_i and b_j are, respectively, the i th row of C and the j th column of B .

The elements of the adjoint $\operatorname{adj}(sI - A)$ are, by definition, cofactors of $(sI - A)$, and are therefore polynomials in s of degree $(n - 1)$ or lower.

Consequently, when $d_{ij} = 0$, the relative degree (denominator degree minus numerator degree) of this transfer function is unity or higher. Obviously, when $d_{ij} \neq 0$ the relative degree is zero (i.e., there is a direct-feed path through the transfer function). A relative degree of zero implies that a finite step-output component will appear immediately in response to a finite step-input. Also, in a simulation, a transfer function of zero relative degree allows the possibility of a closed-loop subsystem with a direct-feed path. It may then be necessary to solve algebraic equations within the simulation in order to establish the steady-state conditions in the loop.

Transfer Function Poles

The *poles* of a transfer function are the positions in the s -plane at which its magnitude becomes infinite. It is evident from (3.7-12) and (3.7-13) that all of the individual SISO transfer functions have the same poles, given by the roots of the n th-degree polynomial

$$|sI - A| = 0. \quad (3.7-14)$$

This is also the defining equation for the eigenvalues of the A matrix; therefore, the "system" poles are given by the eigenvalues of A . The natural modes of oscillation of a dynamic system are directly specified by the system poles, and therefore the ability to calculate the system poles is important to us. Fortunately, as noted earlier, excellent eigenvalue algorithms are readily available. When an input is applied to a system the relative amounts of excitation of the modes, and therefore the shape of the time response, depend on the transfer function numerator polynomials. Next we consider the transfer function numerators.

Transfer Function Zeros

The *zeros* of the individual SISO transfer functions are the positions in the s -plane where their magnitudes become zero, that is, the roots of the numerator polynomials. The number of zeros of each SISO transfer function will range from none to n , depending on the relative degree of the transfer function. We shall represent transfer functions in terms of their poles and zeros and make very little use of the numerator and denominator polynomials. It will become evident that this is because our design tools are, in many cases, based on poles and zeros (e.g., the poles and zeros are important in *root-locus design*). Furthermore, for numerical stability reasons we shall normally avoid using the numerator and denominator polynomials when calculating poles and zeros from the LTI state equations. Nevertheless, we now consider a method of evaluating a SISO numerator polynomial that will be useful in the theoretical development of algorithms for computing zeros.

Suppose that the $(n + 1)$ -square matrix

$$R(s) = \begin{bmatrix} sI - A & b_j \\ -c_i & d_{ij} \end{bmatrix} \quad (3.7-15)$$

is formed from the coefficients of the state equations corresponding to the SISO transfer function (3.7-13). This matrix is the *Rosenbrock system matrix* [Rosenbrock, 1970] for the state-space system. In the SISO case it has the property that its determinant is the transfer function numerator polynomial. This fact can be verified by using the matrix identity

$$\begin{bmatrix} T & U \\ -V & W \end{bmatrix} = \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ -V & I \end{bmatrix} \begin{bmatrix} I & T^{-1}U \\ 0 & VT^{-1}U + W \end{bmatrix}.$$

The determinant of a product of square matrices is equal to the product of the individual determinants. Therefore, taking determinants on both sides of the identity above yields

$$\begin{vmatrix} T & U \\ -V & W \end{vmatrix} = |T| |VT^{-1}U + W|.$$

When this result is applied to the system matrix (3.7-15) we obtain the formula

$$|R(s)| = |sI - A| (d_{ij} + c_i(sI - A)^{-1}b_j). \quad (3.7-16)$$

Therefore, the transfer function numerator can be found by evaluating the determinant of the system matrix. Determinant operations are unsatisfactory for numerical computation, but (3.7-16) is useful for low-order hand calculations and will shortly be seen to be a useful theoretical result.

Computation of Transfer Function Zeros

A well-known algorithm for computing transfer function numerator polynomial coefficients is Leverrier's algorithm [Fadeeva, 1959]. This algorithm is relatively simple to program and is commonly used. Unfortunately, it is not very stable numerically and the results may be inaccurate, even for polynomials of relatively low order (e.g., order 10 or lower). A better approach is to turn the computation of zeros into an eigenvalue problem, and a method of doing this is now described.

Various kinds of zeros can be defined for MIMO systems [MacFarlane and Karcanias, 1976; Laub and Moore, 1978], and a numerically stable algorithm to compute these zeros has been published [Emami-Naeini and Van Dooren, 1982]. This algorithm makes use of the excellent numerical properties of

modern eigenvalue routines. The SISO zeros corresponding to individual elements of the transfer function matrix will be sufficient for our needs, and a SISO version of the algorithm, with computation of transfer function gain, will be derived.

Consider the ij th SISO transfer function (3.7-13), and for brevity, omit the i and j subscripts. With the idea of turning the computation of zeros into an eigenvalue problem, we will form the inverse (reciprocal) of $g(s)$. The matrix inverse $(sI - A)^{-1}$ can be removed, and the connection to system poles broken, if this scalar equation is inverted using the following matrix identity:

$$(A + BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1}, \quad (3.7-17)$$

where A , B , C , and D are general matrices. This identity, known as the *matrix inversion lemma* [Kailath, 1980], is valid as long as the indicated inverses exist and the matrices are conformable. Assuming that d is nonzero, the inverse of (3.7-13) is

$$g^{-1}(s) = d^{-1} - d^{-1}c[(sI - A) + bd^{-1}c]^{-1}bd^{-1}. \quad (3.7-18)$$

Since $g(s)$ and $g^{-1}(s)$ satisfy $g^{-1}g = 1.0$, except at a finite number of discrete values of s corresponding to the poles of each function, the zeros of the transfer function $g(s)$ occur at the poles of the transfer function $g^{-1}(s)$. The poles of $g^{-1}(s)$ are the eigenvalues of the matrix $(A - bd^{-1}c)$, so an eigenvalue algorithm can be used to find the required zeros. The problem of what to do when d^{-1} does not exist must be addressed next.

The transfer function numerator polynomial is given by the determinant of the system matrix (3.7-16) irrespective of the existence of the direct feed d . This fact can be exploited in the derivation of an algorithm for finding zeros from eigenvalues. Assume that an $(n \times n)$ nonsingular transformation matrix, H , exists such that

$$c_i H = [0 \quad 0 \quad \cdots \quad 0 \quad \sigma], \quad (3.7-19)$$

where σ is nonzero (in general). If the matrix H is used to perform a state-space transformation on the original system, the system transfer function is unchanged, and the determinant of the system matrix (3.7-15) can be expanded very easily. A Laplace expansion about the last row, with $d_{ij} = 0$, yields

$$|R(s)| = (-1)^{2n+1} \sigma \begin{vmatrix} s - a'_{11} & a'_{12} & \cdots & a'_{1,n-1} & b'_1 \\ a'_{21} & s - a'_{22} & \cdots & a'_{2,n-1} & b'_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a'_{n1} & a'_{n2} & \cdots & a'_{n,n-1} & b'_n \end{vmatrix}, \quad (3.7-20)$$

where the primes indicate that the coefficients have been transformed by the

state-space transformation using H . The matrix of this determinant is still in Rosenbrock form, so that if b'_n is nonzero, the determinant can be evaluated by the eigenvalue technique. Otherwise, the process of reduction can be continued until a nonzero coefficient is obtained in the lower right corner.

A transformation matrix that will zero any contiguous block of vector elements, as in (3.7-19), is the Householder matrix [Golub and Van Loan, 1989]. A Householder matrix also has the orthogonality property, and this makes the state-space transformation very easy to perform.

The determinant on the right-hand side of (3.7-20) is a polynomial with the coefficient of the highest power of s equal to b'_n . When the polynomial factors formed from the transfer function zeros are multiplied together, a monic polynomial (coefficient of highest power equal to unity) is formed. The monic polynomial must be multiplied by a coefficient, k , equal to the product of b'_n and the factors $(-1)^{2n+1}\sigma$ that have accumulated during the reduction, in order to obtain the complete transfer function numerator polynomial. The transfer function denominator polynomial, formed according to (3.7-14) from the eigenvalues of the A matrix, is also monic. Therefore, the complete transfer function, obtained by eigenvalue techniques, is of the form

$$g(s) = \frac{k(s + z_1)(s + z_2) \cdots (s + z_m)}{(s + p_1)(s + p_2) \cdots (s + p_n)}. \quad (3.7-21)$$

When $g(s)$ is a feedback-control-loop transfer function in this factored form with all coefficients of s equal to unity, or expressed as the ratio of two monic polynomials, the coefficient k is known as the *static loop sensitivity*. Note that if there are no poles or zeros at the origin, the dc gain is finite and is determined by k and the zero and pole positions. If the relative degree is zero, k is the high-frequency gain of the transfer function. A program PZG, which incorporates the above algorithm, was used for the examples throughout this book.

Interpretation of the Transfer Function

A transfer function carries some very basic information about the way in which an aircraft (or any other system) will respond to an input. Two theorems that are fundamental in interpreting the transfer function are the Laplace transform initial and final value theorems:

$$\text{Initial Value: } \lim_{t \rightarrow 0} f(t) = \lim_{s \rightarrow \infty} sF(s) \quad (3.7-22)$$

$$\text{Final Value: } \lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s). \quad (3.7-23)$$

The initial value theorem shows that the sign of the initial response will be determined by the factor k in (3.7-21), and the type of initial response will be

determined by the transfer function relative degree and the input. The final value theorem shows that if the final value is nonzero, it will be determined by the pole and zero positions and k .

If the transfer function is non-minimum-phase (i.e., contains right-half-plane zeros), the initial response may have the opposite sign to the final response (depending on the number of NMP zeros). This is an undesirable type of response from the point of view of a human operator. NMP zeros are also undesirable in feedback controller design, since as we shall see later regarding root-locus plots, a right-half-plane zero tends to attract the closed-loop poles to the right-half s -plane. These types of zeros occur when there are two or more different paths to the system output, or two or more different physical mechanisms, producing competing output components.

Finally, we shall observe later that when a group of poles and zeros are close together in the s -plane, poles and zeros can often be canceled with each other in the transfer function. We now give two examples of aircraft transfer functions that illustrate these points.

Example 3.7-5: F-16 Elevator-to-Pitch-Rate Transfer Function. For this example a full 13-state Jacobian A matrix was obtained for the straight and level flight conditions used in Examples 3.7-3 and 3.7-4 (i.e., the second set of trim conditions from Table 3.4-3). The B and C Jacobian matrices were also obtained. This set of Jacobians was used as input to the PZG program; the row of C corresponding to pitch rate, q , and the column of B corresponding to elevator deflection, δ_e , were selected. Table 3.7-1 shows the "gain" and poles and zeros resulting from double-precision computation, rounded to

TABLE 3.7-1. F-16 Model, Elevator-to-Pitch-Rate Transfer Function^a

Zeros		Poles		
Real Part	Imaginary Part	Real Part	Imaginary Part	
0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	N
0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	E
0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	ψ
-4.3987E - 01	3.2200E + 00	-4.3987E - 01	3.2200E + 00	dutch
-4.3987E - 01	-3.2200E + 00	-4.3987E - 01	-3.2200E + 00	dutch
-3.6009E + 00	0.0000E + 00	-3.6009E + 00	0.0000E + 00	roll
-1.2835E - 02	0.0000E + 00	-1.2835E - 02	0.0000E + 00	spiral
-8.8010E - 04	0.0000E + 00	-2.0874E - 03	0.0000E + 00	altitude
-1.0000E + 00	0.0000E + 00	-1.0000E + 00	0.0000E + 00	engine
0.0000E + 00	0.0000E + 00	-1.2040E + 00	1.4923E + 00	short period
-2.1785E - 02	0.0000E + 00	-1.2040E + 00	-1.4923E + 00	short period
-9.8713E - 01	0.0000E + 00	-7.6538E - 03	7.8119E - 02	phugoid
		-7.6538E - 03	-7.8119E - 02	phugoid

^aStatic loop sensitivity = -10.453 (degree units).

seven digits. The poles and zeros have been ordered to suit the purposes of this example.

The poles have been identified in the rightmost column of the table; it is evident that poles that are not relevant to the elevator to pitch-rate dynamics have been automatically canceled by zeros. Cancellation of at least the first six digits has occurred, so the final transfer function can be expected to be a good approximation to the q/δ_e dynamics. This also justifies the decoupling of longitudinal- and lateral-directional dynamics as used in Examples 3.7-3 and 3.7-4.

The first three poles are at the origin and represent the integration of velocity components that lead to the north, east, and directional (ψ) states. These are canceled by zeros at the origin. The dutch roll, roll mode, and spiral mode poles are also canceled by zeros. The altitude pole is not exactly at the origin, as explained earlier (in modal decomposition), and is not canceled exactly because it is coupled to the longitudinal dynamics. The engine pole is canceled exactly because the engine-lag model is driven only by the throttle input. The remaining four poles (phugoid and short-period modes) and three zeros yield the transfer function

$$\frac{q}{\delta_e} = \frac{-10.45s(s + 0.9871)(s + 0.02179)}{(s + 1.204 \pm j1.492)(s + 0.007654 \pm j0.07812)} \frac{\text{deg/s}}{\text{deg}}.$$

The phugoid mode has a natural frequency of 0.079 rad/s and a damping ratio of 0.10; the corresponding figures for the short-period mode are 1.9 rad/s and 0.63.

The elevator to pitch-rate transfer function has a dc gain of zero (because of the zero at the origin), indicating that a constant elevator deflection will not sustain a steady pitch rate. If the phugoid poles are canceled with the zero at the origin and the zero at $s = -0.02$, a *short-period approximation* transfer function is obtained:

$$\frac{q}{\delta_e} = \frac{-10.45(s + 0.9871)}{s + 1.204 \pm j1.492} \frac{\text{deg/s}}{\text{deg}}.$$

This transfer function has a finite dc gain and shows that constant elevator deflection tends to produce constant pitch rate over an interval of time that is short compared to the phugoid period. The short-period approximation will be used in controller designs in Chapter 4, and its validity will be demonstrated. In the next subsection the short-period approximation will be examined in the frequency domain. ■

Example 3.7-6: Transport Aircraft Throttle Response. In this example we examine the throttle-to-speed transfer function for the transport aircraft

model in Section 3.3. The model was trimmed for level flight at sea level, in the clean configuration with $x_{cg} = 0.25\bar{c}$ and a true airspeed of 250 ft/s, and the following Jacobian matrices were derived:

$$A = \begin{bmatrix} v_T & \alpha & \theta & q & h \\ \begin{bmatrix} -1.6096E-02 & 1.8832E+01 & -3.2170E+01 & 0.0000E+00 & 5.4000E-05 \\ -1.0189E-03 & -6.3537E-01 & 0.0000E+00 & 1.0000E+00 & 3.7000E-06 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 & 0.0000E+00 \\ 1.0744E-04 & -7.7544E-01 & 0.0000E+00 & -5.2977E-01 & -4.1000E-07 \\ 0.0000E+00 & -2.5000E+02 & 2.5000E+02 & 0.0000E+00 & 0.0000E+00 \end{bmatrix} \end{bmatrix}$$

$$B = \begin{bmatrix} \delta_{th} \\ \begin{bmatrix} 9.9679E+00 \\ -6.5130E-03 \\ 0.0000E+00 \\ 2.5575E-02 \\ 0.0000E+00 \end{bmatrix} \end{bmatrix} \quad C = [1 \ 0 \ 0 \ 0 \ 0] (v_T)$$

The altitude state (h) has very small coupling to the other states and was initially neglected. The throttle-to-speed transfer function (with the elevator fixed) as determined from the states v_T, α, θ, q was found to be

$$\frac{v_T}{\delta_{th}} = \frac{9.968(s - 0.0601)(s + 0.6065 \pm j0.8811)}{(s + 2.277E-4 \pm j0.1567)(s + 0.5904 \pm j0.8811)}. \quad (1)$$

As expected, this transfer function essentially involves only the phugoid mode, and when the short-period poles are canceled with the complex zeros, we are left with

$$\frac{v_T}{\delta_{th}} = \frac{9.968(s - 0.0601)}{s + 2.277E-4 \pm j0.1567}. \quad (2)$$

The poles and zeros of (2) are quite close to the origin and the relative degree is unity, so throttle inputs are initially integrated. However, the phugoid mode will soon take over and hide this effect under a very lightly damped oscillation in speed. In addition, the non-minimum-phase (NMP) zero indicates that there are competing physical mechanisms at work. It may be remembered that the engine thrust line is offset below the cg, and this will cause the aircraft to tend to pitch up and consequently slow down, in response to a sudden increase in throttle. Furthermore, at this relatively low speed the aircraft is trimmed with a large amount of "up elevator," so that any initial increase in speed tends to create an increase in the nose-up pitching moment and again counteract the increase in speed. These facts can be confirmed by changing the engine offset and by trimming the model at higher speeds where less elevator is required. The NMP zero can thus be made to move to the origin and into the left-half plane.

In general, when the throttle is opened, the extra power input may produce an increase in speed and/or a gain in altitude, and the phugoid mode is associated with the subsequent interchange of potential and kinetic energy. In this case we see that the positive static loop sensitivity, and single NMP zero, correspond to a negative dc gain. Therefore, when the throttle is opened a very lightly damped phugoid oscillation will be initiated, starting with an increase in speed but with a mean value corresponding to a lower speed. The increased thrust will therefore be converted to an increase in altitude. This can be confirmed with a time-history simulation, by applying a step throttle input to the linear model from which transfer function (1) was obtained.

Now we consider a more accurate transfer function model of the aircraft. If the aircraft altitude state is included in the A matrix, it is found that because of the atmosphere model, there are small coupling terms from altitude to several other states. The transfer function corresponding to (1) then becomes

$$\frac{v_T}{\delta_{th}} = \frac{9.968(s - 0.01506)(s - 0.04528)(s + 0.6066 \pm j0.8814)}{(s + 3.305E-5)(s + 0.5905 \pm j0.8813)(s + 6.788E-5 \pm j0.1588)}. \quad (3)$$

The very slow altitude pole (at $s = -3 \times 10^{-5}$) has now appeared in the transfer function. An additional NMP zero is also present, and the dc gain of the transfer function is now positive. The physical explanation is that since the decrease in atmospheric density with altitude is now modeled, the tendency to gain altitude is reduced and the speed will now increase in response to a throttle increase.

Simulation results (see Problem 3.7-6) show that for the linear model without the altitude state, the average airspeed (averaged over the phugoid period) decreases in response to a throttle step. When the altitude state is included, the average airspeed decreases at first, and then increases. The altitude increases in either case. The response of the nonlinear model with a relatively small throttle step increment (10% increase) agrees closely with the linear model (with altitude included). ■

Frequency Response. Plots of the magnitude and phase angle of a SISO transfer function, as the transform variable takes on values $s = j\omega$ in the range $\omega = (0, \infty)$, are referred to as frequency-response plots. They play an important role in classical design of control systems. If we attempt to calculate frequency response from the transfer function matrix elements, then an $(n \times n)$ complex matrix inversion, $(j\omega I - A)^{-1}$, must be performed for every value of ω . Also, only a few of the elements of this matrix will

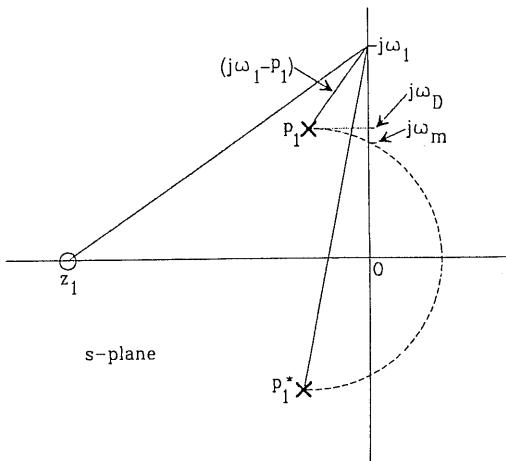


Fig. 3.7-3 Frequency response maximum from a pole-zero plot.

typically be involved in a specific frequency response. Therefore, we shall opt for a simpler computation in which $s = j\omega$ is substituted in the factored transfer function (3.7-21), and the complex product of all the numerator factors is divided by the complex product of all the denominator factors. This approach also has a useful geometrical interpretation.

Consider the pole-zero plot in Fig. 3.7-3, in which vectors have been drawn from the poles and zeros to the point $s = j\omega_1$ on the s -plane $j\omega$ axis. A vector drawn from $s = p_i$ to $s = j\omega_1$ represents the complex quantity $(j\omega_1 - p_i)$ in magnitude and phase, and this is a typical factor of (3.7-21). Therefore, the magnitude of the frequency response is given by the product of the lengths of all the vectors drawn from the zeros to $s = j\omega_1$, divided by the product of the lengths of all the vectors drawn from the poles. The phase is given by the sum of all the angles of the vectors drawn from the zeros, minus the sum of all the angles of the vectors drawn from the poles. As an illustration of a geometrical property, assume that the two complex poles in Fig. 3.7-3 dominate the frequency response. The two vectors drawn from these poles form a triangle whose base and altitude are, respectively, $(p_1 - p_1^*)$ and $\text{Re}[p_1]$. The triangle area is thus fixed while ω_1 varies. Since the area is also given by the product of the lengths of the two vectors and the sine of the angle between them, the product of the lengths must have a minimum when the included angle is 90° . Therefore, these two poles create a peak in the frequency response at the frequency ω_m , located by drawing a semicircle with the line joining the two poles as diameter (as shown in the figure). It follows that the frequency ω_m is always lower than the damped frequency ω_D . Pole-zero plots will provide a great deal of insight as we

proceed with control system design and, in conjunction with the root-locus technique, are an invaluable tool.

Example 3.7-7: F-16 Elevator-to-Pitch-Rate Frequency Response. The poles and zeros of the elevator-to-pitch-rate transfer function, given in Example 3.7-5, will now be used to generate the corresponding frequency-response plots. Figure 3.7-4a shows a Bode plot (decibel magnitude versus frequency on a logarithmic scale) of the magnitude response of both the complete transfer function and the short-period approximation. The phase plots are shown in Fig. 3.7-4b. The magnitude plot shows a large peak in the response at a frequency close to that of the lightly damped phugoid mode, and a smaller peak due to the more heavily damped short-period mode.

Both the magnitude and phase plots show that the short-period approximation is a good approximation to the pitch-rate transfer function at frequencies above about 0.05 Hz. The upper cutoff or corner frequency of the short-period transfer function is about 0.9 Hz, and this gives some feel for the speed of response in pitch when different aircraft are compared. Note that the exact phase plot starts at $+90^\circ$ due to the zero at the origin, rises toward 180° because of the additional phase lead of the zero at $s = -0.02$, and then falls back rapidly because of the 180° lag effect of the phugoid

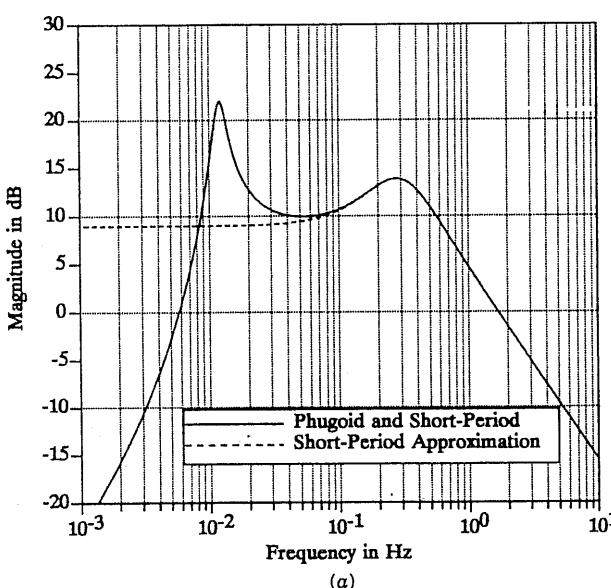


Fig. 3.7-4 (a) Magnitude of elevator to pitch-rate frequency response.

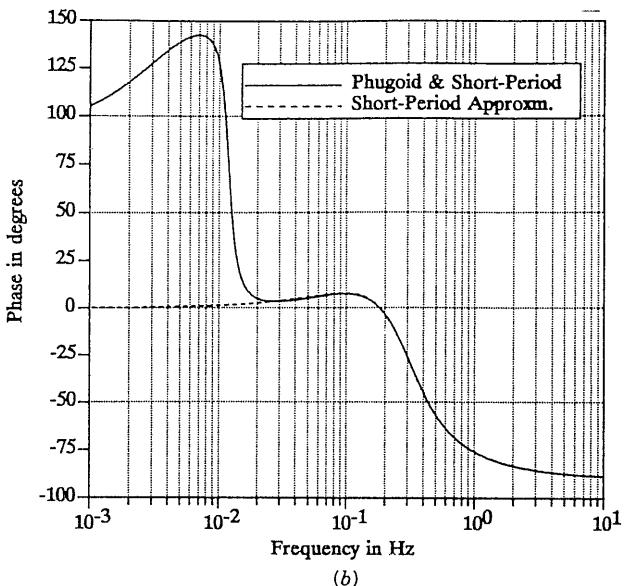


Fig. 3.7-4 (b) Phase of elevator to pitch-rate frequency response.

poles. The zero at $s = -0.99$ causes another small lead effect before the lag of the short-period poles takes over; the high-frequency asymptotic phase shift is -90° because the relative degree of the transfer function is unity. ■

Frequency-domain concepts are very convenient for many of our purposes. For example, a computer model of the mechanical structure of the fuselage and wings of an aircraft will provide the frequencies of the "structural modes" of the aircraft. Typically, the aircraft control systems are designed on the basis of the rigid-body aircraft model and the structural modes represent "unmodeled high-frequency dynamics." The control-systems designer may allow for them [AFWAL-TR-84-3105, 1984] in an ad hoc way by placing sensors to minimize their effects, using "equalization" filters to cancel the "mechanical resonances," and limiting the bandwidth of the control systems. This problem of making the control system design "robust" to unmodeled dynamics is described in more detail in Chapter 6.

Frequency-domain techniques are also used to design compensation networks to stabilize or improve the transient response of feedback controllers. The frequency-domain concepts of gain margin and phase margin (see next section) may be used to assess the robustness of a feedback control system to parameter variations in the controlled dynamics. In the case of multiloop feedback systems, the gain and phase margins of individual loops may not provide a reliable indication of robustness, and in Chapter 6 we introduce

frequency-domain plots of matrix singular values as a means of assessing robustness. An example of frequency-domain design will be given in the next section after we have introduced the necessary feedback theory.

3.8 FEEDBACK CONTROL

The major portion of this book is concerned with performing feedback control design on aircraft dynamics, and in this section we review the design techniques and develop the framework for the software tools that will be needed. An understanding of basic classical control theory is assumed.

If properly designed, *negative feedback* can stabilize unstable dynamic modes, improve the damping of lightly damped modes, increase speed of response, make the dynamic behavior more independent of internal parameter variations in the *plant* (i.e., the controlled dynamics), and make a controlled variable accurately track a command input with a high degree of immunity to external disturbances. All of these features are relevant in aircraft applications.

The classical representation of a SISO feedback control system is shown in Fig. 3.8-1, in terms of both real and Laplace-transformed variables. Negative feedback will always be assumed (as indicated by the negative sign on the summing junction), and the *error signal* $e(t)$ is equal to the difference between the *reference input* (or command) $r(t)$ and the *controlled variable* $y(t)$. The feedback transfer function $H(s)$ may often represent only the effect of a unity-gain wide-bandwidth measurement transducer; then, if the system is designed to have a small error, the output $y(t)$ will closely match the input $r(t)$.

In Fig. 3.8-1, $N(s)$ and $D(s)$ signify, respectively, numerator and denominator polynomials of the transfer functions. The *closed-loop* transfer function has the well-known form

$$\frac{Y(s)}{R(s)} = \frac{G}{1 + GH} = \frac{k_1 N_1 D_2}{D_1 D_2 + k_1 k_2 N_1 N_2}. \quad (3.8-1)$$

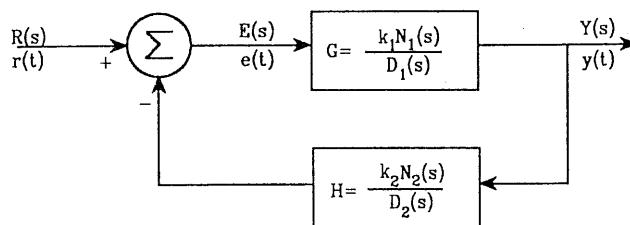


Fig. 3.8-1 Block diagram of a SISO control system.

The numerator and denominator polynomials are assumed to be monic, so that $K = k_1 k_2$ is the static loop sensitivity of the *loop transfer function* $G(s)H(s)$. We shall often simply refer to K as the loop gain. If K is increased, we expect in general to obtain improvements in closed-loop steady-state behavior (e.g., closer tracking, better disturbance rejection), but possibly accompanied by degraded transient performance (e.g., more oscillatory response) or even instability.

The stability of the control system, and the properties of the closed-loop dynamic modes (natural frequency, damping, time constant), can be determined from the zeros of the *characteristic equation*

$$1 + G(s)H(s) = 0 \quad (3.8-2a)$$

that is, the roots of the *characteristic polynomial*

$$D_1 D_2 + K N_1 N_2. \quad (3.8-2b)$$

The positions of the characteristic roots in the complex plane can be ascertained by using a polynomial root-solving program on the characteristic polynomial, or from the *A*-matrix eigenvalues. Graphically oriented techniques that provide information about the location of the roots are the *root-locus method* and, more indirectly, the *Nyquist diagram* and *Bode plots*. We review these techniques briefly in the following subsections, but first some general design considerations will be reviewed.

System Type

Analysis of Fig. 3.8-1 shows that the transfer function that determines the error, for any given input, is

$$\frac{E(s)}{R(s)} = \frac{1}{1 + GH} = \frac{D_1 D_2}{D_1 D_2 + k_1 k_2 N_1 N_2}. \quad (3.8-3)$$

Assume that the forward-path transfer function, $G(s)$, contains q pure integrations, and that the feedback path, $H(s)$, contains neither pure integrations nor zeros at the origin. $H(s)$ will normally correspond to *unity feedback* (i.e., $[H(s) \equiv 1]$) in our designs, so that the output will attempt to track the input. Let the system have a polynomial input $t^n U_{-1}(t)$ [where $U_{-1}(t)$ is the unit step function]. Then the final value theorem gives the following expression for the steady-state control error:

$$e_{ss}(t) = \lim_{s \rightarrow 0} s E(s) = \lim_{s \rightarrow 0} \frac{s^q D'_1(s) D_2(s)}{s^{n+1} (D_1 D_2 + k_1 k_2 N_1 N_2)}. \quad (3.8-4)$$

The term D'_1 indicates that s^q has been factored out of the D_1 polynomial, and there can be no factors of s in the numerator or denominator other than those shown. Therefore, (3.8-4) shows that in order to track a polynomial input of degree n , with finite (or zero) steady-state error, the control system must be a Type n (or higher) control system. That is, it must have ($q = n$) pure integrations in the forward path. An additional integration will reduce the steady-state error to zero, while one less integration will cause the steady-state error to grow without bound. Each integration adds 90° of phase lag to the loop transfer function and makes the control system progressively more difficult to stabilize, so that systems are restricted to Type 2 or lower in practice. System type requirements are a preliminary design consideration, and this idea is used in the designs in Chapter 4.

Frequency-Domain Design

A *Nyquist plot* is a polar plot of the loop transfer function $G(j\omega)H(j\omega)$ over the frequency range from zero to infinity. In control theory texts it is shown that for a stable closed-loop system, the Nyquist plot of GH should encircle the point $(-1, j0)$ as many times as there are poles of $G(s)H(s)$ in the right half of the s -plane (the encirclements being in the clockwise direction).

In nearly all practical cases the loop gain GH is itself a stable function and then, if the locus of GH is plotted as the frequency is varied from $\omega = \infty$ to $\omega = 0$, the closed-loop system is unstable if the critical point lies to the left of the locus. This is illustrated in Fig. 3.8-2.

We have no real need of the Nyquist stability criterion because eigenvalue techniques are much more convenient to use in our design framework (Fig. 3.1-1). However, it is useful to know that if the frequency-response locus passes close to the point $(-1, j0)$, the stability boundary is being approached and the system time response is likely to be underdamped. This idea leads to the concepts of *gain margin* and *phase margin*.

The gain margin of a feedback loop is the increase in gain (in decibels) that can be allowed before the loop becomes unstable. It can be calculated by finding the gain at the phase-crossover frequency, shown in Fig. 3.8-2. The phase margin is the phase angle of GH , when its magnitude is unity, plus 180° . It can be calculated from the gain-crossover shown in Fig. 3.8-2. As a rule of thumb a phase margin of 30° to 60° will be required to obtain a good closed-loop transient response, and this should be accompanied by a gain margin of 6 to 15 dB.

In classical frequency-domain design, compensation networks (e.g., phase-lead, phase-lag) are cascaded with the loop transfer functions to allow the phase margin to be increased or to allow the loop gain to be increased for a given phase margin. The fourth and fifth networks in Table 3.2-1 are, respectively, lead and lag compensators and their properties are now briefly reviewed.

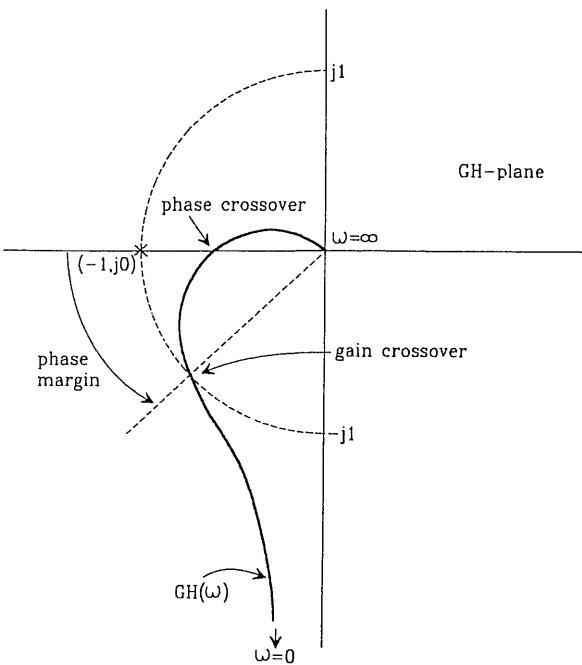


Fig. 3.8-2 Stability margins on the Nyquist plot.

The lead and lag transfer functions can both be written (apart from a gain constant) as

$$G_c(s) = \frac{s + z}{s + p} \quad \begin{aligned} p > z &\equiv \text{lead} \\ p < z &\equiv \text{lag}. \end{aligned} \quad (3.8-5)$$

The frequency-response properties can be found by substituting $s = j\omega$, calculating the phase and magnitude expressions, and differentiating to determine the conditions for maximum phase lead or lag. If ω_{ϕ_M} denotes the frequency of maximum lead or lag, ϕ_M the value of the maximum lead or lag, and G_{ϕ_M} the gain at ω_{ϕ_M} , then

$$\begin{aligned} \omega_{\phi_M} &= \sqrt{pz} \\ \sin \phi_M &= \frac{p - z}{p + z} \\ G_{\phi_M} &= \sqrt{z/p}. \end{aligned} \quad (3.8-6)$$

Note that the high-frequency (hf) gain of the compensator (3.8-5) is 1.0, the low-frequency (lf) gain is z/p , and the gain at ω_{ϕ_M} is the geometric mean

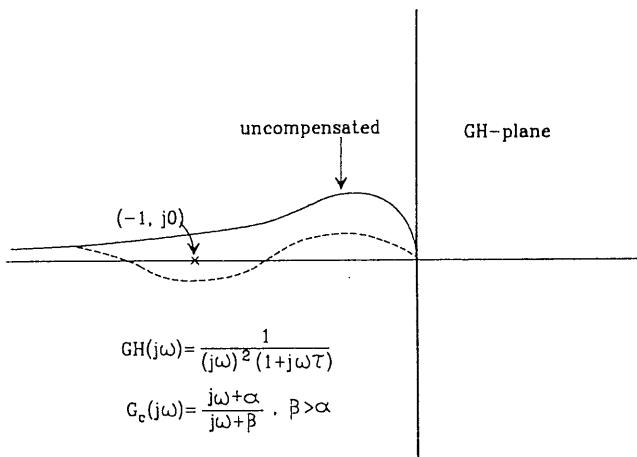


Fig. 3.8-3 Lead compensation on the Nyquist plot.

of these two. In the case of a lead compensator ($p > z$), the hf gain is greater than the lf gain, and vice versa for a lag compensator. The rising gain characteristic of a lead compensator approximates a differentiator and tends to accentuate system noise. Therefore, the phase-lead compensator is usually limited to $p/z \leq 10$. Figure 3.8-3 illustrates how a lead compensator could be used to give a Type 2 control system an adequate phase margin. Numerical design examples of lead and lag compensation are given in Chapter 4.

SISO Root-Locus Design

Equation (3.8-1) shows that the zeros of the closed-loop transfer function are determined by the zeros of $G(s)$ and the poles of $H(s)$ and are independent of K . The closed-loop poles depend on K through the roots of the characteristic equation. The root-locus technique applies to a polynomial in the form (3.8-2b) and allows the loci of the roots to be constructed as K varies. Note that (3.8-2b) is in the form (open-loop poles) + $K \times$ (open-loop zeros), and this causes the root loci to start on the open-loop poles and terminate on the open-loop zeros.

It also follows quite simply from (3.8-2b) that when K is positive, a root-locus branch lies on any section of the s -plane real axis that is to the left of an odd number of poles and zeros. When K is negative we have the so-called *zero-angle root locus*, which is on the axis to the left of an even number of poles and zeros. The factor K will comprise the transfer function static loop sensitivity and any added gain factors. Therefore, when applying a root-locus program, gain factors with the opposite sign to the static loop sensitivity will yield the zero-angle root locus, and vice versa.

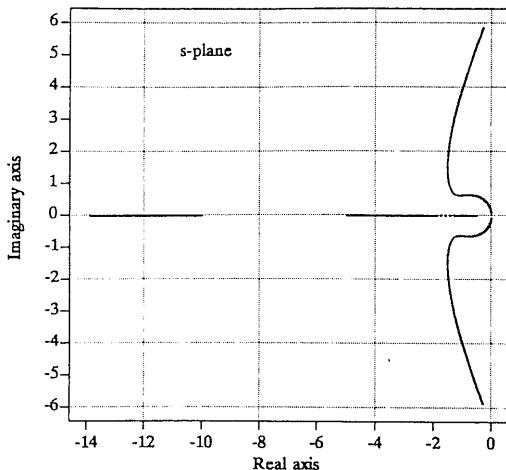


Fig. 3.8-4 Lead compensation on the root locus plot.

We shall assume familiarity with the basic rules for constructing root loci. Because of the digital computer a root-locus plot can be constructed without knowledge of these rules, but this knowledge is required when root-locus concepts are used to design a compensator. Figure 3.8-4 is a root-locus plot that illustrates how a lead compensator stabilizes the Type 2 system used as an example in the preceding subsection. Note that zeros tend to attract the loci, and the compensator zero has pulled the two previously unstable branches into the left-half s -plane.

The root-locus technique works well with low-order dynamics. With a large number of poles and zeros it becomes necessary to switch to the frequency-domain techniques illustrated in the preceding subsection. We shall use root-locus plots in Chapter 4, and in the following subsections a state-equations framework will be developed that can be used to produce root-locus data.

Multiloop and MIMO Design

Figure 3.8-5a shows the structure of a typical SISO, multiloop, classical control system. Inner-loop feedback is typically derivative feedback from a tachometer or rate gyro, and the transfer function H_1 may be a high-pass filter so that the rate signal is not present at the plant input during the steady state. The classical design technique for this multiloop structure is *successive loop closure*, which is a trial-and-error iterative process.

An understanding of the effect of the inner-loop feedback will reduce the amount of trial-and-error design. Derivative feedback in the inner loop has

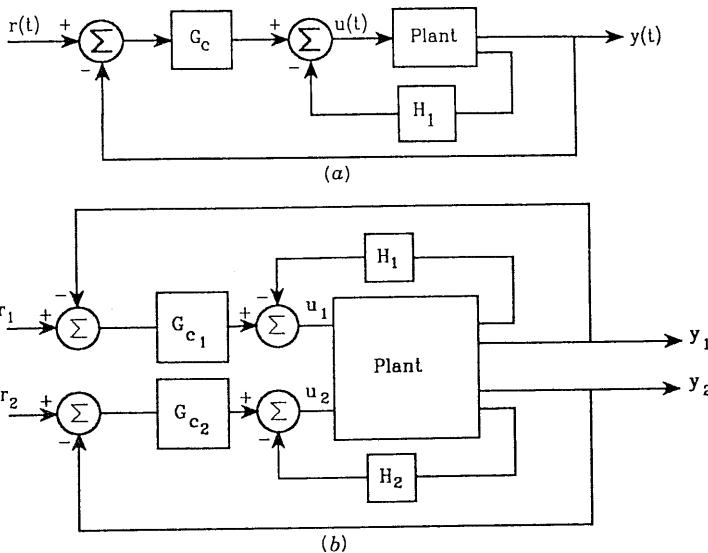


Fig. 3.8-5 Multi-loop control system (a) SISO; (b) MIMO.

the effect of increasing the damping of complex plant poles; this allows higher outer-loop gain and tends to increase the closed-loop bandwidth. Only the poles of the plant transfer function are changed by the inner-loop feedback; the zeros remain fixed. The cascade compensator G_C can be used to add zeros to the loop transfer function, and it is designed by the frequency-domain and root-locus techniques and system Type requirements, described earlier.

Figure 3.8-5b is a MIMO control structure and presents additional design difficulties compared to the SISO system. These difficulties arise because the interaction between the input-output pairs of the plant causes the zeros, as well as the poles, of one channel to be changed by the loop closures in the other channel. Furthermore, the extra inputs and outputs demand more complicated design goals. For example, the requirements may call for a noninteracting design, in which commands at one input have minimal effect on the opposite output.

The classical designer could deal with a MIMO plant in two ways. He could design a decoupling network, or precompensator, to precede the plant dynamics and minimize the interaction between the two channels (i.e., diagonalize the transfer function matrix). Alternatively, if the cross-coupling was not a problem, he could simply proceed with a successive loop-closure trial-and-error design. Modern control theory provides additional design techniques for MIMO systems, and these can achieve results that are not possible with a simple decoupling approach. They are described in Chapter 5.

The decoupling network or precompensator is sometimes used in aircraft applications, and the need for this will now be briefly reviewed. For an instructive application to ship dynamics the reader should see Roberts and Towill [1987].

In the case of aircraft, the control surfaces often produce strong moments about more than one axis. If two (or more) independent sets of controls are available, or one set of control surfaces may be used either collectively or differentially, a "control mixer" network may be designed. This can provide pseudo roll, pitch, and yaw inputs, by feeding signals in different proportions to all the control surfaces. We shall encounter a situation similar to this in Chapter 4 when we design an aileron–rudder interconnect (ARI) to feed aileron signals to the rudder control system.

The control mixer approach does not involve any dynamics and its parameters can easily be changed (scheduled) as the aircraft dynamics change with flight conditions. This scheduling is easy to accomplish with an onboard flight control computer. The designer may then follow the classical approach of designing separate controllers for the longitudinal and lateral–directional dynamics.

We shall now review the theory of feedback around a MIMO plant, and a way in which the zeros of the individual MIMO transfer functions may be determined. Assume that the open-loop transfer function matrix for a two-input, two-output system is defined by

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{D(s)} \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (3.8-7)$$

where $D(s)$ is the characteristic polynomial and N_{ij} (not necessarily monic) are the various numerator polynomials. Let the feedback be only from output i to input i , according to

$$\begin{aligned} u_1 &= -k_1(s)y_1 \\ u_2 &= -k_2(s)y_2. \end{aligned} \quad (3.8-8)$$

This "diagonal" (because of the form of the feedback gain matrix) feedback often has a practical rationale, as we shall see later. The closed-loop characteristic equation is now given by (problem 3.8-4)

$$D^2 + (k_1 N_{11} + k_2 N_{22}) D + k_1 k_2 (N_{11} N_{22} - N_{12} N_{21}) = 0. \quad (3.8-9)$$

The classical successive loop-closure approach to designing the feedback control laws for a multiloop system usually starts with the innermost loop if the loops are nested. Usually, a root-locus plot will be constructed for each loop closure, and the design procedure will be repeated until the final closure can be made to yield a satisfactory result. If we close the k_1 loop first (i.e.,

$k_2 = 0$), the loci of the closed-loop poles will start at the open-loop poles given by $D(s) = 0$, and end (when $k_1 = \infty$) on the zeros given by $N_{11} = 0$. When a trial value of k_1 with suitable pole positions has been chosen, the k_2 loop can be closed and the poles will in general move off in new directions determined by a new set of zeros. To find the new set of zeros and use the root-locus rules, we must put (3.8-9) into the form (3.8-2b). This can be done by defining the *coupling numerator*, N_c , according to

$$D(s)N_c(s) \equiv N_{11}N_{22} - N_{12}N_{21}. \quad (3.8-10)$$

The properties of determinants can be used to show that N_c is a polynomial [McRuer et al., 1973]. The characteristic equation (3.8-9) now becomes

$$D + k_1N_{11} + k_2(N_{22} + k_1N_c) = 0. \quad (3.8-11)$$

Therefore, the pertinent zeros are given by the roots of $k_2(N_{22} + k_1N_c) = 0$, and a root-locus plot may be used to choose the gain constant in $k_2(s)$ during the second loop closure. This procedure will now be illustrated with an example.

Example 3.8-1: MIMO System with Successive Loop Closures. Let the coefficient matrices of a two-input, two-output, second-order system be

$$A = \begin{bmatrix} -8 & 3 \\ 4 & -4 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 3 & -1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The transfer function matrix is easy to construct for this low-order system:

$$G(s) = C(sI - A)^{-1}B = \frac{1}{(s+2)(s+10)} \begin{bmatrix} s+13 & s+1 \\ 3s+28 & -(s+4) \end{bmatrix}.$$

Suppose that we now use diagonal feedback again and close the loop $u_1 = -k_1y_1$; the characteristic equation becomes

$$(s+2)(s+10) + k_1(s+13) = 0.$$

Since we are not attempting a realistic design, the feedback gain will be chosen arbitrarily as $k_1 = 3$. The closed-loop poles are then at $s = (-7.5 \pm j\sqrt{11}/2)$, while the SISO zeros remain unchanged at $(-13, -1, -28/3, -4)$.

To see the effect of closing the second loop, we must calculate the coupling numerator according to (3.8-10); the result is $N_c = -4$. The characteristic equation for the second loop closure is then

$$(s^2 + 15s + 59) + k_2[-(s+4) + 3(-4)] = 0,$$

and the zero of the y_2/u_2 transfer function has moved from $s = -4$ to

$s = -16$. A root-locus plot for the second closure could now be constructed from this characteristic equation.

In the theory presented above we have made use of the transfer function matrix and polynomial operations (e.g., to find the coupling numerators) to illustrate the effects of feedback on transfer function poles and zeros. An extensive treatment of coupling numerators can be found in McRuer et al. [1973], where they are tabulated for the various aircraft transfer functions. For computer-aided design we have found a state-equation approach to be more convenient. It is a simple matter to close multiple feedback loops around the matrix state equations, and transfer function poles and zeros can readily be determined at any step with eigenvalue routines and the zeros algorithm given in the preceding section. A wide variety of classical and modern design techniques is compatible with this approach, and in the next subsection we shall build a suitable state-equation framework.

A State-Space Formulation for Feedback Design

Figure 3.8-6 shows a feedback structure that is general enough for most aircraft applications (and many others). The outer unity-feedback loop allows the *performance output*, z , to track the reference input, r . Note that r and z are vectors in the MIMO case. A cascade dynamic compensator (A_D, B_D, C_D, D_D) is allowed for, and inner feedback loops (e.g., rate damping) are permitted through the plant output coupling matrix C_P . The plant and compensator outputs are applied as negative feedback to the plant input through a gain matrix K . The plant is assumed to have a null direct-feed matrix D_P , and the plant dynamics x_P may contain additional compensator dynamics. A state-space description of Fig. 3.8-6 can be written down by inspection; thus

$$\begin{bmatrix} \dot{x}_P \\ \dot{x}_D \end{bmatrix} = \begin{bmatrix} A_P & 0 \\ -B_D H_P & A_D \end{bmatrix} \begin{bmatrix} x_P \\ x_D \end{bmatrix} + \begin{bmatrix} B_P \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ B_D \end{bmatrix} r \quad (3.8-12a)$$

$$\begin{bmatrix} y_P \\ y_D \end{bmatrix} = \begin{bmatrix} C_P & 0 \\ -D_D H_P & C_D \end{bmatrix} \begin{bmatrix} x_P \\ x_D \end{bmatrix} + \begin{bmatrix} 0 \\ D_D \end{bmatrix} r \quad (3.8-12b)$$

$$u = -K \begin{bmatrix} y_P \\ y_D \end{bmatrix} = -[k_P \ k_D] \begin{bmatrix} y_P \\ y_D \end{bmatrix}. \quad (3.8-12c)$$

These equations are of the form

$$\begin{aligned} \dot{x} &= Ax + Bu + Gr, & x \in \mathbf{R}^n, \quad u \in \mathbf{R}^m \\ y &= Cx + Fr, & r \in \mathbf{R}^{m1}, \quad y \in \mathbf{R}^p \\ u &= -Ky. \end{aligned} \quad (3.8-13)$$

Also, note that $z = H_P x_P$, $z \in \mathbf{R}^{m1}$.

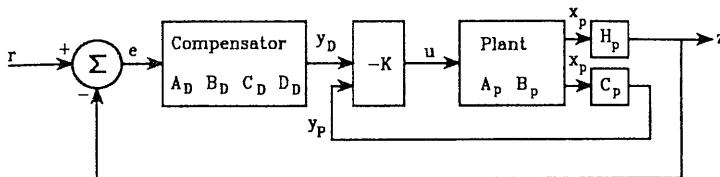


Fig. 3.8-6 A general feedback structure for use in analysis.

When the feedback loop is closed by the gain matrix K , the closed-loop equations are

$$\begin{aligned}\dot{x} &= A_C x + B_C r \\ z &= Hx,\end{aligned}\quad (3.8-14)$$

where

$$A_C = A - BKC, \quad B_C = G - BKF.$$

It is now a simple matter to write a program to vary a specified element of the gain matrix, K , compute the closed-loop poles, and generate a data file for a root-locus plot. Such a program should also have the capability to file A_C and B_C for any chosen gain, so that they may be used to obtain a closed-loop step response (in conjunction with the time-history program). An additional convenient feature is the ability to break the loop at some point to obtain the coefficient matrices for a frequency response. For example, the matrices $A, B, -KC$ will provide the transfer function (or transfer function matrix) corresponding to breaking the loop at the plant input. This transfer function will provide frequency response data to determine gain and phase margins.

Equations (3.8-13) and (3.8-14) will be used for controller design in Chapters 4, 5, and 6; and a program, LOC, was written to use these equations to perform classical root-locus design in Chapter 4. The program accepts the state-space coefficient matrices and closes the feedback loops with a specified K matrix; root-locus data can be generated as any given gain element is varied over a specified range. The use of the state-space equations and the LOC program will now be illustrated on the Type 2 plant and phase-lead compensator used earlier. The gain K will only be a scalar in this simple example.

Example 3.8-2: Root-Locus Design Using the State-Space Formulation. This example illustrates the procedure for putting plant and compensator equations into the form of (3.8-13) and demonstrates a root-locus design using a simple phase-lead compensator on an unstable Type 2 plant.

The plant transfer function is assumed to be

$$G(s) = \frac{100}{s^2(s + 10)}.$$

A partial-fraction expansion of the transfer function was used, in conjunction with the method of Section 3.2, to derive the following coefficient matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -10 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad C = [10 \quad -1 \quad 1].$$

The phase-lead compensator will be assumed to be in the form of (3.8-5), and Fig. 3.8-7 shows a simulation of this compensator cascaded with the plant in a unity feedback arrangement. The figure is helpful in writing the state equations in the form (3.8-13), and the end result is

$$\dot{x} = \begin{bmatrix} A & 0 \\ 0 & 0 \\ -C & -p \end{bmatrix} x + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r$$

$$y = y_D = [-C \mid z - p] x + [1]r.$$

Referring to Fig. 3.8-7, note that in this case we should use negative values of K to obtain implicit negative feedback around the feedback loop. Then the loop transfer function will have a positive value for the static loop sensitivity, and the root locus will lie on the real axis when it is to the left of an odd number of poles and zeros.

The parameters of the lead compensator were chosen as $z = 0.5$ and $p = 5$, based on a rough sketch of the expected behavior of the root-locus plot. The LOC program was run with negative gain selected, and a logarithmic variation of k from 0.01 to 10 with 100 points per decade. The root-locus plot is the plot shown as Fig. 3.8-4, and was commented on earlier.

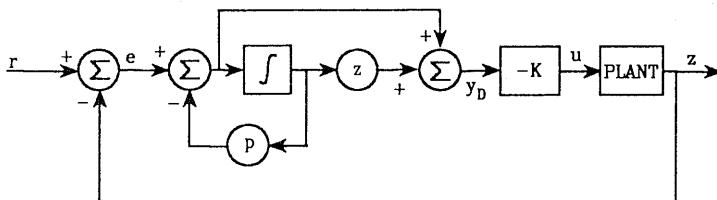


Fig. 3.8-7 A feedback control system with a cascade compensator.

3.9 SUMMARY

In this chapter we have developed all of the components shown in Fig. 3.1-1. Two nonlinear state-space aircraft models have been provided in the form of FORTRAN source code. Programs for trimming, linearization, time-response simulation, poles and zeros, and loop closure on state equations have been described, and some source code is given in Appendix B. All of the development has been illustrated with applications to aircraft, so that the reader should be well prepared for aircraft control system design in Chapter 4. Our review of linear systems and feedback control has been limited to theory and techniques that we use in the text. For additional background material the reader should consult some of the current control theory texts [Brogan, 1991; D'Azzo and Houpis, 1988; Kailath, 1980; Kuo, 1987].

REFERENCES

- AFWAL-TR-84-3105, *Proc. Aeroservoelast. Spec. Meet.*, Oct. 2-3, 1984, Vols. I and II, USAF Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio.
- Brogan, W. L., *Modern Control Theory*, 3rd ed. Englewood Cliffs, N.J.: Prentice Hall, 1991.
- Chen, C. F., and I. J. Haas, *Elements of Control System Analysis*. Englewood Cliffs, N.J.: Prentice Hall, 1968.
- D'Azzo, J. J., and C. H. Houpis, *Linear Control System Analysis and Design*, 3rd ed. New York: McGraw-Hill, 1988.
- DeRusso, P. M., R. J. Roy, and C. M. Close, *State Variables for Engineers*. New York: Wiley, 1965, p. 397.
- Dommasch, D. O., S. S. Sherby, and T. F. Connolly, *Airplane Aerodynamics*, 4th ed. Belmont, Calif.: Pitman, 1967.
- Emami-Naeini, A., and P. Van Dooren, "Computation of zeros of linear multivariable systems," *Automatica*, vol. 18, no. 4, 1982, pp. 415-430.
- Etkin, B., *Dynamics of Atmospheric Flight*. New York: Wiley, 1972.
- Eykhoff, P., *System Identification*. New York: Wiley, 1974.
- Fadeeva, V. N., *Computational Methods of Linear Algebra*. New York: Dover, 1959.
- Gear, C. W., *Numerical Initial Value Problems in Ordinary Differential Equations*. Englewood Cliffs, N.J.: Prentice Hall, 1971.
- Golub, G. H., and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore: The Johns Hopkins University Press, 1989.
- Goodwin, G. C., and K. S. Sin, *Adaptive Filtering Prediction and Control*. Englewood Cliffs, N.J.: Prentice Hall, 1984.
- Hamming, R. W., *Numerical Methods for Scientists and Engineers*. New York: McGraw-Hill, 1962.
- Harris, C., and S. A. Billings, eds., *Self-Tuning and Adaptive Control: Theory and Applications*. London: Peter Peregrinus, 1981.

- Healey, M., "Study of methods of computing transition matrices," *Proc. IEE*, vol. 120, no. 8, Aug. 1973, pp. 905-912.
- Hindmarsh, A. C., "Large ordinary differential equation systems and software," *IEEE Control Syst.*, Dec. 1982, pp. 24-30.
- Hodgkinson, J., "Equivalent systems approach for flying qualities specification," MCAIR 79-017, presented at *SAE Aerosp. Control Guid. Syst. Comm. Meet.*, Denver, Colo., Mar. 7-9, 1979.
- IMSL, *Library Contents Document*, 8th ed., International Mathematical and Statistical Libraries, Inc., 7500 Bellaire Blvd., Houston, TX, 77036, 1980.
- Isaacson, E., and H. B. Keller, *Analysis of Numerical Methods*. New York: Wiley, 1966.
- Kailath, T., *Linear Systems*. Englewood Cliffs, N.J.: Prentice Hall, 1980.
- Kuo, B. C., *Automatic Control Systems*. Englewood Cliffs, N.J.: Prentice Hall, 1987.
- Laning, J. H., and R. H. Battin, *Random Processes in Automatic Control*. New York: McGraw-Hill, 1956, Appendix C.
- Laub, A. J., and B. C. Moore, "Calculation of transmission zeros using QZ techniques," *Automatica*, vol. 14, 1978, pp. 557-566.
- MacFarlane, A. G. J., and N. Karcanias, "Poles and zeros of linear multivariable systems, a survey of the algebraic, geometric, and complex-variable theory," *Int. J. Control*, vol. 24, 1976, pp. 33-74.
- Maine, R. E., and K. W. Iliffe, "Formulation and implementation of a practical algorithm for parameter estimation with process and measurement noise," paper 80-1603, *AIAA Atmos. Flight Mech. Conf.*, Aug. 11-13, 1980, pp. 397-411.
- McKern, R. A., "A study of transformation algorithms for use in a digital computer," M. S. thesis, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, Jan. 1968.
- McRuer, D., I. Ashkenas, and D. Graham, *Aircraft Dynamics and Automatic Control*. Princeton, N.J.: Princeton University Press, 1973.
- Moler, C., and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM Rev.*, vol. 20, no. 4, Oct. 1978, pp. 801-836.
- Nelder, J. A., and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, 1964, pp. 308-313.
- Nguyen, L. T., et al., "Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability," *NASA Tech. Pap. 1538*, NASA, Washington, D.C., Dec. 1979.
- Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*. New York: Cambridge University Press, 1986.
- Ralston, A., *A First Course in Numerical Analysis*. New York: McGraw-Hill, 1965.
- Roberts, G. N., and D. R. Towill, "Maneuvering control of a modern warship," *Proc. Am. Control Conf.*, Minneapolis, Minn., 1987, pp. 2109-2113.
- Rosenbrock, H. H., *State-Space and Multivariable Theory*. London: Nelson, 1970.
- Shampine, L. F., and M. K. Gordon, *Solution of Ordinary Differential Equations: The Initial Value Problem*. San Francisco: W. H. Freeman, 1975.

Yuan, S. W., *Foundations of Fluid Mechanics*. Englewood Cliffs, N.J.: Prentice Hall, 1967.

Zakian, V., "Rational approximants to the matrix exponential," *Electron. Lett.*, vol. 6, no. 5, Dec. 10, 1970, pp. 814-815.

PROBLEMS

Section 3.2

- 3.2-1** (a) Derive the transfer function of the phase-lead network in Table 3.2-1 (fourth network).
 (b) Derive the state equations.
- 3.2-2** Derive a set of nonlinear state equations for the spool valve and hydraulic actuator described in Example 3.2-1.

Section 3.3

- 3.3-1** (a) Program the F-16 model in Section 3.3.
 (b) Link the program of part (a) with another program to run the test case given in Table 3.3-2.

Section 3.4

- 3.4-1** (a) Program the transport aircraft model in Section 3.3.
 (b) Use an existing optimization algorithm, or devise your own algorithm, to adjust the three variables (thtl , el , α) and reproduce the steady-state trim conditions given in Table 3.4-1.
 (c) Modify your program so that nonzero flight-path angles can be specified, and use it to find out how steeply the aircraft (in clean configuration, with $x_{cg} = 0.25\bar{c}$) can climb for a range of speeds from 200 to 500 ft/s, at sea level. Compute the rate of climb (ROC) for each speed and determine the speed at which the ROC is a maximum.
- 3.4-2** (a) Use your trim program from Problem 3.4-1 to trim the F-16 model longitudinally (lateral/directional states and controls set to zero). Duplicate some of the longitudinal trims given in Table 3.4-2.
 (b) Extend your trim program to trim the complete F-16 model, omit the flight-path constraints and fix the bank angle at the value given by (3.4-6) or (3.4-7) for level flight. Reproduce the trim conditions given in Table 3.4-3.
- 3.4-3** (a) Derive (3.4-3) for the pitch attitude in terms of the flight-path angle.

- (b) Incorporate the flight-path constraints in your program and trim the F-16 model in a climbing coordinated turn with $\gamma = 10^\circ$ and a turn rate of 0.1 rad/s at 10,000 ft altitude with $V_T = 502$ ft/s and $x_{cg} = 0.35\bar{c}$.
- 3.4-4** Modify your trim program used in Problem 3.4-1 to trim the transport aircraft for a prescribed angle of attack by varying V_T . Derive a trim condition for $\alpha = 15^\circ$ at 10,000 ft.

Section 3.5

- 3.5-1** Program the Runge-Kutta integration routine given in Appendix B, also program a simple driver program for time-history simulation. Incorporate the state equations for the quadratic-lag network given in Table 3.2-1 into a subroutine F of the form given in the text. Use these programs to compute the unit-step response of the quadratic lag when $\zeta = 0.5$ and $\omega_n = 1$ rad/s, and plot the results.
- 3.5-2** Use your trim and time-history programs to duplicate the results of Examples 3.5-1, 3.5-2, and 3.5-3.

Section 3.6

- 3.6-1** Program the linearization subroutine given in Appendix B, write a driver program, and use this to duplicate the results of Example 3.6-1.
- 3.6-2** Write a program to compute the matrices $E^{-1}A$ and $E^{-1}B$, for the decoupled longitudinal equations given in Section 2.5, from the stability derivatives. Test it on the transport aircraft model and compare the results with those given in Example 3.6-2.

Section 3.7

- 3.7-1** Derive a formula similar to (3.7-4), but covering two sample intervals, by evaluating the integral in (3.7-3) by Simpson's rule.
- 3.7-2** Make the same kind of assumptions as in Example 3.7-1, and derive a transition matrix solution to the strapdown equation (1.4-12).
- 3.7-3** The inertial navigation system described in Example 3.7-2 and illustrated in Fig. 3.7-1 contains a feedback loop around two integrators. Discuss the stability of this loop, determine the frequency of any unstable oscillations, and speculate on the implications.
- 3.7-4** Write a program to use a standard eigenvalue/eigenvector routine and check the results of Examples 3.7-3 and 3.7-4.
- 3.7-5** Write a program to determine transfer function poles and zeros from the A, B, C, D matrices using the zeros algorithm described in Sec-

tion 3.7 and any standard eigenvalue routine. Use it to check the results of Examples 3.7-5 and 3.7-6.

- 3.7-6 Run linear and nonlinear time-history simulations of a throttle step input to the transport aircraft model under the conditions given in Example 3.7-6. Compare the various speed and altitude responses and comment on these in the light of the transfer function properties discussed in the example.
- 3.7-7 Write a FORTRAN program, using complex arithmetic, to evaluate frequency response from a set of poles and zeros. Use it to check the results of Example 3.7-7.

Section 3.8

- 3.8-1 Write a program like the program LOC used in Example 3.8-2. The program should close output-feedback loops through a gain matrix K on a set of state equations in standard form, or in the form of (3.8-13). Make provision to vary any single element of the gain matrix automatically and file the data for a root-locus plot. Use the program to check the results of Example 3.8-2.
- 3.8-2 Use a frequency-response program to design a lead compensator for the unity-feedback control system in Example 3.8-2 [forward-path transfer function $100/(s^2(s + 10))$]. Use the lead compensator transfer function given in Section 3.8, with a pole-to-zero ratio of 10. Design for the largest possible loop gain consistent with a gain margin of at least 12dB and
(a) a 30° phase margin,
(b) a 45° phase margin.
Derive the state equations with the compensator included (as in Example 3.8-2), close the loop with the appropriate gain matrix (see Problem 3.8-1), and compare the step responses of these two designs.
- 3.8-3 In Problem 3.8-2(a), move the compensator out of the forward path and into the feedback path, and repeat the step-response simulation with a 0.1 amplitude step. Compare the results with those of Problem 3.8-2(a) and comment on the implications for design.
- 3.8-4 Derive Equation (3.8-9).

CHAPTER 4

AIRCRAFT DYNAMICS AND CLASSICAL DESIGN TECHNIQUES

4.1 INTRODUCTION

In the previous chapters we have developed mathematical tools, realistic aircraft models, and algorithms for performing flight simulation and flight controls design. Before we attempt to use all of these tools, models, and algorithms, we must have a clear idea of their applicability and the rationale and design goals for automatic flight control systems. Some idea of the history of the development of automatic flight controls is helpful in this respect.

Historical Perspective

The success of the Wright brothers in achieving the first powered flight in December 1903 has been attributed to both their systematic design approach (they built and used a wind tunnel), and the emphasis they placed on making their aircraft controllable by the pilot rather than inherently stable. However, the difficulties of controlling the early aircraft and the progress toward longer flight times led quickly to the development of an automatic control system. Thus in 1912 an autopilot was developed by the Sperry Gyroscope Company and tested on a Curtiss flying boat. By 1914, the "Sperry Aeroplane Stabilizer" had reached such a state of development that a public flying demonstration was given in which the mechanic walked along the wing while the pilot raised his hands from the controls.

World War I (1914–1918) provided the impetus for great progress in aircraft design. However, a human pilot was perfectly capable of providing the normal stabilizing and control functions for the aircraft of this era, and the time was not ripe for rapid developments in automatic control. The small perturbation theory of aircraft dynamics had been developed [Bryan, 1911] and in the 1920s stability derivatives were measured and calculated, and the theory was confirmed by flight tests. Little practical use was made of the theory because even the problem of finding the roots of a quartic equation was difficult at this time. Development of autopilots continued, using gyroscopes as the reference sensor and pneumatic servomechanisms to position the control surfaces. A Sperry autopilot also made it possible for Wiley Post to fly around the world in less than eight days in 1933.

In the late 1930s classical control theory began to develop. The need to design stable telephone repeater amplifiers with closely controlled gain led to the work of Black in “regeneration theory” and to Nyquist’s frequency-domain stability criterion. The same stimuli also led to Bode’s complex-frequency-domain theory for the relationships between gain and phase and his logarithmic plots of gain and phase. World War II (1939–1945) led to further developments in control theory because of the need for radar control and tracking, and the development of servomechanisms for positioning guns and radar antennas. Once again wartime spurred improvements in aircraft design. The large expansion of the speed–altitude envelope and the need to carry, and dispose of, large payloads led to large variations in the aircraft dynamics, thus creating a need to analyze the dynamic behavior. Larger aircraft required power-boosted control surfaces, and developments in hydraulic servomechanisms resulted. Also, the need to fly at night and in bad weather conditions led to developments in radio-navigation aids and a need to couple the navigation system to the autopilot. Thus, in 1947, a U.S. Air Force C-53 made a transatlantic flight, including takeoff and landing, completely under the control of an autopilot.

By the late 1940s the concepts of frequency response and transfer functions had become more generally known and the first analog computers were becoming available. The root-locus technique, published by W. R. Evans in 1948, was a major development in analyzing and designing control systems (it is even more useful in the computer age!). Analyses of the stability and performance of aircraft under automatic control began to be performed more commonly by the aircraft companies. The aircraft altitude–speed envelope was being expanded rapidly by the first jet fighters and by a series of research aircraft (the “X” series in the United States).

The rocket-powered Bell X-1 aircraft made its first flight in January 1946, in October 1947 it achieved supersonic flight, and in August 1949 an altitude of nearly 72,000 ft was reached. The envelope was extended further by the next generation of X-planes, X-1A through X-1D. After reaching Mach 2.44 and 75,000 ft altitude, *inertia coupling* (see Sections 1.3 and 4.5) caused the X-1A to spin around all three axes, almost killing the pilot, Major Charles

Yeager. Inertia coupling effects were encountered because these aircraft had the basic form of a modern jet fighter with short stubby wings, most of the mass concentrated along the longitudinal axis, and relatively small tail surfaces for directional stability. Before the problem was fully understood, a number of aircraft of the period suffered inertia coupling effects, sometimes with disastrous results. These included the X-2 and X-3 and the F-100 jet fighter during the course of its production program in 1953.

Many other factors besides inertia coupling contributed to the need for a strong analytical approach to aircraft stability and control problems. The changes in aircraft mass properties, together with the need to reduce the area of the aerodynamic surfaces (for lower drag at high speed), caused changes in the natural modes of the aircraft, so that they were no longer easily controllable by the pilot. In addition, the damping of the natural modes tended to decrease as the altitude limits of the airplanes were expanded; these factors made it more important to predict the frequency and damping of the modes analytically. Also, the expansion of the aircraft speed-altitude envelope meant that much greater variations in the dynamics of the aircraft were encountered.

Power-boosted or fully powered control surfaces were introduced because of the increasing aerodynamic loads associated with greater performance and larger aircraft, and because they could eliminate the many hours of flight test needed to balance the control surfaces carefully. Properly balanced control surfaces were previously necessary to provide a suitable feel to the pilot's controls. With power-boosted controls the feel could be modified with springs and bobweights, and with fully powered irreversible controls the feel could be provided completely artificially. Thus the "handling qualities" of the aircraft could be adjusted to be satisfactory over a very wide envelope. Power-boosted controls also made possible the use of *stability augmentation*, in which signals from angular rate sensors could be fed to the control surface actuators to modify the natural modes of the aircraft. In addition, they facilitated the use of more complex autopilots.

The year 1949 saw the first flight of the de Havilland Comet, and this aircraft essentially defined the modern jet transport aircraft. In the early 1950s the problems of supersonic flight up to Mach 3 and beyond were beginning to be investigated. The Lockheed X-7 unmanned rocket plane was built to provide a testbed for a ramjet engine. During a five-year test program beginning in 1951, it also provided information on high-speed aerodynamics, aerothermodynamics, special fuels, and special materials. Data from programs such as this undoubtedly contributed to the design of aircraft such as the F-104 and the SR-71. The X-15 rocket plane, which first flew in 1959, expanded the envelope for manned flight to beyond Mach 6 and above 300,000 ft. This aircraft was equipped with a Honeywell-designed adaptive control system that provided three-axis stability augmentation and a transition from aerodynamic control to reaction control as the aerodynamic controls became ineffective at high altitude.

In the early 1960s small fighter aircraft were approaching Mach 2 speeds, a French Mirage achieved Mach 2.3, and later an F-4 Phantom made a record-breaking Mach 2.4 flight. In the civil aviation field, this was the time of the Boeing 707 and Douglas DC8 passenger jets and the development of the Aerospatiale/British Aerospace Concorde SST. The digital computer was beginning to have a major impact on engineering, the techniques of numerical analysis assumed greater importance, and this stimulated the growth of modern control theory in the mid-1960s.

A great deal of hypersonic aerodynamics knowledge was gained from the X-15 program and from hypersonic wind-tunnel studies in the late 1950s. The X-20 (Dyna-Soar) vehicle, to be built by Boeing under a 1960 contract, was to be a rocket-launched unpowered glider that would gather data to solve the problems of pilot-controlled reentry from orbit. The final design was a unique V-shaped vehicle with a thick wing and upturned wingtips. Although the program was cancelled before completion of the first vehicle, it pioneered the technology for the U.S. space shuttle. Later, the unmanned ASSET (1963 to 1965) and PRIME (X-23A; 1966 to 1967) vehicles provided flight data on structures, materials, control systems, and other technologies for maneuvering reentry. These were followed in 1969 and the early 1970s by the X-24 manned, blunt lifting-body vehicles. These provided data on the low-speed characteristics of maneuverable reentry vehicles, including stability characteristics, pilot experience for comparison with simulators, man-vehicle interface data, and much control system information.

Because of the digital computer the 1970s saw great strides in computational fluid dynamics, structural and flutter (structural divergence) analysis, simulation of complex dynamical systems, and the application of guidance and control theory in real-time onboard digital computers. Simulation techniques made possible realistic pilot training on the ground, and the automatic flight control system on board an aircraft allowed the dynamic behavior of an entirely different aircraft to be simulated. Thus space-shuttle pilots trained on a Gulfstream-II aircraft that simulated the feel of the space shuttle.

Flight control technology had advanced to a stage such that the F-16 aircraft could be designed for "relaxed static stability" and all-electric (full "fly-by-wire") control in the 1970s. Previous aircraft had used "high-authority" electrical control superimposed on the basic electrohydraulic system (e.g., the F-111), or as in the case of the Concorde, an electrical system with mechanical backup. The processing of the electrical signals for the automatic flight control systems was still in analog rather than digital form.

The 1980s saw the flight testing of aircraft with additional aerodynamic control surfaces that provided direct-lift control or direct-sideforce control (such as the AFTI F-16 and the Grumman forward-swept wing X-29A aircraft) and with digital flight control systems (e.g., McDonnell F-15E and F-18). The AFTI F-16 aircraft allowed the use of sideforce control through a ventral fin, and direct-lift control through the combination of the horizontal tail and wing leading-edge flaps. The *decoupled motions* provided by this

control were evaluated for use in combat situations. The X-29A research aircraft is unstable in pitch (-35% static margin at low speed) and has three-surface pitch control (canards, wing flaperons, and strake flaps). The flight control system is a triply redundant digital system (three digital processors with "voting" to eliminate a faulty channel) with analog backup for each processor. These aircraft raise interesting multivariable control problems for modern control theory.

The U.S. space shuttle made its first flight in March 1981, and there was also a resurgence of interest in hypersonic flight during the 1980s. Single-stage-to-orbit vehicles were studied, including the British HOTOL (horizontal takeoff and landing) unmanned satellite launch vehicle and the U.S. TAV (Trans-Atmospheric Vehicle), fully reusable rapid-turnaround vehicles for manned reconnaissance, weapon delivery, and delivery of large payloads to orbit. These were followed in the United States by the NASP (National Aerospace Plane) study contracts on a manned single-stage-to-orbit vehicle. Other studies looked at boosted vehicles; these included the French HERMES vehicle (similar to the space shuttle, manned), and a number of U.S. BGVs (boost-glide vehicles).

Many lessons were learned about the control of hypersonic vehicles. The trajectories must be carefully controlled because the frictional heating in the atmosphere can create temperatures of a few thousand degrees Fahrenheit at critical points on the vehicle. A change in flight conditions can cause localized changes in the airflow, from laminar to turbulent flow, and this can lead to a rapid increase in temperature at some point on the surface of the vehicle. Manual control is difficult or not feasible in most flight phases and, if attempted, would limit the performance. The trajectory can be controlled by feedback comparison with a precomputed reference trajectory, or with real-time trajectory-prediction calculations (as in the case of the space shuttle).

There can be large uncertainties in the aerodynamic coefficients of the hypersonic vehicles, and this complicates the design of the automatic control systems and limits their performance. The control systems must be adapted (gain-scheduled, or self-adaptive) in flight, to allow for the wide variations in vehicle dynamics over the large flight envelope. If fixed "scheduling" is used, difficulties are encountered in sensing the flight conditions. External probes sensing "air data" (dynamic pressure and Mach) can only be used at low Mach numbers, and the air data must be derived from the navigation system and a stored model of the atmosphere. The real atmosphere can show large unpredictable variations in density; therefore, the control systems must be designed to tolerate these variations.

The era of true "aerospace" vehicles introduces many new challenges for the control engineer. He or she must now think in terms of guidance and control, algorithms and simulation, and numerical methods and digital implementation. Many relatively new analytical techniques are required, including numerical optimization, analysis of sensitivity and robustness to parameter variations, adaptive techniques, and multivariable control. Furthermore, the

control engineer can no longer work in isolation; many other technologies will be closely integrated into a design, and constraints will be imposed on the designs from a variety of sources (e.g., structural, thermal, propulsion, energy management and performance, and human factors).

The Need for Automatic Control Systems

The evolution of modern aircraft created a need for power-driven aerodynamic control surfaces and automatic-pilot control systems, as described in the preceding subsection. In addition, the widening performance envelope created a need to augment the stability of the aircraft dynamics over some parts of the envelope. This need for stability augmentation is now described in more detail.

Figure 4.1-1 shows the altitude–Mach envelope of a modern high-performance aircraft; the boundaries of this envelope are determined by a number of factors. The low-speed limit is set by the maximum lift that can be generated (the alpha limit in the figure), and the high-speed limit follows a constant dynamic pressure contour (because of structural limits, including temperature). At the higher altitudes the speed becomes limited by the maximum engine thrust (which has fallen off with altitude). The altitude limit imposed on the envelope is where the combination of airframe and engine characteristics can no longer produce a certain minimum rate of climb (this is the “service ceiling”).

The aircraft envelope covers a very wide range of dynamic pressure. For example, in the landing phase the dynamic pressure may be as low as 50 psf,

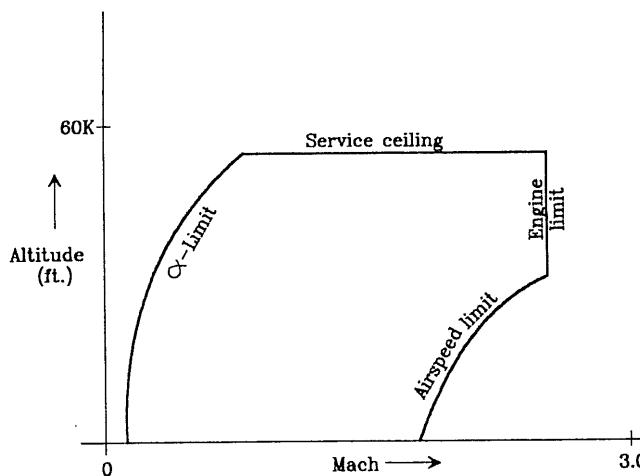


Fig. 4.1-1 Aircraft altitude–speed envelope.

whereas at Mach 1.2 at sea level the dynamic pressure is 2150 psf. Large variations in dynamic pressure cause correspondingly large variations in the coefficients of the dynamic equations. Other factors also contribute to changes in the aircraft dynamics. The basic aerodynamic coefficients change with Mach number, and as functions of the aerodynamic angles, and the mass properties change with different payloads and changing fuel load.

Because of the large changes in aircraft dynamics, a dynamic mode that is stable and adequately damped in one flight condition may become unstable, or at least inadequately damped, in another flight condition. A lightly damped oscillatory mode may cause a great deal of discomfort to passengers or make it difficult for the pilot to control the trajectory precisely. These problems are overcome by using feedback control to modify the aircraft dynamics. The aircraft motion variables are sensed and used to generate signals that can be fed into the aircraft control-surface actuators, thus modifying the dynamic behavior. This feedback must be adjusted according to the flight condition. The adjustment process is called *gain scheduling* because, in its simplest form, it involves only changing the amount of feedback according to a predetermined schedule. The scheduling variable will normally be measured dynamic pressure but will involve other variables in more complicated cases.

In the case of low-performance aircraft with relatively narrow envelopes and control surfaces that are not power driven, an unsatisfactory dynamic mode must be corrected by modifying the basic design. As in the case of the high-performance aircraft, this requires an understanding of the dynamic modes and their dependence on the aerodynamic coefficients and aerodynamic derivatives.

Figure 4.1-2 shows how a fully powered aircraft control system might be implemented, with mechanical, hydraulic, and electrical components. Because the control surfaces are fully power driven, there is no force or motion feedback to the pilot's control stick. This is called an *irreversible* control

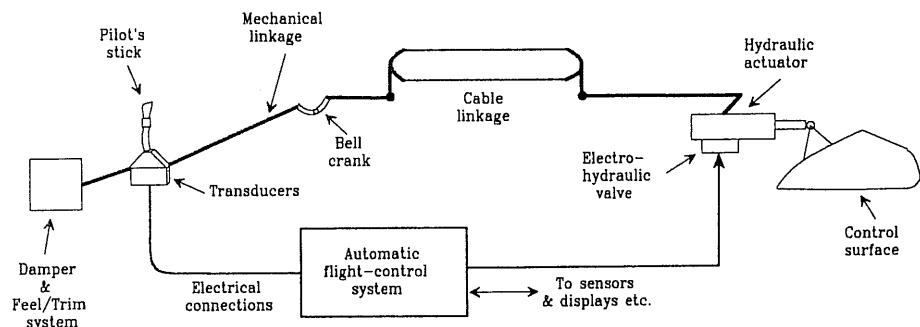


Fig. 4.1-2 Aircraft electromechanical control system.

system, and bobweights and springs (or electrical or hydraulic devices) must be added to the control stick to provide some "feel" to the pilot. The stick and rudder pedals are shown linked to the actuators by a combination of mechanical links and bell cranks, and control wires. The control surfaces are driven by a hydraulic servomechanism that has a follow-up action; that is, the high-power output shaft is driven until its position corresponds to the position of the low-power input shaft.

Augmentation signals are conveniently added to the system of Fig. 4.1-2 by electrical means. The signals from rate gyros (angular-rate measuring devices), accelerometers, the air-data computer, and other sources are processed by the flight-control computer. The electrical output of the flight-control computer (converted to analog form) is used to drive electrohydraulic valves, and these superimpose additional motion on the hydromechanical control system.

The Functions of the Automatic Control Systems

The descriptions and analyses of aircraft modes in Chapter 3 show that they can be divided into different categories. One category includes modes that involve mainly the rotational degrees of freedom; these are the short-period, roll, and dutch roll modes. Their natural frequencies (or time constants, if purely exponential) are determined by the moments of inertia of the aircraft and the moments generated by the aerodynamic surfaces; and their damping is determined by the rate-dependent aerodynamic moments. The remaining modes (phugoid and spiral) involve changes in the flight path and are much slower modes. The phugoid mode involves the translational degrees of freedom and is dependent on the aerodynamic forces of lift and drag, and their variation with speed. The spiral mode depends on aerodynamic moments, but only weak aerodynamic forces are involved.

The responsiveness of an aircraft to maneuvering commands is determined by the speed of the rotational modes. The frequencies of these modes tend to be sufficiently high that a pilot would find it difficult or impossible to control the aircraft if the modes were lightly damped or unstable. Therefore, it is necessary to provide automatic control systems to give these modes suitable damping and natural frequencies. Such control systems are known as *stability augmentation systems* (SAS). If the augmentation system is intended to control the mode and to provide the pilot with a particular type of response to the control inputs, it is known as a *control augmentation system* (CAS). An example of this is a normal-acceleration CAS, in which the pilot's inputs are intended to control the acceleration generated along the negative *z*-axis.

The slow modes (phugoid and spiral) are controllable by a pilot. But since it is undesirable for a pilot to have to pay continuous attention to controlling these modes, an automatic control system is needed to provide "pilot relief." An *autopilot* is an automatic control system that provides both pilot relief functions and special functions such as automatic landing.

The common types of SAS, CAS, and the autopilot functions can be listed as follows:

SAS	CAS	Autopilots
Roll damper	Roll rate	Pitch attitude hold
Pitch damper	Pitch rate	Altitude hold
Yaw damper	Normal acceleration Lateral/directional	Mach hold Automatic landing Bank-angle hold Turn coordination Heading hold/VOR hold

These control systems are described and illustrated by numerical examples in Sections 4.4 to 4.7.

4.2 THE EFFECT OF FLIGHT CONDITIONS ON THE MODES

In this section algebraic expressions for the rigid-body modes will be derived so that their dependence on the stability derivatives and on the flight conditions can be examined, and so that conditions for stability can be deduced. When lat-long decoupling occurs, it becomes feasible to manipulate the aircraft transformed state equations algebraically. Both the longitudinal and lateral-directional dynamics are still fourth-order, so the modes are obtained from the roots of a fourth-order characteristic polynomial. Algebraic solution of a quartic equation is not practicable, but with some simplifying assumptions based on knowledge of the stability derivatives and the physics of flight, this problem can be bypassed.

Transfer function numerators are not required for these purposes, nor even complete denominators. Also, for design in the s -domain we shall use the algorithms of Section 3.7 to obtain poles and zeros directly from the state equations. Nevertheless, for the sake of completeness, comparison with other texts, and additional insight, we shall show how the relevant transfer functions may be derived.

Algebraic Derivation of Longitudinal Transfer Functions and Modes

The coefficient matrices for the decoupled longitudinal state equations are given in (2.5-31). The SISO transfer functions can be derived very easily by applying Cramer's rule to the Laplace transformed state equations, as fol-

lows. The matrix $(sE - A)$ is given by

$$(sE - A) = \begin{bmatrix} s - (X_V + X_{T_V} \cos \alpha_e) & -X_\alpha & g'_0 \cos \gamma_e & 0 \\ -(Z_V - X_{T_V} \sin \alpha_e) & s(V_T - Z_{\dot{\alpha}}) - Z_\alpha & g'_0 \sin \gamma_e & -(V_T + Z_q) \\ 0 & 0 & s & -1 \\ -(M_V + M_{T_V}) & -sM_{\dot{\alpha}} - M_\alpha & 0 & s - M_q \end{bmatrix}, \quad (4.2-1)$$

and the B matrix is

$$B = \begin{bmatrix} X_{\delta_{th}} \cos \alpha_e & X_{\delta e} \\ -X_{\delta_{th}} \sin \alpha_e & Z_{\delta e} \\ 0 & 0 \\ M_{\delta_{th}} & M_{\delta e} \end{bmatrix}.$$

If, for example, the $q/\delta e$ transfer function is required, the second column of B must be substituted for the fourth column of $|sE - A|$. The transfer function is

$$\frac{q(s)}{\delta_e(s)} = \frac{1}{|sE - A|} \begin{vmatrix} s - (X_V + X_{T_V} \cos \alpha_e) & -X_\alpha & g'_0 \cos \gamma_e & X_{\delta e} \\ -(Z_V - X_{T_V} \sin \alpha_e) & s(V_T - Z_{\dot{\alpha}}) - Z_\alpha & g'_0 \sin \gamma_e & Z_{\delta e} \\ 0 & 0 & s & 0 \\ -(M_V + M_{T_V}) & -sM_{\dot{\alpha}} - M_\alpha & 0 & M_{\delta e} \end{vmatrix}. \quad (4.2-2)$$

It is evident from inspection of the determinant that this transfer function is of the form

$$\frac{q(s)}{\delta_e(s)} = \frac{s(b_2 s^2 + b_1 s + b_0)}{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0}. \quad (4.2-3)$$

Expressions for the numerator and denominator coefficients can be derived in a straightforward way by expanding the determinants, and they are given in standard texts. However, the coefficients are quite complicated functions of the dimensional derivatives; they provide very little insight and are tedious to evaluate without a digital computer. This is a feasible method of deriving transfer functions from the stability derivatives, but it relies on lat-long decoupling and also would be very cumbersome to extend to higher-order

longitudinal dynamics. We shall now examine various approximations that lead to transfer functions that are simple enough to provide some insight into the dynamic behavior.

Consider the decoupled longitudinal dynamics; a time-history simulation in Section 3.5 showed that it was possible to excite separately the short-period and phugoid modes. In the phugoid case, speed and theta varied, with alpha and q almost constant; while in the short-period case, alpha, q , and theta varied, with speed constant. This implies additional decoupling in the dynamic equations that will now be investigated.

Assume that the variables in the longitudinal state vector are reordered, and the linear equations are partitioned, as follows:

$$x^T = [\alpha, q, v_T, \theta] \quad (4.2-4)$$

$$\dot{x} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} x + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} u. \quad (4.2-5)$$

Now if it is to be possible for v_T and θ to vary, without significant changes in α and q , 2×2 submatrix A_{12} must introduce very little coupling from the second set of equations into the first. An examination of the appropriate terms of the matrix $E^{-1}A$ in Example 3.6-2 shows that this is the case in that particular example. Therefore, a short-period approximation to the longitudinal dynamics can be obtained by neglecting A_{12} and extracting the alpha and q equations from the complete dynamics. We shall now derive the transfer functions for the short-period approximation.

The Short-Period Approximation

If the α and q equations are extracted from the algebraic longitudinal equations (2.5-31), the result is

$$\begin{bmatrix} V_T - Z_{\dot{\alpha}} & 0 \\ -M_{\dot{\alpha}} & 1 \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} Z_\alpha & V_T + Z_q \\ M_\alpha & M_q \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_{\delta e} \\ M_{\delta e} \end{bmatrix} \delta_e. \quad (4.2-6)$$

The transfer function matrix is given by

$$C(sE - A)^{-1}B = \frac{C}{\Delta_{sp}} \begin{bmatrix} (s - M_q)Z_{\delta e} + (V_T + Z_q)M_{\delta e} \\ (sM_{\dot{\alpha}} + M_\alpha)Z_{\delta e} + [s(V_T - Z_{\dot{\alpha}}) - Z_\alpha]M_{\delta e} \end{bmatrix},$$

where C is the appropriate coupling matrix for α and q , and Δ_{sp} is the short-period characteristic polynomial:

$$\begin{aligned} \Delta_{sp} = & (V_T - Z_{\dot{\alpha}})s^2 - [Z_\alpha + (V_T - Z_{\dot{\alpha}})M_q + (V_T + Z_q)M_{\dot{\alpha}}]s \\ & + M_q Z_\alpha - (V_T + Z_q)M_\alpha. \end{aligned} \quad (4.2-7)$$

The derivatives $Z_{\dot{\alpha}}$ and Z_q are usually neglected, and the final expressions for the individual transfer functions are

$$\frac{\alpha}{\delta_e} = \frac{Z_{\delta e}s + (V_T M_{\delta e} - M_q Z_{\delta e})}{V_T s^2 - (Z_\alpha + V_T M_q + V_T M_{\dot{\alpha}})s + M_q Z_\alpha - V_T M_\alpha} \quad (4.2-8)$$

$$\frac{q}{\delta_e} = \frac{(V_T M_{\delta e} + Z_{\delta e} M_{\dot{\alpha}})s + (M_\alpha Z_{\delta e} - Z_\alpha M_{\delta e})}{V_T s^2 - (Z_\alpha + V_T M_q + V_T M_{\dot{\alpha}})s + M_q Z_\alpha - V_T M_\alpha}. \quad (4.2-9)$$

If the denominator is compared with the quadratic standard form

$$s^2 + 2\zeta\omega_n s + \omega_n^2,$$

then, assuming a complex-conjugate pair of roots, the short-period natural frequency and damping are found to be given by

$$\omega_{n_{sp}} = \left(\frac{M_q Z_\alpha}{V_T} - M_\alpha \right)^{1/2}, \quad \zeta_{sp} = \frac{-(M_q + M_{\dot{\alpha}} + Z_\alpha/V_T)}{2\omega_{n_{sp}}}. \quad (4.2-10)$$

When the dimensionless derivatives are substituted into these expressions, the results are

$$\omega_{n_{sp}} = \frac{1}{2} \rho V_T S \bar{c} \left[\frac{-C_{m_q}(C_D + C_{L_\alpha}) - (4m/\rho S \bar{c}) C_{m_\alpha}}{2m J_Y} \right]^{1/2} \quad (4.2-11)$$

$$\zeta_{sp} = \frac{-\bar{c}}{4} \left[\frac{m}{J_Y} \right]^{1/2} \frac{C_{m_q} + C_{m_\alpha} - 2J_Y(C_D + C_{L_\alpha})/(\bar{c}^2 m)}{\left[-\frac{1}{2} C_{m_q}(C_D + C_{L_\alpha}) - 2m C_{m_\alpha}/(\rho S \bar{c}) \right]^{1/2}}. \quad (4.2-12)$$

Normally, the equilibrium drag coefficient is much smaller than the lift-curve slope ($C_D \ll C_{L_\alpha}$), so if the derivatives are relatively constant (e.g., for low subsonic Mach numbers), the frequency of the short period depends on the flight condition only through ρ and V_T . Therefore, *the frequency increases with dynamic pressure* but not in simple proportion. If the aircraft has a high degree of pitch stiffness, the C_{m_α} term may dominate the frequency expression, and the short-period frequency will be roughly proportional to the square root of dynamic pressure. If the pitch stiffness becomes negative (i.e., $C_{m_\alpha} > 0$), then since $C_{L_\alpha} > 0$ and $C_{m_q} < 0$, the quantity in brackets in (4.2-11) will eventually become negative. The short-period characteristic equation then has real roots; this condition will be examined later.

Similar considerations apply to the short-period damping ratio; the equilibrium drag coefficient can be neglected and the damping then depends on the flight condition only through ρ . *The damping therefore decreases with altitude*. If the pitch stiffness term dominates the denominator of (4.2-12), the damping will be roughly proportional to $\sqrt{\rho}$.

We must next determine the accuracy of these results and also consider the stability of the characteristic roots. Before proceeding with this, we derive similar results for the phugoid mode, but without deriving a transfer function.

The Phugoid Approximation

Approximations for the natural frequency and damping of the phugoid mode will be developed by extending the approach used to derive the short-period results. Refer again to (4.2-5) and assume that only the phugoid mode has been excited. If the derivatives $\dot{\alpha}$ and \dot{q} are then neglected, the first pair of equations reduce to algebraic equations that act as a constraint on the remaining differential equations in the phugoid variables. Therefore, we have

$$0 = A_{11} \begin{bmatrix} \alpha \\ q \end{bmatrix} + A_{12} \begin{bmatrix} v_T \\ \theta \end{bmatrix}$$

$$\begin{bmatrix} \dot{v}_T \\ \dot{\theta} \end{bmatrix} = A_{21} \begin{bmatrix} \alpha \\ q \end{bmatrix} + A_{22} \begin{bmatrix} v_T \\ \theta \end{bmatrix}.$$

When the algebraic equations are used to eliminate α and q from the differential equations, the following equations for the phugoid variables are obtained:

$$\begin{bmatrix} \dot{v}_T \\ \dot{\theta} \end{bmatrix} = (A_{22} - A_{21}A_{11}^{-1}A_{12}) \begin{bmatrix} v_T \\ \theta \end{bmatrix}. \quad (4.2-13)$$

The coefficient matrix for this equation can be found by forming $E^{-1}A$ from (2.5-31), and then partitioning the result according to (4.2-5). The term $M_{\dot{\alpha}}$ is the coefficient of $\dot{\alpha}$ and does not need to be carried through the E -matrix inverse. We shall make the usual assumption, that $\gamma_e = 0$. This greatly simplifies the derivation, but as we shall see later, γ has a significant effect on the phugoid mode. When the submatrices in (4.2-13) are multiplied out, the result is

$$\begin{bmatrix} \dot{v}_T \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} (X_V + X_{T_V} \cos \alpha_e) - \frac{X_\alpha [M_q(Z_V - X_{T_V} \sin \alpha_e) - (V_T + Z_q)(M_V + M_{T_V})]}{\Delta_p} & -g \\ \frac{M_\alpha(Z_V - X_{T_V} \sin \alpha_e) - Z_\alpha(M_V + M_{T_V})}{\Delta_p} & 0 \end{bmatrix} \begin{bmatrix} v_T \\ \theta \end{bmatrix}, \quad (4.2-14)$$

where $\Delta_p = M_q Z_\alpha - M_\alpha(V_T + Z_q)$. The characteristic equation can now be found (from $|sI - A|$), and a comparison with the quadratic standard form

gives the following expressions for the phugoid natural frequency and damping:

$$\begin{aligned}\omega_{n_p}^2 &= g \frac{M_\alpha(Z_V - X_{T_V} \sin \alpha_e) - Z_\alpha(M_V + M_{T_V})}{M_q Z_\alpha - M_\alpha(V_T + Z_q)} \\ 2\zeta_p \omega_{n_p} &= -(X_V + X_{T_V} \cos \alpha_e) \\ &\quad + \frac{X_\alpha [M_q(Z_V - X_{T_V} \sin \alpha_e) - (V_T + Z_q)(M_V + M_{T_V})]}{M_q Z_\alpha - M_\alpha(V_T + Z_q)}.\end{aligned}\quad (4.2-15)$$

These expressions are considerably more complicated than those for the short-period mode; nevertheless, some conclusions can be drawn from them.

Consider the expression for the phugoid frequency, and for simplicity neglect the thrust derivatives. When the dimensionless derivatives are substituted into the natural frequency equation, we obtain

$$\frac{\omega_{n_p}^2}{g} = \frac{(C_D + C_{L_\alpha})(2C_M + C_{m_V}) - C_{m_\alpha}(2C_L + C_{L_V})}{-\frac{1}{2}\bar{c}C_{m_q}(C_D + C_{L_\alpha}) - C_{m_\alpha}[mV_T^2/(\bar{q}S) - \frac{1}{2}\bar{c}C_{L_q}]}.\quad (4.2-16)$$

Now assume that the engine thrust vector passes through the cg, so that the equilibrium aerodynamic pitching moment is zero ($C_M = 0$). Also, neglect the derivatives C_{m_V} , C_{L_V} , and C_{L_q} , and assume that $C_D \ll C_{L_\alpha}$. Equation (4.2-16) then reduces to

$$\frac{\omega_{n_p}^2}{g} = \frac{2C_{m_\alpha}C_L}{\frac{1}{2}\bar{c}C_{m_q}C_{L_\alpha} + 2mC_{m_\alpha}/(\rho S)}.\quad (4.2-17)$$

If the derivatives in (4.2-17) are independent of changes in the flight condition, the phugoid frequency is proportional to the square root of the equilibrium lift coefficient. Then since $C_L = W/(\bar{q}S)$ for level flight, we can say that *the phugoid period will tend to be proportional to the square root of the dynamic pressure*. The ρ term in the denominator introduces additional dependence on density, but *at constant altitude, we can expect the period to be proportional to airspeed*.

If the C_{m_α} term governs the denominator of (4.2-17), we see that

$$\frac{\omega_{n_p}^2}{g} \approx \frac{\rho S}{m} C_L.$$

Then, substituting once again for C_L , the phugoid frequency is found to be

independent of density:

$$\omega_{n_p} \approx \frac{g}{V_T} \sqrt{2}. \quad (4.2-18)$$

This result was given by F. W. Lanchester in 1908 and can be derived for large-amplitude motion from energy considerations. The numerical example in the next subsection will allow the reader to check the accuracy of the assumptions above, if desired. That example shows (see Table 4.2-1) that even without the assumptions on the thrust vector and variation of thrust, the phugoid period is closely proportional to airspeed (at constant altitude).

It is more difficult to derive simple expressions for the damping of the phugoid, and furthermore, in the next subsection the damping equation in (4.2-15) is shown to be quite inaccurate. The equation indicates that damping is dependent on the way the powerplant thrust varies with speed, and Roskam [1979] provides (approximate) comparative analyses for jets, propeller aircraft, rocket aircraft, and unpowered aircraft. In general, *the phugoid damping ratio increases with airspeed and decreases with altitude*. The numerical results in the next subsection, and in Example 4.2-2, show this effect.

Accuracy of the Short-Period and Phugoid Approximations

The short-period approximation is almost always a good approximation for the α and q response to elevator inputs with constant throttle input, and it will play an important role in the numerical designs in Chapter 4. The phugoid approximation usually gives good accuracy for the period of the phugoid oscillation but not for the damping ratio. These facts can be demonstrated using the transport aircraft model.

The dimensional-derivative evaluation program used in Example 3.6-2 was extended to calculate the short-period and phugoid properties from (4.2-7) and (4.2-14), respectively. Thus the characteristic roots (or the frequencies and damping ratios) could be calculated for the transport aircraft from any given set of steady-state flight conditions. The program also calculated the matrix $E^{-1}A$ (as used in Example 3.6-2) so that more accurate dynamic modes could be obtained from this matrix using an eigenvalue program. The flight conditions were level flight at sea level, with different airspeeds and cg positions. Table 4.2-1 shows the results of these calculations.

An asterisk in the table indicates characteristic roots instead of period and damping ratio. The first three sets of entries show the effect of varying airspeed; the last four sets show the effect of moving the cg position further aft, with speed held constant. The short-period approximation is seen to be a very good approximation for the first five cases. The phugoid approximation gives accurate results for the period; the damping ratio is quite inaccurate but the accuracy appears to improve when the period is large. Note that the phugoid mode is unstable at low airspeed (200 ft/s).

TABLE 4.2-1. Accuracy of Short-Period and Phugoid Formulae

Airspeed/cg	Calculation	T_{sp}	ζ_{sp}	T_p	ζ_p
200, 0.25	Approximate	7.44	0.555	32.3	0.102
	Exact	7.33	0.565	32.7	-0.129
400, 0.25	Approximate	3.73	0.551	63.5	0.064
	Exact	3.72	0.551	63.6	0.035
600, 0.25	Approximate	2.48	0.551	96.5	0.112
	Exact	2.48	0.551	96.6	0.099
400, 0.30	Approximate	4.04	0.598	65.4	0.067
	Exact	4.04	0.524	65.5	0.033
400, 0.40	Approximate	5.04	0.744	74.1	0.083
	Exact	5.02	0.652	74.3	0.036
400, 0.50	Approximate	-0.523, -1.33*		476	0.691
	Exact	-0.810 $\pm j0.200^*$		476	0.630
400, 0.55	Approximate	-1.70, -0.158*,		-0.158, 0.128*	
	Exact	-1.44, 0.100*,		-0.150 $\pm j0.123^*$	

When the cg is moved aft, the short-period roots move onto the real axis, and then one real root moves toward the phugoid roots. The short-period and phugoid approximations break down and one real root moves into the right-half plane. At the same time a new oscillatory mode appears that has a phugoid-like period with a short-period damping ratio. This mode is sometimes known as the *third oscillatory mode*, and it is characteristic of a statically unstable airplane (see Section 2.2). Also, the fact that one real root becomes unstable signals an exponential instability in pitch (a pitch "departure") rather than an oscillatory instability. This is the kind of instability that might be intuitively associated with the loss of positive pitch stiffness.

In the example, the stability boundary for the aft-cg location occurs when the cg lies between $0.501\bar{c}$ and $0.502\bar{c}$. It is evident that the characteristic equation of the short-period approximation cannot be used as an accurate means of calculating this cg position. However, the condition for a single real root to move into the right-half plane can be derived quite easily from the complete longitudinal dynamics, as we now show.

Pitch Stability

Section 2.2 described the concept of positive pitch stiffness (or static longitudinal stability), and pointed out that positive stiffness was neither necessary nor sufficient for stability of the longitudinal motion. The stability of the longitudinal motion will now be investigated by means of a dynamic analysis.

The characteristic polynomial of the decoupled longitudinal dynamics can be obtained from the determinant $|sE - A|$, with the E and A matrices as

given in (2.5-31). The constant term in the characteristic polynomial is equal to the product of the roots, and therefore the constant term will vanish when the real root reaches the origin, as the pitch-stability limit is reached. This constant term is obtained by putting $s = 0$ in $|sE - A|$, and therefore the required criterion is $|A| = 0$. If the determinant obtained from (2.5-31) is expanded about the (3, 4) element, with $\gamma_e = 0$, the result is

$$0 = |A| = \begin{vmatrix} X_V + X_{T_V} \cos \alpha_e & X_\alpha & -g \\ Z_V - X_{T_V} \sin \alpha_e & Z_\alpha & 0 \\ M_V + M_{T_V} & M_\alpha & 0 \end{vmatrix} = -g \begin{vmatrix} Z_V - X_{T_V} \sin \alpha_e & Z_\alpha \\ M_V + M_{T_V} & M_\alpha \end{vmatrix}$$

or

$$(Z_V - X_{T_V} \sin \alpha_e) M_\alpha = Z_\alpha (M_V + M_{T_V}).$$

When the dimensionless derivatives are substituted into this equation, the condition becomes

$$C_{m_\alpha}(2C_L + C_{L_V} + C_{T_V} \sin \alpha_e) = (C_D + C_{L_\alpha})(2C_M + C_{m_V} + C_{T_V} Z_E / \bar{c}), \quad (4.2-19)$$

where Z_E is the displacement used to calculate the engine thrust moment about the cg, and the thrust derivative C_{T_V} is given by

$$C_{T_V} = \frac{V_T}{\bar{q}S} \times \frac{\partial F_T}{\partial V_T}.$$

Equation (4.2-19) is quite complex and some special cases are worth considering. First, if the engine thrust is not significantly dependent on airspeed (a jet engine), then $C_{T_V} = 0$. Also, if the thrust acts through the cg ($Z_E = 0$), the equilibrium value of the aerodynamic pitching moment coefficient is zero ($C_M = 0$). Under these conditions (4.2-19) reduces to

$$C_{m_\alpha}(2C_L + C_{L_V}) = (C_D + C_{L_\alpha})C_{m_V}. \quad (4.2-20)$$

This equation shows that the stability boundary $C_{m_\alpha} = 0$, given by the positive pitch-stiffness condition, does not accurately define the true stability boundary when C_{m_V} is not zero. The derivative C_{m_V} was described in Section 2.5; it is associated with the "tuck" phenomenon and may become large in the transonic regime. Note that the conditions $C_D \ll C_{L_\alpha}$ and $C_{L_V} \ll C_L$ are normally satisfied, but (4.2-20) still involves the equilibrium value of the lift coefficient. Therefore, the stability condition is still dependent on the flight conditions even if the stability derivatives involved are relatively independent

of flight conditions. Finally, (4.2-19) is accurately satisfied for the flight conditions in Table 4.2-1 when the cg position is between $0.501\bar{c}$ and $0.502\bar{c}$, and a trial-and-error numerical determination of the critical cg position gives the same result. Therefore, (4.2-19) is an accurate stability criterion.

Algebraic Derivation of Lateral - Directional Transfer Functions

The procedure used to derive the longitudinal transfer functions will now be applied to the lateral-directional equations. The decoupled lateral-directional coefficient matrices are given by (2.5-33). We shall dispense with the E matrix by dividing the first lateral equation by V_T ; the characteristic equation is then

$$|sI - A| = \begin{vmatrix} s - \frac{Y_\beta}{V_T} & \frac{-(g'_0 \cos \theta_e)}{V_T} & \frac{-Y_p}{V_T} & \frac{-(Y_r - V_T)}{V_T} \\ 0 & s & \frac{-\cos \gamma_e}{\cos \theta_e} & \frac{-\sin \gamma_e}{\cos \theta_e} \\ -L'_\beta & 0 & s - L'_p & -L'_r \\ -N'_\beta & 0 & -N'_p & s - N'_r \end{vmatrix} \quad (4.2-21)$$

and the B matrix is

$$B = \begin{bmatrix} \frac{Y_{\delta a}}{V_T} & \frac{Y_{\delta r}}{V_T} \\ 0 & 0 \\ L'_{\delta a} & L'_{\delta r} \\ N'_{\delta a} & N'_{\delta r} \end{bmatrix}. \quad (4.2-22)$$

Cramer's rule can now be used to find any particular transfer function.

It is usual to make the lateral-directional equations manageable by assuming level flight ($\gamma_e = 0$). Then, for example, the aileron-to-roll-rate transfer function is

$$\frac{p(s)}{\delta_a(s)} = \frac{1}{|sI - A|} \begin{vmatrix} s - \frac{Y_\beta}{V_T} & \frac{-(g'_0 \cos \theta_e)}{V_T} & \frac{Y_{\delta a}}{V_T} & \frac{-(Y_r - V_T)}{V_T} \\ 0 & s & 0 & 0 \\ -L'_\beta & 0 & L'_{\delta a} & -L'_r \\ -N'_\beta & 0 & N'_{\delta a} & s - N'_r \end{vmatrix}, \quad (4.2-23)$$

which is of the form

$$\frac{p(s)}{\delta_a(s)} = \frac{ks(s^2 + 2\zeta_\phi\omega_\phi s + \omega_\phi^2)}{a_4s^4 + a_3s^3 + a_2s^2 + a_1s + a_0}. \quad (4.2-24)$$

The subscript ϕ has been used on the numerator quadratic because the same factor appears in the bank-angle transfer function, and the notation is in common use.

Once again, the polynomial coefficients are complicated functions of the dimensional derivatives, but some simplifications are possible. If the side-force and yawing effects of the ailerons are neglected (i.e., neglect $Y_{\delta a}/V_T$ and $N'_{\delta a}$), the determinant in (4.2-23) has a simple expansion about the third column. Then, assuming that $Y_r/V_T \ll 1$, the numerator of (4.2-24) can be written as

$$ks(s^2 + 2\zeta_\phi\omega_\phi s + \omega_\phi^2) = sL'_{\delta a} \left[s^2 - s \left(N'_r + \frac{Y_\beta}{V_T} \right) + \left(N'_\beta + \frac{Y_\beta N'_r}{V_T} \right) \right]. \quad (4.2-25)$$

When the aircraft has negligible *roll-yaw coupling*, the quadratic factor on the right-hand side of (4.2-25) also appears in the lateral-directional characteristic equation. This is shown in the next subsection. The resulting cancellation leaves a particularly simple expression for the aileron-to-roll-rate transfer function.

The lateral-directional characteristic equation does not separate into factors that clearly define each mode. Approximations will be derived that may describe an individual mode reasonably well, but they must be checked for applicability in any given case. Nevertheless, these approximations do provide useful insight into the dynamic behavior and will be derived for this reason. We start with the dutch roll approximation.

The Dutch Roll Approximation

The characteristic determinant, (4.2-21), has a simple reduction about the second column if the gravity term can be neglected. This term is the coefficient of ϕ in the differential equations, and its contribution to the dutch roll mode must be considered. The dihedral derivative, C_{l_B} (see Section 2.5), determines the amount of rolling in the dutch roll mode, and when this derivative is small, the mode will consist mainly of sideslipping and yawing. The dihedral derivative tends to be large in modern swept-wing aircraft and so, as already indicated, we shall use the dutch roll approximation only to gain some insight into the mode.

With the gravity term omitted, the reduction of the subsequent third-order determinant can be further simplified because the terms Y_p/V_T and Y_r/V_T

can be dropped (Y_p is often zero, and $Y_r/V_T \ll 1$). The cross-derivative term N'_β (yawing acceleration due to roll rate) is also often negligible. These assumptions lead to the common "three-degrees-of-freedom dutch roll approximation." The characteristic polynomial is now obtained very easily from the determinant:

$$|sI - A| = s(s - L'_p) \left[s^2 - s \left(N'_r + \frac{Y_\beta}{V_T} \right) + \left(N'_\beta + \frac{Y_\beta N'_r}{V_T} \right) \right]. \quad (4.2-26)$$

This polynomial has a root at the origin that approximates the spiral pole, and a root at L'_p approximating the roll subsidence pole. The quadratic roots correspond to the dutch roll poles.

The dutch roll quadratic in (4.2-26) exactly cancels the quadratic factor in (4.2-25); therefore, an approximation for the aileron-to-roll-rate transfer function (4.2-24) is given by

$$\frac{p(s)}{\delta_a(s)} = \frac{L'_{\delta a}}{s - L'_p}. \quad (4.2-27)$$

Equation (4.2-26) shows that the approximations to the frequency and damping of the dutch roll mode are given by

$$\begin{aligned} \omega_{n_d}^2 &= N'_\beta + \frac{N'_r Y_\beta}{V_T} \\ \zeta_d &= \frac{-(N'_r + Y_\beta/V_T)}{2\omega_{n_d}}. \end{aligned} \quad (4.2-28)$$

We shall now substitute dimensionless stability derivatives into these formulae and examine the dependence on flight conditions.

The derivative N'_β usually dominates the expression for the natural frequency. Therefore, the dutch roll frequency is determined by a combination of the yaw-stiffness and dihedral derivatives (see Section 2.5), according to

$$N'_\beta = \frac{N_\beta + (J'_{XZ}/J'_X)L'_\beta}{1 - J'^2_{XZ}/(J'_X J'_Z)}. \quad (4.2-29)$$

The stability-axes cross-product of inertia, J'_{XZ} , varies rapidly with the equilibrium angle of attack, typically changing from a small positive value at low alpha to a much larger negative value at high alpha. This larger value is still relatively small compared to J'_X and J'_Z , so the primed derivatives are normally quite close to their unprimed values. However, it is possible for N_β to decrease and even change sign under certain conditions (see Section 2.5), and then the effective directional stiffness may be influenced by L'_β .

If we simply use the unprimed derivative N_β in the formula for the dutch roll frequency and substitute the dimensionless derivative, we obtain

$$\omega_{n_d}^2 = \frac{\bar{q}Sb}{J'_Z} C_{n_\beta}. \quad (4.2-30)$$

This equation shows that the dutch roll frequency is proportional to the square root of dynamic pressure, assuming constant C_{n_β} . Therefore, at constant altitude, the *frequency increases in proportion to the airspeed*, and for a given speed the *frequency decreases with altitude*.

In the damping formula of (4.2-28) the two numerator terms are typically of equal importance, and neither can be neglected. Once again unprimed derivatives will be substituted for the primed derivatives, with the same caveats as above. When the dimensionless derivatives are substituted, and the natural frequency expression from (4.2-28) is used, the damping ratio is given by

$$\zeta_d = \frac{-1}{8} \left[\frac{2\rho Sb^3}{J'_Z} \right]^{1/2} \frac{C_{n_r} + 2J'_Z C_{Y_\beta}/(mb^2)}{\left[C_{n_\beta} + b\rho SC_{n_r} C_{Y_\beta}/(4m) \right]^{1/2}}. \quad (4.2-31)$$

This expression indicates that the dutch roll damping is independent of dynamic pressure. *It will decrease with altitude* unless the denominator is dominated by the second term, in which case density will cancel out of the formula.

The dutch roll natural frequency formula tends to be quite accurate, although the damping formula is not. This will be illustrated later, by example. Finally, note that the approximation to the roll subsidence pole, $s = L'_p$, is not very accurate and a more accurate approximation will be derived next.

The Spiral and Roll Subsidence Approximations

The rolling and spiral modes usually involve very little sideslip. The rolling mode is almost pure rolling motion around the x -stability axis, and the spiral mode consists of yawing motion with some roll. It is common for the spiral mode to be unstable and the motion then consists of increasing yaw and bank angle, in a tightening downward spiral.

These facts allow approximations to be devised by modifying the force equation (the $\dot{\beta}$ equation) and leaving the moment equations unchanged. Sideforce due to sideslip is eliminated from the force equation, $\dot{\beta}$ is neglected, and the gravity force is balanced against the force component associated with yaw rate. Thus, in the characteristic determinant, (4.2-21), the term $(s - Y_\beta/V_T)$ is eliminated, and the Y_p/V_T term is again neglected. Because the gravity force is intimately involved in the spiral mode, the mode is dependent on flight-path angle. Unfortunately, the assumption of level flight is needed to allow a reasonably simple analysis and will therefore be

used here. The effect of flight-path angle will be investigated numerically in Example 4.2-2. When the simplified determinant is expanded, the following second-order characteristic equation is obtained:

$$N'_\beta s^2 + \left(L'_\beta N'_p - L'_p N'_\beta - \frac{L'_\beta g}{V_T} \right) s + \frac{(L'_\beta N'_r - N'_\beta L'_r)g}{V_T} = 0. \quad (4.2-32)$$

Equation (4.2-32) normally has two real roots, corresponding to the roll subsidence pole and the spiral pole. Also, the spiral time constant is normally very much greater than the roll time constant. Under these circumstances, if we divide through (4.2-32) by the coefficient of s^2 , the coefficient of s (i.e., the negative of the sum of the roots) yields the (negative of the) roll subsidence root. The constant term (i.e., the product of the roots) can then be used to obtain the spiral root. Therefore, we have the further approximations

Roll Time Constant (τ_R):

$$\frac{1}{\tau_R} \approx -L'_p \left(1 - \frac{L'_\beta N'_p}{N'_\beta L'_p} \right) - \frac{L'_\beta g}{N'_\beta V_T} \quad (4.2-33)$$

Spiral Time Constant (τ_S):

$$\tau_S = \frac{L'_\beta (N'_p - g/V_T) - L'_p N'_\beta}{(L'_\beta N'_r - N'_\beta L'_r)g/V_T}. \quad (4.2-34)$$

Note that a negative value for the time constant will simply mean an unstable exponential mode.

If we once again neglect the primes and substitute dimensionless derivatives in the roll time-constant equation, we obtain

$$\frac{1}{\tau_R} = -\frac{\bar{q}Sb^2}{2V_T J'_X} C_{l_p} \left(1 - \frac{C_{l_\beta} C_{n_p}}{C_{n_\beta} C_{l_p}} \right) - \frac{J'_Z C_{l_\beta} g}{J'_X C_{n_\beta} V_T}. \quad (4.2-35)$$

This equation shows that within the limits of the stated approximations, and assuming constant stability derivatives, the *roll time constant varies inversely as the product of density and speed (\bar{q}/V_T)*. The last term in (4.2-35) usually adds a small positive correction, which tends to offset the effect above by giving the roll time constant a component proportional to speed.

The same procedure can be followed with the spiral time-constant equation, (4.2-34). The result is

$$\tau_S = \frac{V_T}{g} \frac{C_{l_\beta} C_{n_p} - C_{n_\beta} C_{l_p} - 2gJ'_Z C_{l_\beta}/(b^2 \bar{q}S)}{C_{l_\beta} C_{n_r} - C_{n_\beta} C_{l_r}}. \quad (4.2-36)$$

The middle term in the numerator is usually dominant so that, within the limits of this approximation, the spiral time constant is proportional to speed. The last term of the numerator will become more important at low dynamic pressure and will modify this dependence.

Spiral Stability

The condition for a real pole at the origin is given by $|A| = 0$, and in the case of the lateral dynamics this normally represents the spiral pole becoming neutrally stable. Using the characteristic equation (4.2-21), we obtain

$$|A| = \begin{vmatrix} \frac{Y_\beta}{V_T} & \frac{g'_0 \cos \theta_e}{V_T} & \frac{Y_p}{V_T} & \frac{Y_r - V_T}{V_T} \\ 0 & 0 & \frac{\cos \gamma_e}{\cos \theta_e} & \frac{\sin \gamma_e}{\cos \theta_e} \\ L'_\beta & 0 & L'_p & L'_r \\ N'_\beta & 0 & N'_p & N'_r \end{vmatrix}.$$

When this is expanded the spiral stability boundary is obtained as

$$(L'_\beta N'_r - N'_\beta L'_r) \cos \gamma_e + (L'_p N'_\beta - L'_\beta N'_p) \sin \gamma_e = 0. \quad (4.2-37)$$

This equation shows that spiral stability is dependent on flight-path angle, as noted earlier.

Accuracy of the Lateral Mode Approximations

The approximations above are usually quite accurate, apart from the dutch roll damping. The spiral time constant is also accurately predicted when this mode is unstable. This accuracy will be demonstrated in the following example using a model of a business jet in a cruising flight condition.

Example 4.2-1: Lateral Modes of a Business Jet. The following lateral-directional data for a business jet is taken from Roskam [1979].

Flight Condition:

$$W = 13,000 \text{ lb}, \quad h = 40,000 \text{ ft} \quad (\rho = 0.000588 \text{ slug}/\text{ft}^3)$$

$$V_T = 675 \text{ ft/s}, \quad \gamma = 0 \text{ deg}, \quad \alpha = 2.7 \text{ deg}$$

$$J_x = 28,000, \quad J_z = 47,000 \quad J_{xz} = 1350 \text{ slug}\cdot\text{ft}^2 \text{ (body axes)}$$

Geometrical Data: $S = 232 \text{ ft}^2$, $b = 34.2 \text{ ft}$

Stability Derivatives:

$$\begin{array}{lll} C_{y_\beta} = -0.730, & C_{y_p} = 0, & C_{y_r} = +0.400 \\ C_{l_\beta} = -0.110, & C_{l_p} = -0.453, & C_{l_r} = +0.163 \\ C_{n_\beta} = +0.127, & C_{n_p} = +0.008, & C_{n_r} = -0.201 \end{array}$$

A short program was written to convert the moments of inertia to stability axes, calculate the elements of the decoupled A matrix, and evaluate the approximate equations for the modal characteristics [from (4.2-28) and (4.2-32), (4.2-33), (4.2-34)]. Some intermediate results are:

Stability-Axes Moments of Inertia:

$$J'_X = 27,915, \quad J'_Z = 47,085, \quad J'_{XZ} = 450.0$$

Dimensional Derivatives:

$$\begin{array}{lll} Y_\beta = -56.14, & Y_p = 0, & Y_r = 0.7793 \\ L_\beta = -4.188, & L_p = -0.4369, & L_r = 0.1572 \\ N_\beta = 2.867, & N_p = 0.004575, & N_r = -0.1149 \end{array}$$

Primed Dimensional Derivatives:

$$\begin{array}{lll} L'_\beta = -4.143, & L'_p = -0.4369, & L'_r = 0.1554 \\ N'_\beta = 2.800, & N'_p = -0.002469, & N'_r = -0.1124 \end{array}$$

The A matrix was written to a file so that an eigenvalue program could be used to determine the modes exactly. The exact and approximate results are as follows:

Dutch Roll Mode:

$$\begin{aligned} \text{exact: } \omega_{n_d} &= 1.682 \text{ rad/s}, \quad \zeta_d = 0.0373 \\ \text{Eqs. (4.2-28): } \omega_{n_d} &= 1.676 \text{ rad/s}, \quad \zeta_d = 0.0584 \end{aligned}$$

Roll Subsidence Mode:

$$\begin{aligned} \text{exact: } \tau_R &= 1.976 \text{ s} \\ \text{Eq. (4.2-32): } \tau_R &= 1.960 \text{ s} \\ \text{Eq. (4.2-33): } \tau_R &= 1.957 \text{ s} \end{aligned}$$

Spiral Mode:

$$\text{exact: } \tau_s = 978.5 \text{ s}$$

$$\text{Eq. (4.2-32): } \tau_s = 976.7 \text{ s}$$

$$\text{Eq. (4.2-34): } \tau_s = 978.7 \text{ s}$$

These results are in remarkably good agreement, apart from the dutch roll damping.

Envelope Mode Variation from the Nonlinear Model

It is not entirely realistic to use a fixed set of stability derivatives to show the variation of the modal characteristics with flight conditions. Therefore, as a final example we shall use the completely numerical approach to calculate the modes of the nonlinear F-16 model at different flight conditions. The modes will only be calculated accurately since the numerical linearization is set up to produce the state-equation coefficient matrices, not the stability derivatives. The variation of the modes with flight-path angle will also be determined since this could not easily be done with the approximate formulae.

Example 4.2-2: Mode Dependence from the Nonlinear Model. The nonlinear F-16 model allows a realistic examination of the dependence of the modes on flight conditions, since it is not built from a fixed set of aerodynamic derivatives. The following results were obtained by trimming and numerically linearizing the model at the desired flight condition, and then using an eigenvalue program to determine the modes from the full 13-state A matrix. Virtually identical results could be obtained by using the decoupled lat-long matrices.

The effect of flight-path angle was investigated by trimming the model according to the second set of conditions in Table 3.4-3 (502 ft/s, $h = 0$ ft, $cg = 0.3\bar{c}$) but with different values of γ . The modes are shown in Table 4.2-2. It is evident from these results that the "rotational" modes are almost independent of γ . Overall, the properties of the rotational modes are remarkably consistent, considering the nature of the tabular aerodynamic data and the numerical processing (trimming and linearization) required to obtain them. The "trajectory" modes, phugoid and spiral, are strongly influenced by the flight-path angle. The spiral time constant initially increases as the flight-path angle increases, becomes infinite as the stability boundary is approached, and then decreases with flight-path angle when the mode is unstable. The phugoid period is only weakly affected by γ but increases as γ increases. Phugoid damping is more strongly affected, it decreases with increasing γ , and the phugoid becomes unstable at a quite modest flight-path angle.

TABLE 4.2-2. Effect of Flight-Path Angle on the F-16 Modes

γ	-5	0	5	10	15	20	deg
T_d	1.934	1.933	1.934	1.937	1.941	1.946	s
ζ_d	0.1346	0.1353	0.1360	0.1366	0.1371	0.1375	
τ_s	55.33	77.91	133.0	461.9	-312.3	-117.0	s
τ_R	0.2777	0.2777	0.2775	0.2772	0.2766	0.2760	s
T_{sp}	3.281	3.277	3.273	3.269	3.266	3.262	s
ζ_{sp}	0.6277	0.6279	0.6281	0.6282	0.6283	0.6283	
T_p	79.60	80.05	80.93	82.39	84.36	86.82	s
ζ_p	0.1297	0.09751	0.06557	0.03396	0.00227	-0.0298	

In Table 4.2-3 the model is trimmed in level flight with various combinations of speed and altitude to illustrate the effect of these two variables on the modes. The cg position is again at $0.3\bar{c}$. The flight conditions have been chosen to compare different speeds at the same altitude, the same speed at different altitudes, high and low dynamic pressures at the same altitude, and the same dynamic pressure at two greatly different altitudes. The first trim condition (50,000 ft, 900 ft/s) corresponds to 0.93 Mach and is therefore strictly outside the valid Mach range of the model; this is also true to a lesser extent for the third case (0.81 Mach). We do not have a model that includes compressibility effects, and we shall simply consider this example as illustrating the variation of the modes when compressibility is not important. The second trim condition (50,000 ft, 600 ft/s) corresponds to full throttle, while the first case (higher speed) corresponds to only 0.765 throttle. Therefore, a dive and climb maneuver would be needed to get from the second to the first flight condition. The longitudinal dynamics are unstable in the second case. In the fourth flight condition trial-and-error adjustment of the speed was used to make the dynamic pressure the same as case 1.

TABLE 4.2-3. Effect of Speed and Altitude on the F-16 Modes

Alt./speed (dyn. pres.)	50,000, 900 (160)	50,000, 600 (71)	0, 900 (963)	0, 367 (160)	ft, ft/s lb/ft ²
T_d	2.365	2.735	1.143	2.396	s
ζ_d	0.06480	0.07722	0.1272	0.1470	
τ_s	179.2	138.7	122.1	73.52	s
τ_R	1.050	2.230	0.1487	0.4160	s
T_{sp}	4.507	u/s	2.372	4.023	s
ζ_{sp}	0.2615	u/s	0.8175	0.5735	
T_p	102.1	u/s	183.4	56.93	s
ζ_p	0.005453	u/s	0.3242	0.06240	

The tabulated results show that, as expected, the dutch roll has almost the same period at two widely different speed/altitude combinations with the same dynamic pressure. They also show the expected increase in period with altitude (at constant speed), and the decrease in period with airspeed (at constant altitude). The dutch roll damping does tend to be independent of dynamic pressure and to decrease with altitude, as predicted by the theory.

The spiral time constant is expected to vary directly with V_T if the third numerator term in (4.2-36) is negligible, and to vary as V_T/\bar{q} if that term is dominant. The results indicate that the actual variation is somewhere in between these two trends. This is not unexpected because the F-16 has swept wings and C_{I_β} can be expected to play a significant part in (4.2-36).

The time constant of the roll subsidence mode is approximately proportional to V_T/\bar{q} as predicted. The short-period mode also shows the expected trends, the period is roughly the same at the two equal dynamic pressure conditions, and is much smaller at the high dynamic pressure condition. At 50,000 ft the atmospheric density is one sixth of sea-level density and, as predicted, ζ_{sp} changes by about $\sqrt{6}$. The sixfold increase in \bar{q} at sea level has a much smaller effect on ζ_{sp} .

In the case of the phugoid period the two sea-level results show that the sixfold increase in dynamic pressure causes an increase in the period of 3.2 times (compared to the prediction of $\sqrt{6}$). At constant dynamic pressure the period increases with altitude, as expected. The phugoid damping also shows the expected trend, increasing with airspeed, and decreasing with altitude. ■

4.3 THE FLYING-QUALITIES REQUIREMENTS

Control-law design can only be performed satisfactorily if a set of design requirements or performance criteria is available. In the case of control systems for piloted aircraft generally applicable quantitative design criteria are very difficult to obtain. The reason for this is that the ultimate evaluation of a human-operator control system is necessarily subjective and, with aircraft, the pilot evaluates the aircraft in different ways depending on the type of aircraft and phase of flight. For example, in a dynamic maneuvering situation the pilot may be concerned mainly with the control forces that must be exerted and the resulting six-degrees-of-freedom translational and angular accelerations. In a task requiring precision tracking the pilot's evaluation will be more influenced by visual cues and the response of the aircraft to turbulence.

Also, a pilot's opinion of the *flying qualities* of an aircraft is inevitably influenced by factors other than the obvious control-system considerations of response to control inputs and response to disturbance inputs (e.g., turbulence). He or she will be influenced by the ergonomic design of the cockpit controls, the visibility from the cockpit, the weather conditions, the mission requirements, and physical and emotional factors. The variability introduced

by all these factors can only be reduced by averaging test results over many flights and many pilots.

A systematic approach to flying-qualities evaluation is available through *pilot opinion rating* scales such as the Cooper-Harper scale [Cooper and Harper, 1969]. This rating scale is shown in Table 4.3-1. Once a rating scale like this has been established it is possible to begin correlating the pilot

TABLE 4.3-1. Pilot Opinion Rating and Flying-Qualities Level

Aircraft Characteristics	Demands on Pilot in Selected Task or Required Operation	Pilot Rating	Flying-Qualities Level
Excellent; highly desirable	Pilot compensation not a factor for desired performance	1	
Good; negligible deficiencies	Pilot compensation not a factor for desired performance	2	1
Fair; some mildly unpleasant deficiencies	Minimal pilot compensation required for desired performance	3	
Minor but annoying deficiencies	Desired performance requires moderate pilot compensation	4	
Moderately objectionable deficiencies	Adequate performance requires considerable pilot compensation	5	2
Very objectionable but tolerable deficiencies	Adequate performance requires extensive pilot compensation	6	
Major deficiencies	Adequate performance not attainable with maximum tolerable pilot compensation Controllability not in question	7	
Major deficiencies	Considerable pilot compensation required for control	8	3
Major deficiencies	Intense pilot compensation required to retain control	9	
Major deficiencies	Control will be lost during some portion of required operation	10	

opinion rating with the properties of the aircraft dynamic model, and hence derive some analytical specifications that will guarantee good flying qualities. Although this may seem simple in principle, it has proven remarkably difficult to achieve in practice, and after many years of flying-qualities research it is still not possible to precisely specify design criteria for control systems intended to modify the aircraft dynamics. A survey and a large bibliography covering 25 years of flying-qualities research have been given by Ashkenas (1984). The "background information and user guides" for the military flying-qualities specifications MIL-F-8785B and MIL-F-8785C [Chalk et al., 1969; Moorhouse and Woodcock, 1982] also provide much useful information.

We first consider some possible ways in which requirements on dynamic response may be specified. The aircraft model may be linearized in a particular flight condition and the poles and zeros, or frequency response, of a particular transfer function compared with a specification. Alternatively, certain time responses may be derived from the nonlinear model, in a particular flight condition, and compared with specifications. Yet another alternative is to model the human operator as an element in a closed control loop containing the aircraft dynamics, and determine what requirements are placed on the operator if the closed-loop control is to have a satisfactory command or disturbance response. All of these techniques have been, or are being, considered by workers in the field, and we shall examine some of the ideas in more detail.

Pole-Zero Specifications

Suppose that lat-long decoupling is assumed and the pitch axis is considered. In addition, assume linear dynamic behavior. Then, if a transfer function shows that the dynamic response is dominated by a single pair of complex poles (e.g., the short-period poles), the pilot's opinion of the aircraft flying qualities should correlate with the position of these poles. A number of studies have provided data to link pole positions to pilot opinion rating.

In one of the early studies, O'Hara [1967] produced iso-opinion contours for the location of the short-period poles; these were plotted on axes of undamped natural frequency versus damping ratio. They showed that the most satisfactory pilot-opinion rating corresponded to poles inside a closed contour bounded by about 2.4 and 3.8 rad/s, and by damping ratios of about 0.4 and 1.0, with its center at about 3.0 rad/s and $\zeta = 0.65$. This and other similar results form the basis of current pole-position flying-qualities criteria.

Unfortunately for the pole-position criterion, even if the decoupling and linearity assumptions are justified, there are at least two reasons why this approach may not work well. The first is that transfer function zeros are also important (we shall see later that they have a strong effect on step response). Second, the aircraft and control system dynamics may include quite a lot of poles that contribute significantly to the time response. Pilots are very

sensitive to additional dynamics, and the difficulties of specifying requirements on more than just a single pair of poles quickly become prohibitive. The problem of transfer function zeros will be considered first.

The short-period elevator-to-pitch-rate transfer function (4.2-9) plays an important role in the pilot's assessment of the longitudinal-axis flying qualities. In the numerator of this transfer function the Z_{δ_e} terms can usually be neglected, with the following result:

$$\frac{q}{\delta_e} = \frac{Z_\alpha M_{\delta_e} (sV_T/Z_\alpha - 1)}{V_T s^2 - (Z_\alpha + V_T M_q + V_T M_{\dot{\alpha}})s + M_q Z_\alpha - V_T M_\alpha}. \quad (4.3-1)$$

In the flying-qualities literature the dimensional derivative L_α ($\equiv \partial L / \partial \alpha \approx -mZ_\alpha$, $C_D \ll C_{L_\alpha}$) is often used instead of Z_α ; and the time constant associated with the transfer function zero is given the symbol T_{θ_2} (T_{θ_1} is associated with the phugoid mode). Therefore, we see that

$$T_{\theta_2} = \frac{-V_T}{Z_\alpha} \approx \frac{mV_T}{L_\alpha}. \quad (4.3-2)$$

This time constant is also often expressed in terms of the aircraft load factor response to angle of attack, n_α . Aircraft *load factor*, n , is defined as lift (L) divided by the weight (W), and n_α is the gradient of this quantity with respect to alpha [$n_\alpha = (\partial L / \partial \alpha) / W$]. Therefore, we have

$$T_{\theta_2} = \frac{V_T}{gn_\alpha}. \quad (4.3-3)$$

The position of the pitch-rate transfer function zero has been shown to correlate with pilot-opinion ratings of the flying qualities [Chalk, 1963]. Shomber and Gertsen [1967] derived iso-opinion curves involving the short-period frequency and damping, T_{θ_2} , and n_α . When n_α was less than 15 g/rad they found that pilot opinion correlated well with $1/(\omega_n T_{\theta_2})$ and ζ , with the optimum conditions being around $1/(\omega_n T_{\theta_2}) = 0.45$, $\zeta = 0.7$. When n_α was greater than 15, they found that the correlation was with n_α/ω_n (i.e., T_{θ_2} no longer fixed) and ζ , with the optimum conditions near $n_\alpha/\omega_n = 10$, $\zeta = 0.7$. The current military flying-qualities requirements (see later) specify the short-period natural frequency in terms of n_α , and there is still a division of opinion over the importance of T_{θ_2} versus n_α .

The lateral-directional dynamics have proved somewhat less critical than the longitudinal dynamics from the point of view of flying qualities. The normally required changes in the aircraft trajectory can be achieved by a combination of rolling and pitching. O'Hara [1967] used iso-opinion curves to show that lateral dynamics would receive a good rating if the maximum roll acceleration was appropriate to the roll time constant. Both of these quantities are transfer function parameters. Regardless of these studies the current

military requirements provide only specifications for the roll time constant and the time to reach a given bank angle. The latter quantity must be obtained from a nonlinear simulation.

The dutch roll mode is an unwanted complication in this simple picture; it should be fast and adequately damped (see later) so that the airplane will quickly reorient itself after a directional disturbance. Ideally, the dutch roll should have very little involvement in the lateral dynamics and should therefore almost cancel out of the lateral transfer functions. This requires that quantities ω_ϕ and ζ_ϕ for the complex zeros [see (4.2-24)] should coincide with ω_d and ζ_d for the dutch roll poles. The ratio ω_ϕ/ω_d is the most important quantity in this respect, and iso-opinion curves of ω_ϕ/ω_d versus ζ_d have been plotted [Ashkenas, 1966].

As might be expected, the optimum value of ω_ϕ/ω_d is close to unity for a stable dutch roll. However, there is a subtlety in these results; it can be shown that favorable yaw is generated in a turn when $\omega_\phi/\omega_d > 1$, and the converse is true. We shall refer to this again in connection with the lateral directional control augmentation system in Section 4.5.

Finally, consider the case of highly augmented aircraft, where the control systems contribute a number of poles and zeros in addition to those associated with the basic rigid-body transfer functions. Specifications placed on poles and zeros quickly become unmanageable and, as in the case of control system design, one must resort to frequency-response techniques. One way in which frequency-domain ideas have been applied to flying qualities specifications is described in the next subsection.

Frequency-Response Specifications

In general, the goal of an aircraft control system design should be to produce dominant closed-loop poles that resemble the basic rigid-body poles, with satisfactory damping and natural frequency (see later). The effect of the additional dynamics resulting from the control system compensation networks, actuators, and structural filters can be allowed for by determining an "equivalent low-order system" [Craig, 1971; Hodgkinson, 1979; Bischoff, 1981; Gentry, 1982].

In this concept the coefficients are determined for a low-order transfer function that matches the frequency response of the actual transfer function, over a limited frequency range. The gain and phase are matched simultaneously by adjusting the coefficients of the low-order transfer function to minimize a cost function of the form

$$\text{cost} = \frac{20}{n} \sum_{i=1}^n \left[\Delta G(\omega_i)^2 + \frac{\Delta P(\omega_i)^2}{57.3} \right]. \quad (4.3-4)$$

Here n is the number of discrete frequencies (ω_i) used, $\Delta G(\omega_i)$ is the difference in gain (in decibels) between the transfer functions at the fre-

quency ω_i , and $\Delta P(\omega_i)$ is the difference in phase (in degrees) at ω_i . The frequency range used is nominally 0.3 to 10 rad/s, and 20 to 30 discrete frequencies are needed. The upper frequency limit is based on the maximum control frequencies that pilots have been observed to use. The lower limit is based on observations that pilots do not provide continuous closed-loop control at very low frequencies, and the value given does not provide for matching the phugoid mode. The cost function is minimized with a multivariable search routine, in the same way that we obtained steady-state trim in Chapter 3.

The stick-force-to-pitch-rate transfer function is typically used to evaluate the longitudinal dynamics. To compare a particular aircraft with both the short-period and phugoid specifications in the military flying-qualities specifications, the assumed form of this transfer function is

$$\frac{q}{F_s} = \frac{K(s + 1/T_{\theta_1})(s + 1/T_{\theta_2})e^{-\tau s}}{(s^2 + 2\zeta_p \omega_p s + \omega_p^2)(s^2 + 2\zeta_{sp} \omega_{sp} s + \omega_{sp}^2)} \quad \frac{\text{rad/s}}{\text{lb}}. \quad (4.3-5)$$

Here the subscripts p and sp indicate, respectively, the phugoid and short-period modes. The frequency range for matching the transfer functions should be extended down to about 0.01 rad/s when the phugoid is included.

The term $e^{-\tau s}$ is included in the low-order model to provide an equivalent time delay for matching high-frequency effects from, for example, actuator modes, structural modes and mode filters, and noise filters. The time-delay term contributes only a phase shift to the transfer function; this is consistent with the fact that the phase variations from high-frequency dynamics extend over a larger frequency range than gain variations. The current military requirements suggest that for level 1 flying qualities, the maximum allowable value of the equivalent time delay should be 10.0 ms.

If a good fit to an equivalent low-order system is obtained (e.g., a cost of 10.0 or less), the pole-zero criteria can be applied to this equivalent system. If low values of the cost function cannot be obtained, other criteria must be used.

Another example of frequency-domain specifications applied to aircraft control systems (but not directly to flying qualities) is the military standard requirement document for the design, installation, and test of flight control systems (MIL-F-9490, 1975). This provides stability criteria by specifying the minimum gain and phase margins that must be achieved in any actuator path, with all other feedback paths closed. Typical values are 6 dB gain margin and 30° phase margin.

Time-Response Specifications

Placing flying-qualities requirements on the time response has the advantage that a time response can readily be obtained from the full nonlinear model dynamics. It does, however, raise the problems of what type of test input to

apply and which output variables to observe. In the case of the longitudinal dynamics, it is natural once again to specify requirements on the pitch-rate response. However, fighter aircraft control systems are normally designed to give the pilot control over pitch rate at low speed and normal acceleration (acceleration measured along the body z-axis) at high speed. This gives direct control over the variable that stresses the pilot. The two control schemes must be smoothly blended together (see, e.g., Toles, 1985).

Efforts to develop time-response criteria have mostly been linked to the decoupled longitudinal dynamics and have made use of the short-period approximation. They have attempted to define an envelope inside which the pitch rate, angle of attack, or normal acceleration response to an elevator step input should lie. As early as 1963 a step-response envelope for angle of attack was derived from the short-period iso-opinion curves [Rynaski, 1985]. Envelope criteria have been published for the pitch-rate response of an SST and of the space shuttle (see [Rynaski, 1985]).

A time-history envelope criterion called $C^*(t)$ ("C-star") was published in 1965 [Tobie et al., 1966] and is still in use. The C^* criterion uses a linear combination of pitch rate and normal acceleration at the pilot's station:

$$C^*(t) = n_{z_p} + 12.4q, \quad (4.3-6)$$

where n_{z_p} is the incremental acceleration in g's (zero g's = steady-state flight) and q is the pitch rate in radians per second. The envelope for the C^* criterion is shown in Fig. 4.3-1. A $C^*(t)$ elevator step response that falls near the center of the envelope should lead to level 1 longitudinal-axis flying qualities. A more recent time-domain criterion than C^* relates pilot opinion ratings to target tracking error and time on target for a step target tracking task [Onstott and Faulkner, 1978].

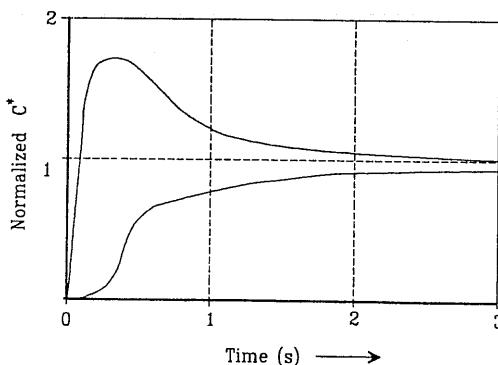


Fig. 4.3-1 C-star envelope.

The cited envelope criteria often give conflicting results and may be in disagreement with the pilot ratings for specific aircraft. Pitch-rate responses having large overshoots and poor settling times have often corresponded to good pilot-opinion ratings. It is known that for fighter aircraft air-combat modes a pitch-rate overshoot is required for good gross acquisition of targets, and a deadbeat pitch-rate response is required for good fine tracking. Rynaski (1985) has argued that angle of attack should be the basic response variable, and it appears that the angle-of-attack response corresponding to good flying qualities may be more like a good conventional step response (i.e., small overshoot and fast nonoscillatory settling).

A time-response criterion, called D^* (or *coordination perception parameter*), has been devised for the lateral-directional response [Kisslinger and Wendle, 1971]. The idea is similar to C^* in that the coordination perception parameter is a blend of lateral acceleration and sideslip angle, and envelope limits for acceptable performance are specified.

Requirements Based on Human Operator Models

For certain types of control tasks it is possible to model a human operator with linear differential equations or a transfer function. An example of such a task is a compensatory tracking task with a random input, that is, a control task in which the operator uses only tracking *error* information to track an unpredictable target. This information may be presented by instruments such as a pilot's artificial horizon display. The human operator model consists of the transfer function and an added nonanalytic output signal called the *remnant*. The purpose of the remnant is to account for the discrepancies between experimental results with a human operator and analyses using the model. The transfer function model is often given the name *human operator describing function* (not to be confused with the describing function of nonlinear control theory).

The human operator transfer function model for the compensatory tracking task is usually assumed to be

$$Y(s) = \frac{K_p e^{-ds} (\tau_l s + 1)}{(\tau_i s + 1)(\tau_n s + 1)}. \quad (4.3-7)$$

In this transfer function the pure delay, d , may be taken to represent the motor-control functions in the cerebellum and the neuromuscular delay, while the lag, τ_n , models the mechanical properties of the muscles and limbs. It is known that the speed of response is severely limited by the delay term rather than the lag, and the latter is neglected in many applications. The gain K_p , lead time constant τ_l , and lag time constant τ_i represent the capability of the human operator to optimize his or her control of a given task. Thus the

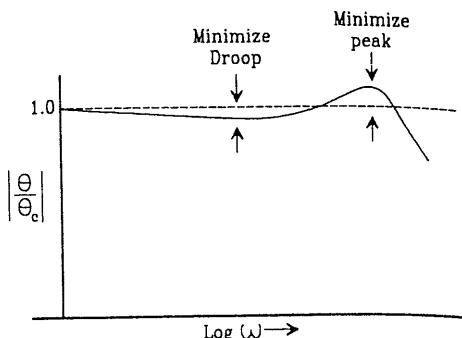


Fig. 4.3-2 Closed-loop frequency response for the Neal-Smith criterion.

operator may use lag compensation to achieve high gain and fine control in some low-bandwidth tasks, or lead compensation to achieve high bandwidth in others.

This model has been applied to aircraft piloting tasks, and hypotheses (the *adjustment rules*) have been developed for the way in which the adaptive parameters will be “chosen” by the pilot [McRuer et al., 1965]. It is also used as the basis of a transfer function method of assessing flying qualities [Neal and Smith, 1970]. Interesting examples of the transfer function model applied to a pilot controlling bank angle are given in Etkin [1972] and Roskam [1979].

In the Neal-Smith method the model (4.3-7) is used in conjunction with the aircraft stick-force-to-pitch-attitude transfer function, in a closed pitch-attitude control loop. It is assumed that the human pilot adjusts the lead, lag, and gain so that the *droop* and *peak magnification* of the closed-loop frequency response are minimized, as shown in Fig. 4.3-2. Therefore, this process is duplicated with the models, the lag τ_n is neglected, and the delay is taken as $d = 0.3$ s. The lead and lag time constants are adjusted, according to the adjustment rules, to optimize the closed-loop frequency response. When this has been done, the maximum lead or lag provided by the pilot model is determined, together with the value of the peak magnification. The pilot opinion rating is then determined from a plot like that shown in Figure 4.3-3.

A later development than the transfer function model of the human operator is the *optimal control model* (OCM), attributable mainly to Baron, Kleinman, and Levison [Kleinman et al., 1970]. It uses a state-variable formulation and optimal control theory and is based on the assumption that “a well-motivated, well-trained human operator behaves in a near optimal manner, subject to his inherent limitations and constraints and his control task.” A description of this model is outside our scope, since it has not found its way directly into flying qualities specifications. More information can be found in the book by Sheridan and Ferrill [1974] and in the references cited. A summary of work in human operator modeling, with a fairly comprehensive bibliography, has been given by Gerlach [Gerlach, 1977].

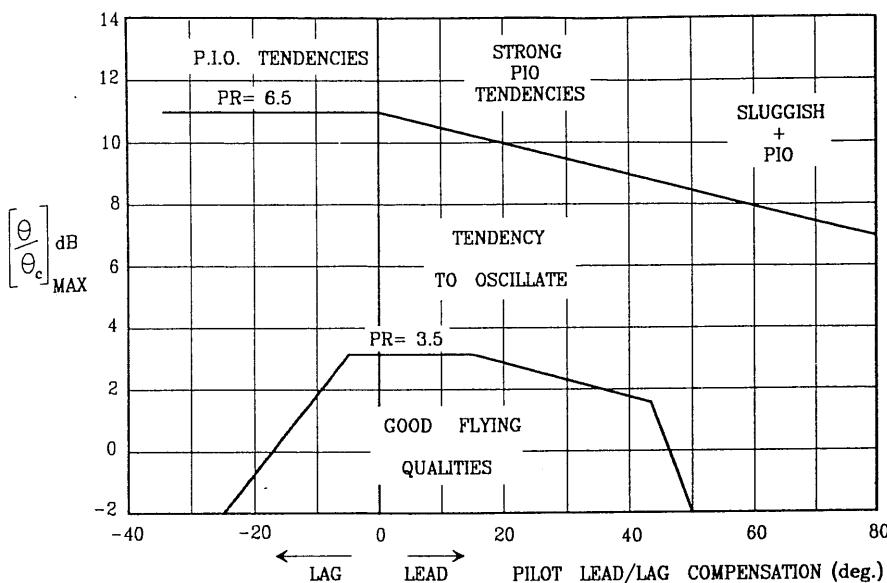


Fig. 4.3-3 Neal-Smith evaluation of pilot rating.

Human operator modeling applied to a pilot performing compensatory tracking tasks has now accumulated quite a long history, and attention has turned to modeling the human operator performing other piloting tasks. In a modern fighter aircraft the workload involved in operating all of the different systems (flight control, navigation, radar, weapons, etc.) is now overwhelming, and modeling the human decision-making process has become important. A survey of the relationship of flying qualities specifications to task performance, and the use of pilot models, has been given by George and Moorhouse [1982].

Other Requirements

The preceding subsections described ways in which the dynamic response of an aircraft and its control systems can be characterized, and how these may lead to flying qualities criteria. There are a number of other requirements that must be satisfied for an aircraft to receive a good flying-qualities rating. Some of these have no direct effect on control system design, but they are "inside the loop" that is closed by the pilot. They will be briefly described because of their importance.

One of the more important characteristics of the aircraft controls is the *control feel*, that is, the force and deflection characteristics of the control stick during a particular maneuver. Aircraft whose control surfaces are not

power boosted require much careful balancing of the control surfaces, and the addition of a mass and springs to the control stick, in order to obtain satisfactory feel. Aircraft with fully powered, irreversible controls require an artificial-feel system.

Artificial feel may take the form of centering springs, an electromechanical damper, and for longitudinal control, a mechanical or hydraulic system that provides a stick reaction force proportional to the normal acceleration in g 's during a pull-up maneuver. Iso-opinion studies have shown that the amount of stick force per g is quite critical and there is an associated optimum value of stick deflection. Stick force per g requirements are given in the military aircraft specifications, in addition to the control forces required in various flight phases.

Another factor that influences a pilot's opinion of flying qualities, particularly in the landing phase, is speed stability. The aircraft response to a speed disturbance is an exponential change, and this response will typically be rated as satisfactory if it is stable with a time constant of less than about 50 s. An unstable exponential response may be acceptable under some conditions, provided that the time constant is greater than about 25 s.

The Military Flying-Qualities Specifications

In the preceding subsections we attempted to convey some idea of the difficulty of specifying analytical performance criteria for the dynamic behavior of piloted aircraft. The civil and military aviation authorities of various countries are also faced with this problem. In general, their requirements documents are not very analytical and do not provide any way out of our difficulty. However, the U.S. "Military Specification for the Flying Qualities of Piloted Airplanes" [MIL-F-8785C, 1980] does provide some analytical specifications that must be met by U.S. military aircraft. A background document and user guide, containing much useful information and a large bibliography, is also available [Chalk et al., 1969]. These documents are readily available, and only the mode specifications of MIL-F-8785C will be summarized here (note that MIL-F-8785C has now been superseded by MIL 1797, which contains additional information, but this document has limited circulation).

The military specification defines airplane classes, flight phases, and flying qualities levels, so that different modes can be specified for the various combinations. These are defined in Table 4.3-2; the flying qualities levels are linked to the Cooper-Harper ratings as shown in Table 4.3-1.

The specifications for the aircraft modes are as follows.

Phugoid Specifications. The military specification dictates that for the different levels of flying qualities, the damping (ζ_p) and natural frequency

TABLE 4.3-2. Definitions: Flying-Qualities Specifications

<i>Airplane Classes</i>	<i>Definitions</i>
Class I	Small, light airplanes.
Class II	Medium weight, low-to-medium-maneuverability airplanes.
Class III	Large, heavy, low-to-medium-maneuverability airplanes.
Class IV	High-maneuverability airplanes.
<i>Flight Phases</i>	<i>Definitions</i>
Category A	Nonterminal flight phases generally requiring rapid maneuvering.
Category B	Nonterminal flight phases normally accomplished using gradual maneuvers without precision tracking, although accurate flight-path control may be required.
Category C	Terminal flight phases normally accomplished using gradual maneuvers and usually requiring accurate flight-path control.
<i>Flying-Qualities Levels</i>	<i>Definitions</i>
Level 1	Flying qualities adequate for the mission flight phase.
Level 2	Flying qualities adequate to accomplish the mission flight phase, but some increase in pilot workload or degradation in mission effectiveness exists.
Level 3	Flying qualities such that the airplane can be controlled safely, but pilot workload is excessive, or mission effectiveness is inadequate, or both.

(ω_{n_p}) of the phugoid mode will satisfy the following requirements:

$$\text{Level 1:} \quad \zeta_p \geq 0.04$$

$$\text{Level 2:} \quad \zeta_p \geq 0.0$$

$$\text{Level 3:} \quad T_{2_p} \geq 55.0 \text{ s.}$$

In the level 3 requirement the mode is assumed to be unstable, and T_2 denotes the time required for the mode to double in amplitude. For an

exponentially growing sinusoidal mode this time is given by

$$T_2 = \frac{\log_e 2}{-\zeta \omega_n} \quad (\zeta \text{ has negative values}).$$

These requirements apply with the pitch control free or fixed; they need not be met transonically in certain cases.

Short-Period Specifications. The short-period requirements are specified in terms of the natural frequency and damping of the *short-period mode* of the equivalent low-order system (as defined earlier). The adequacy of the equivalent system approximation is to be judged by the procuring agency. Table 4.3-3a shows the requirements on the equivalent short-period damping ratio (ζ_{sp}). The requirements on equivalent undamped natural frequency ($\omega_{n_{sp}}$) are given in Table 4.3-3b and are specified indirectly, in terms of the quantity $\omega_{n_{sp}}^2/(n/\alpha)$. The denominator (n/α) of this term is the aircraft load-factor response to angle of attack in g's per radian, as explained in the subsection on pole-zero specifications.

TABLE 4.3-3a. Short-Period Damping Ratio Limits

Level	Categories A and C Flight Phases		Category B Flight Phases	
	Minimum	Maximum	Minimum	Maximum
1	0.35	1.30	0.30	2.00
2	0.25	2.00	0.20	2.00
3	0.15 ^a	No limit	0.15 ^a	No limit

^aMay be reduced at altitude > 20,000 ft with approval.

TABLE 4.3-3b. Limits on $\omega_{n_{sp}}^2/(n/\alpha)$

Level	Category A Phases		Category B Phases		Category C Phases	
	Min.	Max.	Min.	Max.	Min.	Max.
1	0.28 $\omega_n \geq 1.0$	3.60	0.085	3.60	0.16 $\omega_n \geq 0.7$	3.60
2	0.16 $\omega_n \geq 0.6$	10.0	0.038	10.0	0.096 $\omega_n \geq 0.4$	10.0
3	0.16	No limit	0.038	No limit	0.096	No limit

^aThere are some additional limits on the minimum value of n/α and the minimum value of ω_n , for different classes of airplane in category C.

TABLE 4.3-4. Maximum Roll Mode Time Constant (seconds)

Flight Phase Category	Class	Level		
		1	2	3
A	I, IV	1.0	1.4	None
	II, III	1.4	3.0	None
B	All	1.4	3.0	10
C	I, II-C, IV	1.0	1.4	None
	II-L, III	1.4	3.0	None

Roll Mode Specifications. The maximum allowable value of the roll-subsidence mode time constant is given in Table 4.3-4. In addition to these time-constant specifications there are a comprehensive set of requirements on the time required to achieve various (large) changes in bank angle following an abrupt roll command. For example, for air-to-air combat (a flight phase within category A, for class IV airplanes) the minimum allowable time to achieve a certain bank angle depends on airspeed, but for level I flying qualities may be as short as 1.0 s for 90° bank and 2.8 s for a 360° roll.

Spiral Mode Specifications. The spiral mode is allowed to be unstable, but limits are placed on the minimum time for the mode to double in amplitude, as shown in Table 4.3-5. These requirements must be met following a bank-angle disturbance of up to 20° from trimmed-for-zero-yaw-rate wings-level flight, with the cockpit controls free.

Dutch Roll Mode Specifications. The frequency, ω_{n_d} , and damping ratio, ζ_d , of the dutch roll mode must exceed the minimum values given in Table 4.3-6. Note that the quantity $\zeta\omega_n$ is the *s*-plane real-axis coordinate of the roots, and ω_n is the radial distance from the origin for complex roots. Therefore, these requirements define an area of the *s*-plane in which the dutch roll roots must lie. The lower limit on ζ_d is the larger of the two values that come from the table, except that a value of 0.7 need not be exceeded for

TABLE 4.3-5. Spiral Mode, Minimum Doubling Time

Flight Phase Category	Level 1	Level 2	Level 3
A and C	12 s	8 s	4 s
B	20 s	8 s	4 s

TABLE 4.3-6. Dutch Roll Mode Specifications

Level	Category	Flight Phase	Class	ζ_d	$\zeta_d \omega_{n_d}$	ω_{n_d}
1	A		I, IV	0.19	0.35	1.0
			II, III	0.19	0.35	0.4
	B		All	0.08	0.15	0.4
			I, II-C, IV	0.08	0.15	1.0
			II-L, III	0.08	0.15	0.4
	All		All	0.02	0.05	0.4
2			All	0.02	No limit	0.04

class III. Also, class III airplanes may be exempted from some of the minimum ω_{n_d} requirements. Airplanes that have a large amount of roll-yaw coupling, as measured by the ratio of the maximum bank angle to the maximum value of sideslip in a dutch roll oscillation, are subject to a more stringent requirement on $\zeta_d \omega_{n_d}$ (see MIL-F-8785C).

The military requirements document specifies dynamic response mainly through the pole-zero requirements. These have been summarized here so that the reader may evaluate some of the controller designs described later. Much additional information covering other aspects of flying qualities is available in the requirements document, and it is essential reading for anyone with a serious interest in this field.

4.4 STABILITY AUGMENTATION

Most high-performance commercial and military aircraft will not meet the flying-qualities requirements of Section 4.3 without the use of a SAS. Some of these aircraft are actually unstable and would be impossible to fly without an automatic control system. The SAS typically uses sensors to measure the body-axes angular rates of the vehicle, and feeds back processed versions of these signals to the servomechanisms that drive the aerodynamic control surfaces. In this way an aerodynamic moment proportional to angular velocity and its derivatives can be generated and used to produce a damping effect on the motion. If the basic mode is unstable, or if it is desired to change both damping and natural frequency independently, additional feedback signals will be required, as we shall see.

Stability augmentation systems are conventionally designed separately for the longitudinal dynamics and the lateral-directional dynamics, and this is

made possible by the decoupling of the aircraft dynamics in most flight conditions. In the next two subsections the F-16 aircraft dynamics will be used to describe the design of the various augmentation systems.

Pitch-Axis Stability Augmentation

The purpose of a pitch SAS is to provide satisfactory natural frequency and damping for the short-period mode. This mode involves the variables alpha and pitch rate, and feedback of these variables to the elevator control will modify the frequency and damping. The phugoid mode will be largely unaffected by this feedback. Outer feedback control loops will often be closed around the pitch SAS to provide autopilot functions or pilot control of, for example, normal (z-axis) acceleration. Automatic adjustment of the augmentation (inner) loop feedback gains may be arranged when the outer feedback loops are engaged, so that the overall performance is optimal.

Figure 4.4-1 shows feedback of alpha and pitch rate for control of the short-period mode. The transfer function G_A represents the elevator actuator (or the complete closed-loop elevator servo), and the transfer function G_F is a noise filter, used to low-pass filter the angle-of-attack signal. If the short-period mode is lightly damped but otherwise adequate, only pitch-rate feedback is required, and pilot commands can be applied at the u input. If the frequency and damping are both unsatisfactory or the mode is unstable, alpha feedback is necessary and the integrator with outer-loop feedback will be shown to be required.

The pitch-rate sensor is normally a mechanical gyroscopic device, and it measures the absolute angular rate about its sensitive axis. A set of three orthogonal gyros is used to measure the angular rates around the aircraft-body axes. The location of these gyros must be chosen very carefully to avoid picking up the vibrations of the aircraft structure. In an idealized structural oscillation, at a node there is angular motion but no displacement; and at an antinode the converse is true. Thus the first choice for the rate gyro location

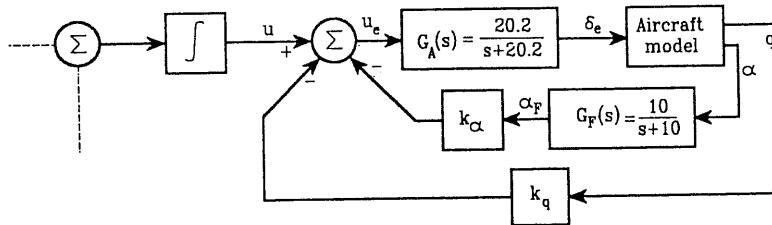


Fig. 4.4-1 A Pitch stability and augmentation system.

is an antinode corresponding to the most important structural mode. Flight tests must then be used to adjust the position of the gyros. A bad choice of gyro locations can adversely affect flying qualities or, in extreme cases, cause oscillations in the flight control systems (see, e.g., AFWAL-TR-84-3105 [1984]). Some degree of filtering of the gyro output will normally be required.

The alpha sensor may be pressure probes or a small wind vane mounted on the side of the aircraft forebody and positioned (after much testing and calibration) to measure alpha over a wide range of flight conditions. Two sensors may be used, on opposite sides of the aircraft, to provide redundancy and possibly to average out measurement errors caused by sideslipping. A "true" angle of attack must usually be computed (in the air-data computer) from the "indicated angle of attack," airspeed, and Mach number, in order to relate the free-stream angle of attack of the airframe to the direction of the flow field at the sensor position. The signal from the alpha sensor is often noisy due to turbulence, and the noise filter reduces the amount of noise injected into the control system.

Alpha feedback is avoided if possible because of the difficulty of getting an accurate noise-free measurement and because of the vulnerability of the sensor to mechanical damage. Noise from the alpha sensor can make it difficult to achieve precise pointing (e.g., for targeting), so the amount of alpha feedback is normally restricted. We shall see later that it can be eliminated completely when the basic aircraft dynamics are stable.

The longitudinal (four-state) Jacobian matrices for the nominal condition in Table 3.4-3 are

$$\begin{aligned} A &= \begin{bmatrix} v_T & \alpha & \theta & q \\ -1.9311E-02 & 8.8157E+00 & -3.2170E+01 & -5.7499E-01 \\ -2.5389E-04 & -1.0189E+00 & 0.0000E+00 & 9.0506E-01 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ 2.9465E-12 & 8.2225E-01 & 0.0000E+00 & -1.0774E+00 \end{bmatrix} \\ B &= \begin{bmatrix} 1.7370E-01 \\ -2.1499E-03 \\ 0.0000E+00 \\ -1.7555E-01 \end{bmatrix} \\ C &= \begin{bmatrix} 0.000000E+00 & 5.729578E+01 & 0.000000E+00 & 0.000000E+00 \\ 0.000000E+00 & 0.000000E+00 & 0.000000E+00 & 5.729578E+01 \end{bmatrix} \begin{matrix} \alpha \\ q \end{matrix} \end{aligned} \quad (4.4-1)$$

The single input is the elevator deflection δ_e , in degrees, and the two outputs are the appropriate feedback signals: alpha and pitch rate. The entries in the C matrix are the conversions to units of degrees, for consistency with the input.

Either of the two SISO transfer functions obtained from the coefficient matrices (4.4-1) will exhibit the dynamic modes for this flight condition; the elevator-to-alpha transfer function is

$$\frac{\alpha}{\delta_e} = \frac{-0.1232(s + 75.00)(s + 0.009820 \pm j0.09379)}{(s - 0.09755)(s + 1.912)(s + 0.1507 \pm j0.1153)}. \quad (4.4-2)$$

Unlike the transfer functions for stable cg positions (e.g., $x_{cg} = 0.3\bar{c}$) in Chapter 3, this transfer function does not exhibit the usual phugoid and short-period poles. The pole at $s \approx 0.1$ indicates an unstable exponential mode with a time constant of 10 s. The complex-conjugate pole pair corresponds to an oscillatory mode with a period of 33 s and damping ratio of 0.79; this is like a phugoid period with a short-period damping ratio. This mode is the “third oscillatory mode” of the statically unstable airplane (see Section 4.2).

The modes described above obviously do not satisfy the requirements for good handling qualities, and providing continuous control of the unstable mode would be a very demanding job for a pilot. A pitch SAS can be designed to restore a conventional short period and is essential for this airplane. The explanation of the unstable mode can be traced back to Chapter 2, where the condition of positive pitch stiffness was described. The F-16 pitching moment curve has a positive slope over some ranges of alpha and the model is statically unstable in these regions, which leads to an unstable dynamic mode. This provides an additional clue to the type of feedback needed to stabilize the unstable mode. If deviations from the trim angle of attack are sensed and fed back to the elevator servo to generate a restoring pitching moment, the slope of the pitching moment curve can be made negative in the region around the trim angle of attack. The short-period mode will then be stable. Furthermore, the overall pitching moment curve and the trimmed elevator deflection will not be affected, thus preserving the trim-drag and maneuverability characteristics that the designer built into the basic airplane design.

The “*trim angle of attack*” loses its meaning as the aircraft maneuvers, and in order to make this scheme work, a way must be found to compute an alpha reference signal continuously. One way in which an alpha reference may be provided automatically is to include an integrator in the control loops. The loops can be arranged so that if the pilot’s command inputs have been satisfied, there is no input to the integrator and its output remains constant at a value that corresponds to the reference angle of attack. This integrator is shown in Fig. 4.4-1.

Alpha feedback and pitch-rate feedback together provide virtually complete control of the position of the short-period poles. This is demonstrated in the following example.

Example 4.4-1: The Effects of Pitch-Rate and Alpha Feedback. The configuration shown in Fig. 4.4-1 will be used, together with the aircraft dynamics in (4.4-1), to illustrate the effects of pitch-rate and alpha feedback. First, the basic aircraft dynamics will be augmented with the actuator and noise-filter models. These will be taken from the original F-16 model report (Nguyen et al., 1979) and are both simple-lag filters with time constants $\tau_a = 1/20.2$ s and $\tau_F = 0.1$ s, respectively. A simple-lag model and its state equation were described in Chapter 3, and when the coefficient matrices

(4.4-1) are augmented with these additional dynamics, the result is

$$\dot{x} = \begin{bmatrix} A & B \\ \hline 0 & -1/\tau_a \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ q \\ x_a \\ x_F \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/\tau_a \\ 0 \end{bmatrix} u_e \quad (1)$$

$$y = \begin{bmatrix} \alpha \\ q \\ \alpha_F \end{bmatrix} = \begin{bmatrix} C \\ \hline 0 & 0 & 0 & 0 & 0 & 57.29578 \end{bmatrix} x.$$

Notice that the original state equations are still satisfied, and that the original δ_e input is now connected to the actuator state x_a . The actuator is driven by a new input u_e . Also, the α filter is driven by the α state of the aircraft dynamics, and an additional output has been created so that the filtered signal, α_F , is available for feedback. These state equations could also have been created by simulating the filters as part of the aircraft model and running the linearization program again.

The state equations (1) can now be used to obtain the loop transfer functions needed for root-locus design. In the case of the innermost (α) loop, we already know that the α -loop transfer function will consist of (4.4-2) with the two lag filters in cascade, and the effect of the feedback k_α can be anticipated using a sketch of the pole and zero positions. The goal of the α feedback is to pull the unstable pole, at $s = 0.098$, back into the left-half s -plane. According to the root-locus rules, the locus from this pole will move to the left along the real axis if the static loop sensitivity is positive (assuming negative feedback as shown at the summing junction in the figure). The static loop sensitivity resulting from the transfer function (4.4-2) is negative; therefore, negative values of k_α will be used with the root-locus program.

Figure 4.4-2a and b show the root-locus plot for the inner loop on two different scales. The expanded scale near the origin (Fig. 4.4-2b) shows that the effect of the α feedback is to make the loci from the third-mode poles come together on the real axis (near $s = -0.2$). The branch going to the right then meets the branch coming from the unstable pole, and they leave the real axis to terminate on the complex zeros near the origin. This provides a pair of closed-loop poles that correspond to a phugoid mode. The left branch from the third mode poles meets the branch from the pole at $s = -1.9$, and they leave the axis near $s = -1$ to form a short-period mode.

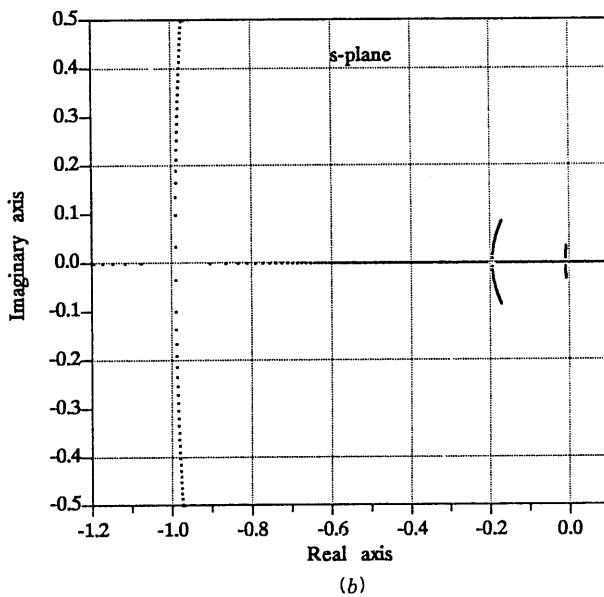
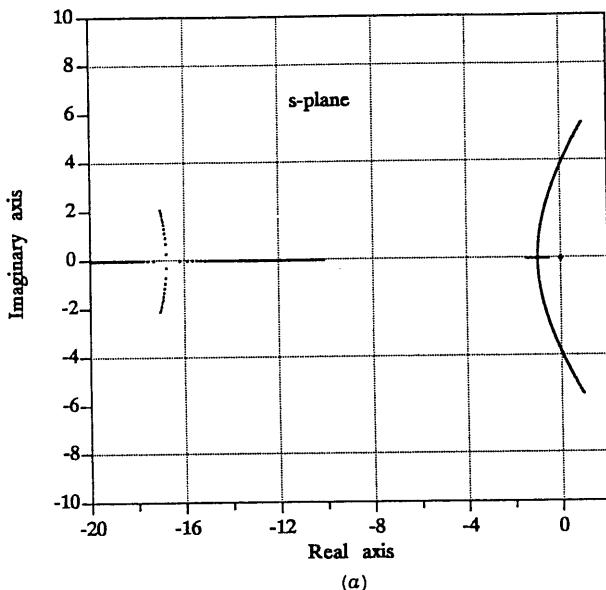


Fig. 4.4-2 Inner-loop root-locus plots for the pitch SAS.

Alpha feedback has therefore produced the anticipated effect: the aircraft is stable with conventional longitudinal modes.

The larger-scale plot (Fig. 4.4-2a) shows that as the magnitude of the alpha feedback is increased, the frequency of the new short-period poles increases and they move toward the right-half plane. The movement toward the right-half plane is due to the repulsion effect of the filter and actuator poles out at $s = -10$ and $s = -20.2$; the effect on the short-period locus is relatively small because they are well to the left in the s -plane. Therefore, a fast (and expensive) actuator allows a higher short-period frequency to be achieved. The position of the short period poles for $k_\alpha = -0.5$ is $(-0.70 \pm j2.0)$. At this position the natural frequency is about 2.2 rad/s, which is acceptable according to the flying qualities requirements, but the damping ratio ($\zeta = 0.33$) is quite low.

The root-locus program can now be applied to determine the effect of varying k_q , with k_α fixed at -0.5 . The q/u transfer function with $k_\alpha = -0.5$ and $k_q = 0$, is

$$\frac{q}{u} = \frac{-203.2s(s + 10.0)(s + 1.027)(s + 0.02174)}{(s + 20.00)(s + 10.89)(s + 0.6990 \pm j2.030)(s + 0.008458 \pm j.08269)}. \quad (2)$$

Note that the zeros of this transfer function are the unaugmented open-loop zeros, with the addition of the zero at $s = -10$. This zero originally canceled the alpha-filter pole out of the pitch-rate transfer function, but the inner-loop feedback has now moved the alpha-filter pole to $s = -10.89$.

Figure 4.4-3 shows the root-locus plot for negative values of k_q . The short-period poles follow an approximately circular arc (constant natural

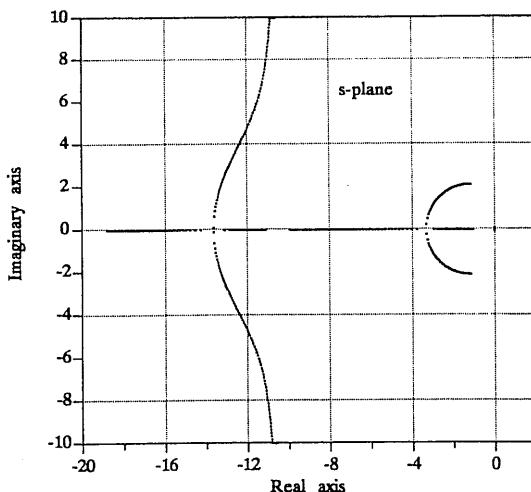


Fig. 4.4-3 Outer-loop root-locus plot for the pitch SAS.

frequency, increasing damping) as the pitch rate feedback is increased. The poles become real for quite low values of k_q , and, with larger values, a new higher-frequency oscillatory mode is created by the filter and actuator poles. Such values of k_q are probably not feasible design values because actuator rate limiting would prevent the system from following the higher-frequency mode. Furthermore, the slow pole (near the origin) created from the short-period poles may contribute significantly to the response. Therefore, selecting a low value of k_q , the value $k_q = -0.25$ places the short-period poles at $s = -2.02 \pm j1.94$. This corresponds to a natural frequency of 2.8 rad/s and a damping ratio of $\zeta = 0.72$. The corresponding closed-loop transfer function for pitch rate is given by

$$\frac{q}{u} = \frac{-203.2s(s + 10.0)(s + 1.027)(s + 0.02174)}{(s + 16.39)(s + 11.88)(s + 2.018 \pm j1.945)(s + 0.008781 \pm j0.06681)}. \quad (3)$$

The original actuator pole has moved from $s = -20.2$ to $s = -16.39$, and the α -filter pole has moved from $s = -10$ to $s = -11.88$. Apart from these factors, this transfer function is very similar to the stable-cg transfer function in Example 3.7-5 but with improved short-period pole positions.

Finally, note that both feedback gains are negative, and when the negative sign at the summing junction (Fig. 4.4-1) is taken into account, this means that an increasing pitch angle or increasing alpha both give the elevator a positive displacement. According to the sign convention in Section 3.3, this creates a nose-down pitching moment. This is a restoring moment, and therefore the feedback polarities, arrived at by a straightforward application of the root-locus principles, agree with a physical interpretation of the feedback requirements. ■

The ability of angle-of-attack and pitch-rate feedback to allow the short-period poles to be positioned at will has now been demonstrated. When the basic airplane is stable in pitch, desirable short-period pole locations may be achievable with pitch-rate feedback only, thus eliminating the alpha sensor. An integrator was assumed to be present at the reference input to the SAS to provide the angle-of-attack reference. This is not a practical arrangement unless an outer loop is closed around the integrator. Therefore, the SAS alone would not be considered practical for an unstable aircraft. However, as a hypothetical alternative, a SAS with washed-out alpha feedback will now be investigated.

Example 4.4-2: A Pitch SAS with Washed-out Alpha-Feedback. A washout circuit is a simple high-pass filter, as shown in Table 3.2-1. If it is included in the alpha feedback path in Fig. 4.4-1, a steady-state angle of attack will develop a steady voltage across the washout-circuit capacitor. Angle-of-attack changes which are fast relative to the time constant of the washout will be fed back directly to the elevator actuator, thus achieving the

desired incremental alpha feedback. If the coefficient matrices in Example 4.4-1 are augmented with the washout filter, the result is

$$\dot{x} = \begin{bmatrix} A & & B \\ \hline 0 & 0 & 0 & 0 & -1/\tau_a & 0 & 0 \\ 0 & 1/\tau_F & 0 & 0 & 0 & -1/\tau_F & 0 \\ 0 & 0 & 0 & 0 & 0 & 57.3 & -1/\tau_W \end{bmatrix} \begin{bmatrix} v_T \\ \alpha \\ \theta \\ q \\ \dot{x}_a \\ x_F \\ x_W \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1/\tau_a \\ 0 \\ 0 \end{bmatrix} u_e$$

$$y = \begin{bmatrix} q \\ \alpha_F \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 57.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 57.3 & -1/\tau_W \end{bmatrix} x,$$

where "57.3" indicates the radians-to-degrees conversion 57.29578.

If a 5-s washout time constant is chosen and the gains k_α and k_q are again set to -0.5 and -0.25 , respectively, the following closed-loop pitch-rate transfer function is obtained:

$$\frac{q}{u} = \frac{-203.2s(s + 10.0)(s + 1.027)(s + 0.02174)(s + 0.20)}{(s + 16.38)(s + 11.90)(s - 0.020)(s + 2.072 \pm j2.044)(s + 0.05497 \pm j.07130)}.$$

This transfer function agrees closely with the transfer function (3) in Example 4.4-1, apart from the zero created by the washout in the alpha feedback loop and the unstable slow pole at $s = 0.020$. The unstable pole comes from the original aircraft pole at $s = 0.098$, which is now unable to move into the left-half plane because of the washout zero at the origin. The unstable exponential mode corresponding to this pole is slow enough that a pilot could easily control it, but it represents an inconvenience that makes this approach unlikely to be used. ■

Lateral – Directional Stability Augmentation

Figure 4.4-4 shows the most basic augmentation system for the lateral–directional dynamics. Body-axis roll rate is fed back to the ailerons to modify the

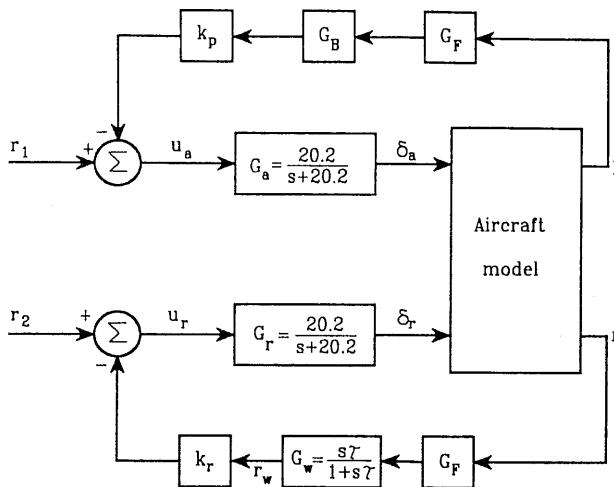


Fig. 4.4-4 Lateral/directional augmentation system.

roll-subsidence mode, and yaw rate is fed back to the rudder to modify the dutch roll mode. The lateral (rolling) motion is not, in general, decoupled from the yawing and sideslipping (directional) motions. Therefore, the augmentation systems will be analyzed with the aid of the multivariable state equations (two inputs, ailerons and rudder, and two or more outputs), as implied by the figure. This analysis will be restricted to the simple feedback scheme shown in the figure; in a later section, additional feedback couplings will be introduced between the two channels (roll and yaw).

The purpose of the stability augmentation yaw-rate feedback is to use the rudder to generate a yawing moment that opposes any yaw rate that builds up from the dutch roll mode. The resulting feedback configuration is a Type 0 yaw-rate command system with zero command input. This raises a difficulty; in a coordinated steady-state turn the yaw rate has a constant nonzero value. It may be thought that the pilot could easily coordinate the turn by applying the correct yaw rate command through the rudder pedals. Flight control experience shows that this is not the case.

The yaw-rate feedback will oppose the turn unless the pilot applies the necessary yaw rate command, and the increased command needed to overcome the finite error of the Type 0 control loop has been found to be unacceptable. Even in a steeply banked turn, very little rudder deflection is needed (as seen in Table 3.4-3) to produce the required yaw rate, and in a high-performance aircraft the turn is essentially produced by rolling to the required bank angle and applying back-pressure to the control stick to produce the normal acceleration. A simple solution to the problem of implementing the yaw damper is to use a washout circuit on the output of the yaw-rate sensor. The high-pass filtering action of the washout circuit removes

the steady-state component of the yaw rate during turns. The output of the washout approximates differentiated yaw rate, and this is still a suitable feedback for damping the dutch roll mode.

In Fig. 4.4-4, G_W is the washout circuit, the transfer function G_a represents an equivalent transfer function for differential actuation of the left and right ailerons, and G_r is the rudder actuator. The transfer functions G_F represent noise filtering and any effective lag at the output of the roll-rate and yaw-rate gyros, and G_B is a *bending mode filter*. The bending mode filter is needed because the moments generated by the ailerons are transmitted through the flexible-beam structure of the wing, and their effect is sensed by the roll-rate gyro in the fuselage. The transfer function of this path corresponds to a general low-pass filtering effect, with resonances occurring at the bending modes of the wing. Because the wing bending modes are relatively low in frequency, they may contribute significant phase shift, and possibly gain changes, within the bandwidth of the roll-rate loop. The bending-mode filter is designed to compensate for these phase and gain changes.

To understand the purpose of the roll-rate feedback, consider the following facts. In Section 4.2 the variation of the roll time constant with flight conditions was analyzed, and in Chapter 2 the change of aileron effectiveness with angle of attack was described. These effects cause large, undesirable variations in aircraft roll performance that result in the pilot flying the aircraft less precisely. Closed-loop control of roll rate is used to reduce the variation of roll performance with flight conditions, and if the feedback is changed as a function of angle of attack, the amount of variation in the closed-loop properties can be made relatively small.

While the roll time constant is a feature of the linear small-perturbation model and gives no indication of the maximum roll rate or time to roll through a large angle, it is relevant to the initial speed of response and control of smaller-amplitude motion. Figure 4.4-5 shows a plot of the reciprocal of the F-16 roll time constant versus alpha and shows that this time constant may become unacceptably slow at high angles of attack. The plot was derived by trimming the F-16 model in straight and level flight at sea level, with the nominal cg position, over a range of speeds. At angles of attack greater than about 20° , the roll pole coupled with the spiral pole to form a complex pair.

Landing approach takes place at a relatively high angle of attack, and the roll-rate feedback may be needed to ensure good roll response. Also, satisfactory damping of the dutch roll mode is particularly important during landing approach in gusty crosswind conditions. Our F-16 model is not equipped with flaps and landing gear, so the design of the augmentation loops will simply be illustrated on a low-speed, low-altitude flight condition. If we take the F-16 model dynamics at zero altitude, with the nominal cg position and an airspeed of 205.0 ft/s ($\alpha = 18.8^\circ$), the roll pole is real and quite slow ($\tau = 1.44$ s), and the dutch roll is very lightly damped ($\zeta = 0.2$). The state equations can be found with the aid of the linearization

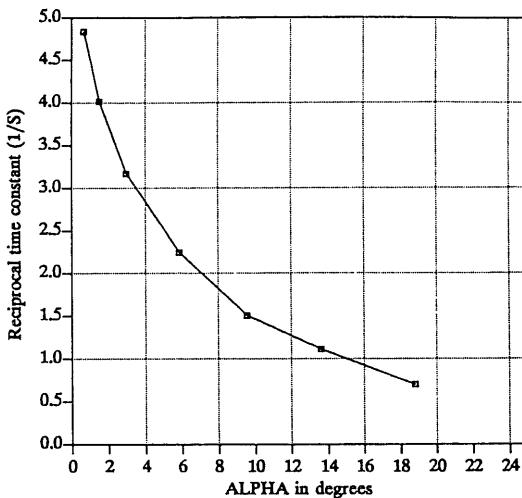


Fig. 4.4-5 Variation of F-16 roll time constant with alpha.

program, and a five-state set of lateral-directional equations can be decoupled from the full 13-state set. The coefficient matrices are found to be

$$\begin{aligned}
 A &= \begin{bmatrix} \beta & \phi & \psi & p & r \\ -0.13150 & 0.14858 & 0.0 & 0.32434 & -0.93964 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.33976 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0561 \\ -10.614 & 0.0 & 0.0 & -1.1793 & 1.0023 \\ 0.99655 & 0.0 & 0.0 & -0.0018174 & -0.25855 \end{bmatrix} \\
 B &= \begin{bmatrix} \delta_a & \delta_r \\ 0.00012049 & 0.00032897 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \\ -0.1031578 & 0.020987 \\ -0.0021330 & -0.010715 \end{bmatrix} \\
 C &= \begin{bmatrix} 0.0 & 0.0 & 0.0 & 57.29578 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 57.29578 \end{bmatrix} r.
 \end{aligned} \tag{4.4-3}$$

The null column in the A matrix shows that the state ψ is not coupled back to any other states, and it can be omitted from the state equations when designing an augmentation system. The C matrix has been used to convert the output quantities to degrees (to correspond to the control surface inputs).

The transfer functions of primary interest are

$$\frac{p}{\delta_a} = \frac{-5.911(s - 0.05092)(s + 0.2370 \pm j1.072)}{(s + 0.06789)(s + 0.6960)(s + 0.4027 \pm j2.012)} \quad (4.4-4)$$

$$\frac{r}{\delta_a} = \frac{-0.1222(s + 0.4642)(s + 0.3512 \pm j4.325)}{(s + 0.06789)(s + 0.6960)(s + 0.4027 \pm j2.012)} \quad (4.4-5)$$

$$\frac{p}{\delta_r} = \frac{+1.202(s - 0.05280)(s - 2.177)(s + 1.942)}{(s + 0.06789)(s + 0.6960)(s + 0.4027 \pm j2.012)} \quad (4.4-6)$$

$$\frac{r}{\delta_r} = \frac{-0.6139(s + 0.5078)(s + 0.3880 \pm j1.5439)}{(s + 0.06789)(s + 0.6960)(s + 0.4027 \pm j2.012)}. \quad (4.4-7)$$

The dutch roll poles are not canceled out of the p/δ_a transfer function by the complex zeros. Therefore, coupling exists between the rolling and yawing motions, and the dutch roll mode will involve some rolling motion. These transfer functions validate the decision to use the MIMO state equations for the analysis. At lower angles of attack the dutch roll poles will typically be canceled out of the p/δ_a transfer function, leaving only the roll-subsidence and spiral poles.

The two roll-rate transfer functions, given above, contain NMP zeros close to the origin. This is because gravity will cause the aircraft to begin to sideslip as it rolls. Then if the dihedral derivative, C_{l_B} , is negative (positive roll stiffness), the aircraft will have a tendency to roll in the opposite direction. This effect will be more pronounced in a slow roll when the sideslip has a chance to build up.

The rudder-to-roll-rate transfer function has another NMP zero farther away from the origin, corresponding to faster-acting non-minimum-phase effects. A positive deflection of the rudder directly produces a positive rolling moment (see Table 3.3-1) and a negative yawing moment. The negative yawing moment rapidly leads to positive sideslip, which will in turn produce a negative rolling moment if the aircraft has positive roll stiffness. This effect tends to cancel the initial positive roll, and the NMP zero is the transfer function manifestation of these competing effects.

Example 4.4-3: A Roll Damper / Yaw Damper Design. The aileron and rudder actuators in Fig. 4.4-4 will be taken as simple lags with a corner frequency ($1/\tau$) of 20.2 rad/s (as in the original model), and the bending mode filter will be neglected. The state equations are obtained by augmenting the coefficient matrices (4.4-3) with the actuator models and a washout

filter, leading to

$$\dot{x} = \begin{bmatrix} A & & B \\ \hline 0 & 0 & 0 & 0 & -1/\tau_a & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1/\tau_r & 0 \\ 0 & 0 & 0 & 57.3 & 0 & 0 & -1/\tau_w \end{bmatrix} \begin{bmatrix} \beta \\ \phi \\ p \\ r \\ \delta_a \\ \delta_r \\ x_w \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/\tau_a & 0 \\ 0 & 1/\tau_r \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_a \\ u_r \end{bmatrix} \quad (1)$$

$$y = \begin{bmatrix} p \\ r_w \end{bmatrix} = \begin{bmatrix} 0 & 0 & 57.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 57.3 & 0 & 0 & -1/\tau_w \end{bmatrix} x,$$

where x_w is the washout state. Note that "57.3" indicates the radians-to-degrees conversion 57.29578.

The washout filter time constant is a compromise; too large a value is undesirable since the yaw damper will then interfere with the entry into turns. The following root-locus design plots can also be used to show that too small a value will reduce the achievable dutch roll damping (see Problem 4.4-3). The time constant is normally of the order of 1 s, and $\tau_w = 1.0$ s is used here.

Experience shows that the roll-damping loop is the less critical loop, and is conveniently closed first. This can be done by using a root-locus program and computing the root loci as a function of k_p . The p/u_a transfer function is the same as (4.4-4) with an additional pole at $s = -20.2$ and the static loop sensitivity changed to -119 (i.e., 20.2 times the original value of -5.91). Figure 4.4-6 is the root-locus plot for negative values of k_p . It shows that the feedback has had the desired effect of speeding up the roll-subsidence pole, which moves to the left in the s -plane and eventually combines with the actuator pole to form a complex pair. The spiral pole moves a little to the right (toward the NMP zero at $s = 0.05$), and the dutch roll poles change significantly as they move toward the open-loop complex zeros. If the feedback gain is made too high in this design, it will be found to be excessive at lower angles of attack. Furthermore, a high value will simply cause the aileron actuators to reach their rate and deflection limits more rapidly, as they become less effective at the higher angles of attack. A feedback gain of $k_p = -0.2$ puts the roll subsidence pole at $s = -1.37$, which is about twice as fast as the open-loop value. This is a suitable starting value for investigating the effect of closing the yaw damper loop.

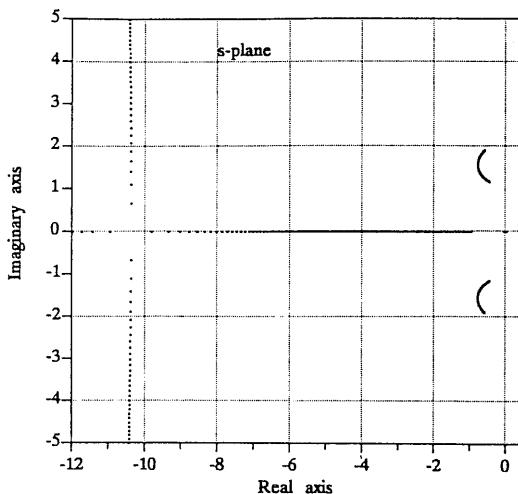


Fig. 4.4-6 Root-locus plot for the roll damping loop.

The transfer function of the yaw-rate loop can be found by using a program to close the roll-rate loop in (1), with the feedback gain matrix

$$K = \begin{bmatrix} k_p & 0 \\ 0 & 0 \end{bmatrix} \quad (2)$$

and $k_p = -0.20$. From the resulting closed-loop matrices, the transfer function r_w/u_r (with the roll-rate loop closed) is found to be

$$\frac{r_w}{u_r} = \frac{-12.4s(s + 18.8)(s + 0.760)(s + 0.961 \pm j0.947)}{(s + 1)(s + 18.9)(s + 1.37)(s + 0.0280)(s + 20.2)(s + 0.752 \pm j1.719)}. \quad (3)$$

A root-locus plot for closing the yaw-rate loop through the feedback gain k_p is shown in Fig. 4.4-7. Although not shown in the figure, one of the actuator poles is effectively canceled by the zero at $s = -18.8$, the remaining actuator pole moves to the right to meet the roll pole and form a new complex pair. As the magnitude of k_p is increased, the spiral pole moves slightly closer to the washout zero at the origin, and the washout pole moves toward the zero at $s = -0.76$. The dutch roll poles move (approximately)

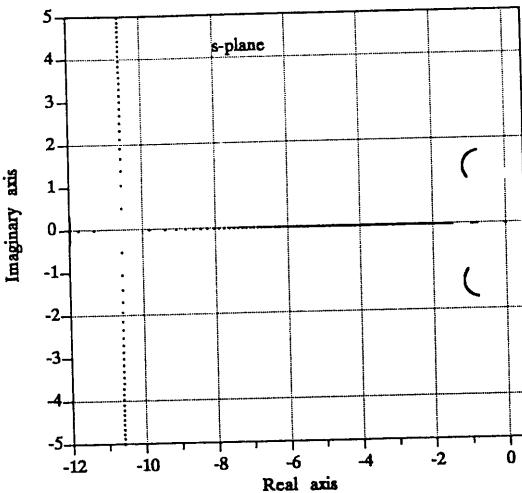


Fig. 4.4-7 Root-locus plot for the yaw-rate loop.

around an arc of constant natural frequency and increasing damping ratio toward the complex zeros. After k_r reaches about -3.5 , the natural frequency begins to decrease and the damping ratio tends to remain constant. This feedback gain was considered to be the optimum value for the dutch roll poles, so the complete feedback gain matrix was taken as

$$K = \begin{bmatrix} -0.2 & 0 \\ 0 & -3.5 \end{bmatrix}. \quad (4)$$

The principal transfer functions were then found to be

$$\frac{p}{r_1} = \frac{-119.4(s + 17.4)(s - 0.0502)(s + 3.74)(s + 0.262 \pm j0.557)}{(s + 18.7)(s + 17.7)(s + 0.0174)(s + 3.29)(s + 0.861)(s + 1.18 \pm j1.33)} \quad (5)$$

$$\frac{r}{r_2} = \frac{-12.4(s + 18.8)(s + 1.00)(s + 0.760)(s + 0.961 \pm j0.947)}{(s + 17.7)(s + 18.7)(s + 3.29)(s + 0.861)(s + 0.0174)(s + 1.18 \pm j1.33)}, \quad (6)$$

where r_1 and r_2 are the roll-rate and yaw-rate reference inputs, as shown in Fig. 4.4-4.

Transfer functions (5) and (6) show that the dutch roll poles, and the washout pole (at $s = -0.861$), do not cancel out of the p/r_1 transfer

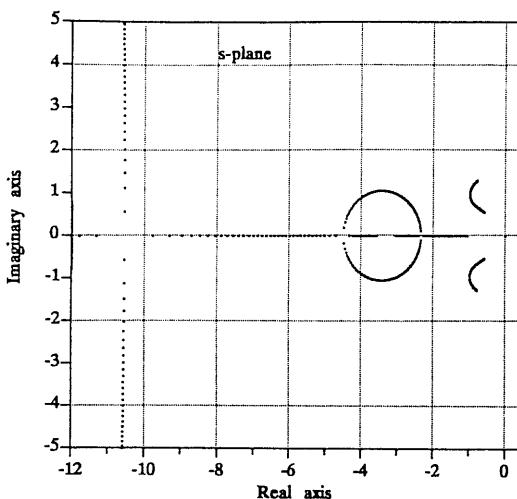


Fig. 4.4-8 Root-locus plot for the yaw-rate loop.

function, so there is still strong coupling between the roll and yaw channels. The dutch roll natural frequency and damping ($\omega_n = 1.78 \text{ rad/s}$, $\zeta = 0.67$) are now satisfactory, but the appearance of the relatively slow washout pole in the lateral dynamics may mean that the roll response is not much improved. Since we no longer have a simple dominant poles situation, a time-response simulation is needed to assess the design. Before this is undertaken, the effect of a higher gain in the roll-rate loop will be considered.

If the roll-rate loop is closed with $k_r = -0.4$, the roll subsidence pole moves out to $s = -3.08$, and the zero in the yaw-rate loop transfer function (3) moves from $s = -0.76$ to $s = -3.40$. This causes different behavior in the root-locus plot for the yaw-rate loop, as shown in Fig. 4.4-8. The washout pole now moves to the left instead of the right. A comparison of Figs. 4.4-7 and 4.4-8 shows that the price paid for this potential improvement in roll response is that the maximum dutch roll frequency is reduced. If the yaw-rate loop is closed with $k_r = -1.3$, to obtain the highest possible frequency for the dutch roll poles, the closed-loop transfer functions are

$$\frac{p}{r_1} = \frac{-119.4(s + 19.27)(s + 1.74)(s - 0.0507)(s + 0.334 \pm j0.787)}{(s + 19.25)(s + 17.4)(s + 0.00767)(s + 2.82)(s + 1.57)(s + 0.987 \pm j0.984)} \quad (7)$$

$$\frac{r}{r_2} = \frac{-12.40(s + 1.00)(s + 17.1)(s + 3.40)(s + 0.486 \pm j0.459)}{(s + 19.25)(s + 17.4)(s + 0.00767)(s + 2.82)(s + 1.57)(s + 0.987 \pm j0.984)} \quad (8)$$

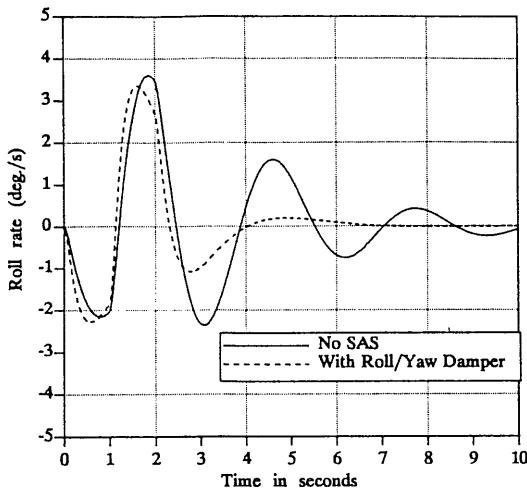


Fig. 4.4-9 Roll-rate response to an aileron doublet.

The dutch roll frequency has decreased to $\omega_n = 1.39$ rad/s, and the damping has increased to $\zeta = 0.71$; these values still represent good flying qualities (see Table 4.3-6). An improvement in the roll response should have been obtained since the slow washout pole is nearly canceled by the zero at $s = -1.74$, and the roll-subsidence pole (at $s = -2.82$) may now dominate the roll response. Note the way in which one actuator pole almost cancels out of each transfer function. Also, in the yaw-rate response, note the zero at $s = -1$ that originally canceled the washout pole. The transfer functions still show significant roll-yaw coupling.

The roll response of this design can only be assessed with a simulation, and because of the presence of the slow spiral pole in the transfer functions, a doublet pulse should be used as the input. The time responses were obtained by closing the yaw-rate and roll-rate loops with the feedback gains above ($k_p = -0.4$, $k_r = -1.3$) and running a time-history program on the closed-loop coefficient matrices. Figure 4.4-9 compares the roll-rate response of the open-loop dynamics (augmented with the actuators) with the closed-loop response. The doublet input is of 2 s duration, with unit amplitude in the open-loop case. In the closed-loop case the overall gain is different, and the doublet was adjusted to 1.8° so that the responses were of similar amplitude. The figure exhibits the major improvement in the dutch roll damping and the small but significant improvement in the roll-rate speed of response. ■

This example indicates the difficulties of multivariable design when significant cross-coupling is present in the dynamics. It also shows the difficulty of

obtaining a good roll response at low dynamic pressure and high alpha. The design could be pursued further by investigating the effect of changing the washout time constant and using compensation networks, such as a phase lead, in the yaw-rate feedback loop. As pointed out earlier, increasing the bandwidth of the control loops may simply lead to saturation of the control-surface actuators, and the limitations of the basic aircraft must be considered first.

4.5 CONTROL AUGMENTATION SYSTEMS

When an aircraft is under manual control (as opposed to autopilot control) the stability augmentation systems of the preceding section are, in most cases, the only automatic flight control systems needed. But in the case of high-performance military aircraft, where the pilot may have to maneuver the aircraft to its performance limits and perform tasks such as precision tracking of targets, specialized *control augmentation* systems are needed. Flight control technology has advanced to the point where the flight control system (FCS) can provide the pilot with selectable "task-tailored control laws." For example, although the role of a fighter aircraft has changed to include launching missiles from long range, the importance of the classical dogfight is still recognized. A dogfight places a premium on high maneuverability or "agility" in the aircraft and a control system that allows the pilot to take advantage of this maneuverability. In this situation a suitable controlled variable for the pitch axis is *normal acceleration* (or load factor) on the aircraft. This is the component of acceleration measured by an accelerometer in the negative direction of the aircraft *z*-axis. It is directly relevant to performing a maximum-rate turn and must be controllable up to the structural limits of the airframe, or the pilot's physical limits. Therefore, for a dogfight, a "g-command" control system is an appropriate mode of operation of the flight-control system. Other reasons for using this type of system will be described when we come to consider an example.

Another common mode of operation for a pitch-axis control augmentation system is as a pitch-rate command system. When a situation requires precision tracking of a target, by means of a sighting device, it has been found that a deadbeat response to pitch-rate commands is well suited to the task. Control of pitch rate is also the preferred system for approach and landing. Systems have been designed [Toles, 1985] which blend together the control of pitch rate and normal acceleration.

With respect to lateral-directional control, the most prevalent control system is a roll-rate command system. This system may be designed to roll the aircraft around its own velocity vector rather than the body axis, and the reasons for this are described in the following sections.

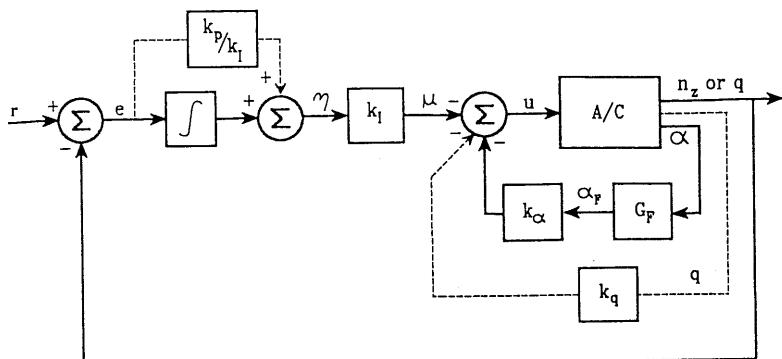


Fig. 4.5-1 Pitch-axis control augmentation system.

Pitch-Rate Control Augmentation Systems

Figure 4.5-1 is a block diagram of a pitch-axis CAS, and in this case the controlled variable is pitch rate. An integrator has been included in the forward path to make the control system Type I, thus ensuring that the aircraft will hold a zero pitch-rate trajectory when no pressure is applied to the control stick. The integrator also provides the angle-of-attack reference if alpha feedback is used and separates the inner-loop feedbacks from the controlled-variable feedback.

In the figure the dashed lines indicate two different design options. In one case pitch rate is fed back directly to the elevator actuator; in the other case the "proportional path" of the proportional-plus-integral (PI) compensator feeds the pitch-rate signal to the elevator actuator. As far as feedback signals are concerned, the two paths are equivalent (apart from a sign change). When using root-locus designs to place the closed-loop poles, the number of successive loop closures can be reduced by working with the PI compensator rather than the pitch-rate feedback (k_q). Thus an alpha-loop root-locus design can be performed first and then, based on the transfer function q/μ , the zero of the PI compensator (at $s = -k_I/k_p$) can be chosen and a second root-locus design performed to determine only k_I . This procedure avoids three successive root-locus loop closures to determine k_α , k_q , and then k_I , although a number of trial positions of the PI zero may be required.

The two dashed-line paths are not equivalent as far as the overall closed-loop transfer function is concerned. If a PI compensator is used, rather than inner-loop pitch-rate feedback, the compensator zero will be present in the closed-loop transfer function. The root-locus design technique is concerned only with placing the closed-loop poles, and this zero will often cause the closed-loop step response to have a large overshoot, even though

the closed-loop poles appear satisfactory. Therefore, in the two-loop design we shall examine the effect of finally removing the proportional feedback and setting k_q equal to $-k_p$.

The design of the pitch-rate CAS will now be illustrated by example. It will be shown that the design can be performed on the short-period dynamics, but some caution must be used.

Example 4.5-1: A Pitch-Rate CAS Design. The F-16 longitudinal dynamics corresponding to the nominal flight condition in Table 3.4-3 will be used once again. The A, B, C coefficient matrices are given in (4.4-1). These equations do not exhibit a short-period mode, but the α and q equations are only loosely coupled to v_T and θ and can be extracted as in Section 4.2. The final design will be verified on the complete dynamics. The elevator actuator and α -filter dynamics will be those used in Section 4.4.

To include both the PI compensator and inner-loop feedbacks, the state equations must be put into the form of (3.8-13):

$$\begin{aligned}\dot{x} &= Ax + Bu + Gr \\ y &= Cx + Fr \\ u &= -Ky, \quad z = Hx.\end{aligned}$$

The command (r) is a pitch-rate command, pitch rate is the performance output (z), and the outputs for feedback (y) are α_F , the compensator output η , and pitch rate q . The states for the elevator actuator, alpha filter, and error integrator will be denoted, respectively, by x_e , x_α , and x_I . Therefore, the augmented dynamic equations are given by

$$\dot{x} = \left[\begin{array}{c|c|c|c|c} A & B & 0 & 0 \\ \hline 0 & 0 & -1/\tau_e & 0 & 0 \\ 1/\tau_\alpha & 0 & 0 & -1/\tau_\alpha & 0 \\ 0 & -57.3 & 0 & 0 & 0 \end{array} \right] \begin{bmatrix} \alpha \\ q \\ x_e \\ x_\alpha \\ x_I \end{bmatrix} + \left[\begin{array}{c|c|c|c|c} 0 \\ 0 \\ 1/\tau_e \\ 0 \\ 0 \end{array} \right] u + \left[\begin{array}{c|c|c|c|c} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{array} \right] r \quad (1)$$

$$Y = \begin{bmatrix} \alpha_F \\ \eta \\ q \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 57.3 & 0 \\ 0 & -57.3k_p/k_I & 0 & 0 & 1 \\ 0 & 57.3 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ k_p/k_I \\ 0 \end{bmatrix} r \quad (2)$$

$$H = [0 \ 57.3 \ 0 \ 0 \ 0], \quad K = [k_\alpha \ k_I \ k_q],$$

where "57.3" indicates 57.29578 once again.

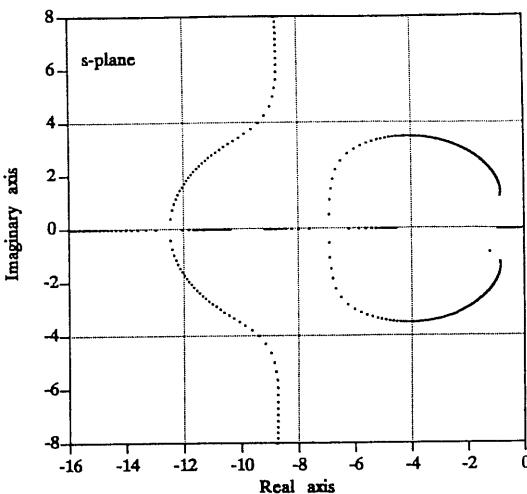


Fig. 4.5-2 Root-locus plot for the pitch-rate CAS.

The final design will be relatively slow unless the integrator pole can be moved well to the left or made to coincide with a zero. Some trial designs show that this demands a smaller amount of alpha feedback than that used in Example 4.4-1; this will be demonstrated by comparing two different values of k_α .

Consider first the situation with $k_\alpha = -0.20$; the q/μ transfer function is then given by

$$\frac{q}{\mu} = \frac{203.2(s + 10.0)(s + 1.029)}{(s + 10.38)(s + 20.13)(s + 0.8957 \pm j1.152)}. \quad (3)$$

The behavior of the outer-loop root locus with the added PI compensator can now be anticipated. As k_I is varied, the integrator pole will move toward the zero at -1.029 ; the compensator zero should be placed to the left of this zero, and the short-period poles will circle around the compensator zero. This behavior is shown in Fig. 4.5-2 for a compensator zero at $s = -3.0$.

The filter and actuator poles form a second complex pair when k_I reaches about 1.5, the integrator pole has moved to $s = -0.91$, and the short-period poles are at $s = -3.2 \pm j3.4$. Increasing k_I causes the second complex pair quickly to become less damped, while the integrator pole moves only slightly farther left. If the amount of alpha feedback is reduced, the integrator pole can be moved closer to the zero at $s = -1.029$ before the second complex pole pair appears, while maintaining a satisfactory short-period pair.

The alpha feedback was eventually reduced to $k_\alpha = -0.08$, and the compensator zero was retained at $s = -3.0$ with the intention of causing the

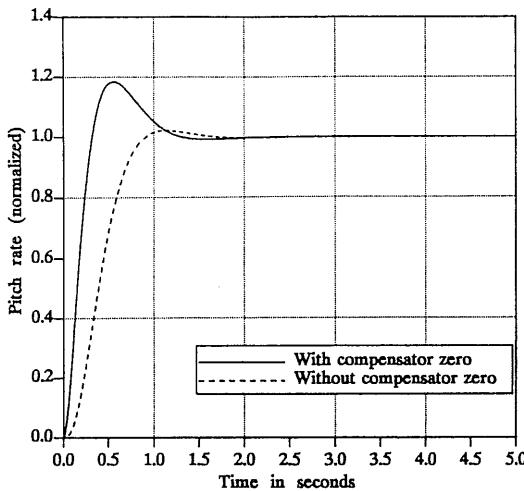


Fig. 4.5-3 Step response of the pitch-rate CAS.

short-period poles to pass near $s = -4 \pm j3$ ($\omega_n = 5$, $\zeta = 0.8$). The root locus was the same shape as Figure 4.5-2. With $k_I = 1.5$ the slow integrator pole reached $s = -1.02$ and stopped moving left, the short-period poles reached $s = -3.4 \pm j3$, and the actuator and filter poles were still short of combining to form a complex pair. This was considered to be a promising design and the closed-loop matrices were filed, ready for a step-response simulation.

The closed-loop transfer function was

$$\frac{q}{r} = \frac{101.6(s + 3.00)(s + 10.0)(s + 1.029)}{(s + 10.7)(s + 13.7)(s + 1.02)(s + 3.43 \pm j3.03)}, \quad (4)$$

and Fig. 4.5-3 (solid curve) shows the step response. This response has a fast rise time and a large overshoot (almost 20%) and would not be satisfactory for fine-tracking purposes. The effect of removing the transfer function zero at $s = -3$ can be ascertained very easily by putting $k_p = 0$ in the C and F matrices, and using the equivalent pitch-rate feedback ($k_q = -k_p = -0.5$). The closed-loop transfer function will then be identical to (4), but with no zero at $s = -3$ and a threefold increase in static loop sensitivity.

Figure 4.5-3 (dashed curve) shows the pitch-rate step response without the compensator zero. The rise time is now longer (but quite satisfactory), and the overshoot is only about 2%. Rather than attempt to refine the design, we shall move on to apply the same feedback gains to the complete longitudinal dynamics.

When the feedback gains $k_\alpha = -0.08$, $k_r = 1.5$, and $k_q = -0.5$ (i.e., $k_p = 0$) are used on the full dynamics given by (4.4-1), the closed-loop transfer function is

$$\frac{q}{r} = \frac{304.8(s + 10.0)(s + 1.027)(s + 0.02174)s}{(s + 10.75)(s + 13.67)(s + 1.016)(s + 3.430 \pm j3.032)(s + 0.02173)s}. \quad (5)$$

Observe that this transfer function contains the subset of poles and zeros given by the short-period approximation, and that the phugoid mode has degenerated to two real poles with this small amount of alpha feedback. Also, the phugoid poles are almost exactly canceled by zeros, and so would play no part in the pitch-rate response in this case.

This example illustrates some of the features of a pitch-rate CAS. If a larger amount of alpha feedback is used and the integrator pole creates an unacceptably slow component in the response, a phase-lead compensator might be used in addition to the PI compensator to speed up the response. An actual design can only be optimized by careful comparison with the flying qualities requirements, piloted simulation, and flight test. During the design process nominal designs must be performed at several points throughout the speed-altitude envelope, and the feedback gains will be a function of some "scheduling" parameters such as dynamic pressure. ■

Nonlinear Simulation of Controllers

Simulation of a control system on the nonlinear aircraft dynamics is an essential part of the complete control design process. We now describe a simple way to build the composite simulation model and illustrate the technique with the pitch-rate CAS from Example 4.5-1. The control laws, compensator dynamics, and actuator dynamics will be simulated in a separate program module from the aircraft model. This saves a great deal of code-compilation time when repeated changes to the control laws have to be made. More important, in a big organization, it allows different groups of people to be responsible for the aircraft model and the control systems and allows the integrity of the aircraft model to be maintained.

Figure 4.5-4 shows a simple controller subroutine that implements the control laws derived in Example 4.5-1 for the F-16 model. Three controller states have been created, x_{14} is the elevator actuator state, x_{15} the alpha-filter state, and x_{16} the error integrator. This subroutine calls the aircraft-model subroutine (F), and before this is done the aircraft control inputs must be set. In this case only the elevator input needs to be set by the controller. After the aircraft model has been called, the variables in the "output" common block are available for feedback, and the feedback control law can be implemented. The gains and the signs in the control law match Example 4.5-1; note that units of degrees (ALPHD and QD), rather than radians, have been used in the control law.

```

SUBROUTINE FC(TIME,X,XD)
DIMENSION X(*),XD(*)
COMMON/CONTROLS/THTL,EL,AIL,RDR,QCOM
COMMON/OUTPUT/AN,ALAT,AX,QBAR,AMACH,VT,ALPHD,THTAD,QD
C
      EL      = X(14)                      ! set elevator
      CALL    F(TIME,X,XD)                  ! call aircraft
      XD(16)= QCOM - QD                  ! integrator input
      U      = -(1.5*X(16) -.5*QD -.08*X(15)) ! control law
      XD(14)= 20.2*(U - X(14))          ! el. actuator
      XD(15)= 10.0*(ALPHD - X(15))      ! alpha filter
      RETURN
      END

```

Fig. 4.5-4 Controller subroutine for the pitch-rate CAS.

The controller subroutine must also assign the state derivatives ready for integration. In the case of \dot{x}_{16} , this is the input to the error integrator; it is formed from the pitch-rate command QCOM (that has been added to the controls common block) and the aircraft pitch rate. The other derivatives are computed from the state equation of a lag filter as given in Table 3.2-1. Controller nonlinearities (e.g., amplitude and rate limits) have not been included; these will be modeled later in the chapter. Also, in a more comprehensive model the actuators might have to be included with the aircraft model so that the effects of the aerodynamic loads and the airframe structural rigidity could be accounted for.

A check that the controller and aircraft are performing correctly can be made by linking these models and numerically linearizing the complete dynamics. Steady-state values were calculated for the controller states (as described in the example below) and the Jacobian matrices were found. The Jacobians and the closed-loop poles and zeros, found in this way, agreed with those obtained in Example 4.5-1 to six or seven digits. The controller subroutine will be used to perform a flight simulation in the following example.

Example 4.5-2: Pitch-CAS Nonlinear Simulation. This simulation will be performed on the 6-DOF F-16 model with the lateral-directional dynamics left uncontrolled. If the lateral-directional states and inputs are accurately initialized with their steady-state values, those dynamics will remain unexcited for at least a few minutes of simulation, provided that the aircraft is not flown into a stall. In addition, accurate initialization allows the longitudinal dynamics to be exercised without waiting for an initial transient to die out.

The alpha-filter state should be initialized with the trim value of alpha, and the elevator state with the trim value of elevator deflection (both in degrees). The initial value of the error integrator state can be calculated as follows. From Fig. 4.5-1 we see that the steady-state (no integrator input)

elevator deflection is given by

$$EL = -k_I x_I - k_q Q - k_\alpha \alpha_F.$$

When the trim values are inserted in this equation, the trim value of the integrator output is found to be $x_I = 0.6186446$. The original trim-data file can now be augmented with the initial conditions for the three additional states.

A simulation can be flown by linking the time-history program with the controller and aircraft subroutines and using a discrete-time command subroutine to exercise the controller. Figure 4.5-5 shows the code for the driver program and the discrete-time command routine. The aircraft is given full throttle and a pitch-up command at $t = 10$ s; the pitch rate command has been adjusted to bring the aircraft vertical at $t = 20$ s. Another pitch-rate command ($10^\circ/\text{s}$) is applied at $t = 50$ s. Figure 4.5-6 shows the aircraft trajectory produced by these commands, viewed in the altitude versus distance-north plane. At $t = 50$ s the airspeed has fallen to about 300 ft/s and the altitude is approximately 16,000 ft; therefore, the flight conditions are greatly different from the controller design conditions (sea level and 502 ft/s). Pitch rate as a function of time is shown in Fig. 4.5-7, and the pitch-rate response has a large overshoot at $t = 50$ s because of the off-nominal design conditions. The aircraft dynamics change with flight conditions as described in Section 4.2, and in a practical controller design, the gain coefficients would be programmed as functions of dynamic pressure (other possible *gain-scheduling* variables are Mach number and alpha).

```

PROGRAM FLY
EXTERNAL FC,D
COMMON/PARAM/XCG
XCG= .35
CALL TRESP(FC,D)           ! generic time-response program
END
C
SUBROUTINE D(TIME,TS,X,XD,DFLAG)
DIMENSION X(*),XD(*)
COMMON/CONTROLS/THTL,EL,AIL,RDR,QCOM
LOGICAL DFLAG
IF (TIME .LT. 10.0) THEN
  RETURN
ELSE IF (TIME .LT. 20.0) THEN
  QCOM= 8.65
  THTL= 1.0
ELSE IF (TIME .LT. 50.0) THEN
  QCOM= 0.0
ELSE
  QCOM= 10.0
  THTL= 1.0
END IF
RETURN
END

```

Fig. 4.5-5 Driver programs for nonlinear simulation of pitch-rate CAS.

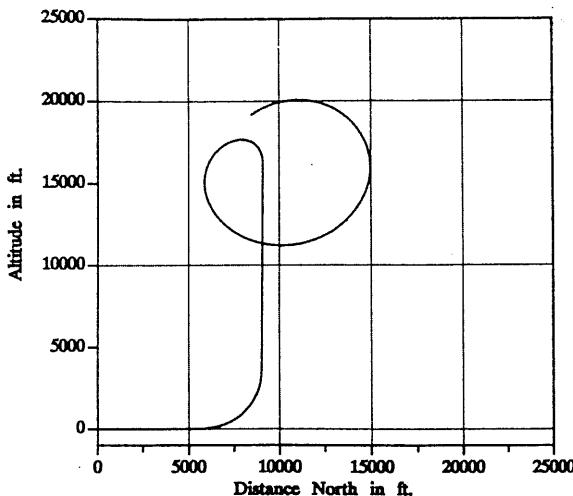


Fig. 4.5-6 Aircraft trajectory with the pitch-rate CAS.

In this example no attention was paid to the control surface deflections and rates; these limitations are taken into account in later examples. The time of flight was 100 s for the trajectory shown, and the lateral-directional dynamics did not become significantly involved in the motion. The sideslip angle peaked up to about 0.1° after the pitch-over command at $t = 50$ s but then returned to very small values. The angle of attack reached a peak of approximately 15° at $t = 55$ s.

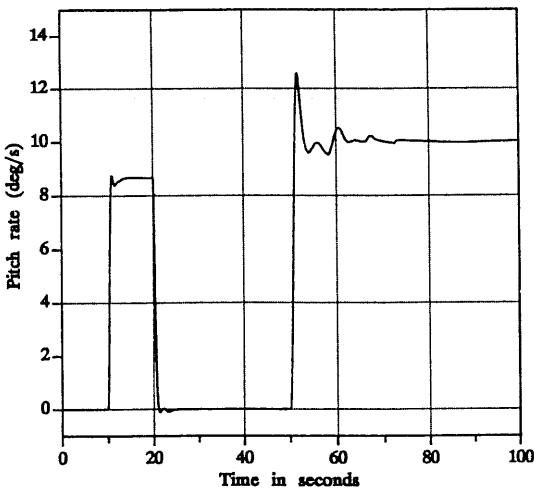


Fig. 4.5-7 Pitch-rate response along the aircraft trajectory.

Normal Acceleration Control Augmentation Systems

It was shown in Section 1.3 that an accelerometer, fastened to the aircraft body, measures the acceleration produced by the sum of the aerodynamic and thrust forces, plus tangential and centripetal acceleration components due to angular motion. If the accelerometer has its sensitive axis aligned with the aircraft z -axis, it will measure the so-called *normal acceleration*. Assume, for simplicity, that the accelerometer is on the x -axis, at a distance x_a forward of the aircraft cg. According to (1.3-14c) the expression for the measured normal acceleration, in g 's, is

$$A_z = (-F_z/m + x_a \dot{Q})/g, \quad (4.5-1)$$

where F_z is the z -component of the aerodynamic and thrust forces at the cg and \dot{Q} is the pitch acceleration. The force component F_z depends chiefly on the lift force, which is in turn dependent on the angle of attack. Therefore, normal acceleration feedback can be used to stabilize a longitudinally unstable aircraft in the same manner as alpha feedback.

A linear equation for the *incremental* normal acceleration, a_n , can be found algebraically (see McRuer et al. [1973]), but for our purposes, it is more convenient (and accurate) to use numerical linearization. Equation (4.5-1) is included in the nonlinear F-16 model, and if, for example, the model is linearized at the nominal flight condition in Table 3.4-3, the output equation for normal acceleration at the cg ($x_a = 0$) is found to be

$$n_z \equiv a_n = 0.003981v_T + 15.88\alpha + 1.481q + 0.03333\delta_e. \quad (4.5-2)$$

Both symbols n_z and a_n are commonly used to denote incremental normal acceleration in units of g 's; α and q are in radian units, and δ_e is in degrees.

The incremental normal acceleration depends on the v_T , α , and q (the quantities that define the longitudinal aerodynamic forces) and on elevator deflection, which produces aerodynamic forces directly. This direct feed term was also noted in Example 3.6-1 and leads to a normal acceleration transfer function of relative degree zero. Note that the normal acceleration does not depend on the pitch attitude variable θ , since gravity does not enter into (4.5-1).

The elevator-to-normal-acceleration transfer function corresponding to (4.5-2) can be found from the Jacobian matrices, and is

$$\frac{n_z}{\delta_e} = \frac{0.03333(s - 0.003038)(s + 0.01675)(s + 6.432)(s - 13.14)}{(s - 0.09756)(s + 1.912)(s + 0.1507 \pm j0.1153)}. \quad (4.5-3)$$

This transfer function has the same poles that were noted in Section 4.4 and two zeros close to the origin, as noted there. The interesting features of the transfer function are the two "short-period zeros" at $s = -6.432$ and $s = 13.14$. Because of the NMP zero at $s = 13.14$, the normal acceleration response to a negative step elevator-command (aircraft nose-up) will be an initial negative acceleration, quickly followed by the expected positive normal acceleration.

The physical explanation for the non-minimum-phase behavior is that when the elevator control surface is deflected trailing edge upward to produce a positive normal acceleration, this creates a downward increment of force on the tail. The result is that the cg of the aircraft may drop momentarily during the pitch-up, so the normal acceleration may briefly become negative before it builds up positively. At the pilot's station ahead of the cg, the normal acceleration also depends on the pitch acceleration about the cg, so only a positive normal acceleration may be felt. Table 4.5-1 shows the elevator-to-normal-acceleration transfer function zeros for a range of accelerometer positions, from the cg forward. The zeros close to the origin do not change significantly from the positions given in (4.5-3), and only the static loop sensitivity and short-period zeros are shown.

Table 4.5-1 shows that as the accelerometer position is moved forward, the NMP zero moves out toward infinity and the static loop sensitivity decreases, thus keeping constant the transfer function dc gain. Eventually, the static loop sensitivity changes sign and a zero comes in from infinity along the negative real axis, finally combining with the other real zero to form a complex pair. At a position near 6.1 ft forward of the cg the non-minimum-phase effect disappears, and this point corresponds to an "instantaneous center of rotation" when an elevator input is suddenly applied.

In modern fighter aircraft the accelerometer is placed close to the pilot's station, so that the controlled variable is the same as the physically sensed quantity. Note that in the case of the real F-16 aircraft, the pilot's station is approximately 15 ft ahead of the cg and is therefore not close to the instantaneous center of rotation. It is also important to place the accelerome-

TABLE 4.5-1. Transfer Function Zeros Versus Accelerometer Position

x_a (ft)	Static-Loop-Sensitivity and Numerator Factors
0	$0.03333(s + 6.432)(s - 13.14)$
5	$0.006042(s + 9.171)(s - 50.82)$
6	$0.0005847(s + 10.68)(s - 450.7)$
6.1	$0.00004005(s + 10.90)(s - 6448.2)$
7	$-0.004872(s + 14.73)(s + 39.23)$
15	$-0.04852(s + 3.175 \pm j6.925)$

ter close to a node of the most important fuselage bending mode. If this is not done, structural oscillations will be coupled into the rigid-body control system and may degrade the flying qualities or even cause an "aeroservoelastic" limit-cycle oscillation (see, e.g., AFWAL-TR-84-3105). Inevitably, the design of a normal-acceleration control system to achieve good flying qualities is difficult and requires a good deal of refinement based on flight test results. A control system that has a good normal acceleration step response may have a pitch-rate response with a very large overshoot, and conversely, a reduction in the pitch-rate overshoot may lead to a sluggish normal acceleration response. The C^* criterion is appropriate for initial evaluation of the control system, since it is based on a blend of normal acceleration at the pilot's station and pitch rate.

Finally, note that an accelerometer is an internal (within the fuselage) sensor, with higher reliability and lower noise than the external alpha sensor. However, both accelerometers and alpha sensors are typically employed on modern fighter aircraft, and this reduces the dependence on the alpha sensor. A disadvantage of normal acceleration feedback is that the gain of the transfer function (4.5-3) varies widely with dynamic pressure [this can be deduced from (4.5-1)]. Accelerometer noise may become a problem if, at low dynamic pressure, the gain has to be greatly boosted to achieve a desirable closed-loop response. We shall now investigate the features of the normal acceleration CAS by means of a design example.

Example 4.5-3: A Normal Acceleration CAS Design. The configuration shown in Fig. 4.5-1 will again be used, with the controlled variable changed to normal acceleration and the alpha feedback removed. The same dynamics as Example 4.5-1 will be used, but an output equation for normal acceleration must be determined. Therefore, using the numerical linearization program on the F-16 model with the accelerometer 15 ft forward of the cg (i.e., at the pilot's station), and the nominal flight condition from Table 3.4-3, the output equation is found to be

$$n_z = 0.0039813v_T + 16.262\alpha + 0.97877q - 0.048523\delta_e. \quad (1)$$

The state equations (3.8-13) do not allow for a plant direct-feed term as in equation (1). This does not create any difficulty; direct feed terms will always disappear when the plant is augmented with control surface actuator dynamics (assumed to comprise a lag model with no direct feed).

In (1) the dependence on v_T is quite weak, so the states v_T and θ will be dropped, with the final results checked on the complete dynamics. Therefore, following the same lines as Example 4.5-1, the augmented coefficient

matrices are

$$\dot{x} = \begin{bmatrix} A & B \\ 0 & -\frac{1}{\tau_e} \\ -16.26 & -0.9788 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{\tau_e} \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{\tau_e} \end{bmatrix} r$$

$$y = \begin{bmatrix} q \\ \eta \end{bmatrix} = \begin{bmatrix} 0 & 57.29578 & 0 & 0 \\ -16.26 & -0.9788 & 0.04852 & \frac{k_I}{k_p} \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \quad (2)$$

$$H = [16.26 \quad 0.9788 \quad -0.04852 \quad 0],$$

$$K = [k_q \quad k_p].$$

Note that the PI compensator equation has been put in the form $\eta = (e + x_I k_I / k_p)$, so that it is not necessary to multiply all of the normal acceleration coefficients by k_p/k_I .

The plant transfer function from elevator actuator to normal acceleration is found to be

$$\frac{n_z}{u} = \frac{-0.9802(s + 3.179 \pm j6.922)}{(s + 20.20)(s + 1.911)(s + 0.1850)}. \quad (3)$$

The effect of the inner-loop pitch-rate feedback is to speed up the two slow poles, and at quite low gain the pole from $s = -1.911$ combines with the actuator pole to form a complex pair. Speeding up these poles is desirable for a fast time response, but as noted previously, the amount of pitch-rate feedback is limited by practical considerations (pickup of structural noise). The value $k_q = -0.4$ (degrees of elevator deflection per degree per second of pitch rate) is in line with our past experience (-0.25 to -0.5) and leads to the following forward-path transfer function for Figure 4.5-1:

$$\frac{n_z}{\mu} = \frac{0.9802(s + 3.179 \pm j6.922)}{(s + 13.78)(s + 7.661)(s + 0.8601)}. \quad (4)$$

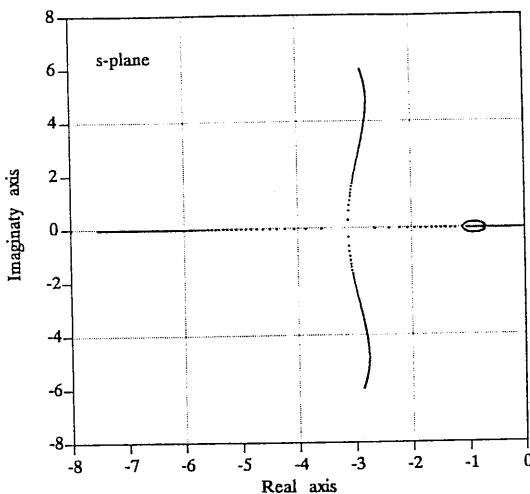


Fig. 4.5-8 Root-locus plot for the normal acceleration CAS.

The outer-loop root locus with the added PI compensator can now be anticipated. The compensator pole (at $s = 0$) will combine with the pole at $s = -0.8601$ to form a complex pair, and these poles will move toward the complex zeros of (4). This behavior will be modified depending on the position of the compensator zero. If the compensator zero is well to the left of $s = -0.8601$, these branches will be deflected only slightly to the left before landing on the complex zeros. At the same time the pole at $s = -7.661$ will move toward the compensator zero, creating a potential slow-pole problem. The complex zeros are not well damped and it is difficult to achieve fast, well-damped, complex poles together with a fast real pole.

The alternative is to place the compensator zero close to the pole at $s = -0.8601$ so that this pole is effectively canceled. The loci for the short-period poles will then break away from the real axis somewhere closer to the pole at $s = -7.661$ before proceeding to the complex zeros. Some trial and error shows that this approach leads to a better time response, and it will be followed here. In practice, the sensitivity of the poles to gain variations, noise pickup, and possible advantages of additional compensator poles and zeros would have to be considered.

Figure 4.5-8 shows the outer-loop root locus (i.e., k_p varied) when the PI compensator zero is placed at $s = -0.9$ (to demonstrate that exact cancellation is not required). The effect of the imperfect cancellation is visible near $s = -0.9$; and the locus of the short-period poles shows that satisfactory damping and natural frequency can be achieved without the use of an additional lead compensator.

The short-period poles should be made well damped because the compensator zero can be anticipated to cause an overshoot in the closed-loop step response. When the root locus is calibrated with a few values of k_p , a value $k_p = 5$ puts the short-period poles at $s = -3.00 \pm j2.18$ ($\omega_n = 3.7$, $\zeta = 0.81$). The closed-loop transfer function is then

$$\frac{n_z}{r} = \frac{4.901(s + 0.9000)(s + 3.179 \pm j6.922)}{(s + 20.28)(s + 0.9176)(s + 3.000 \pm j2.180)}. \quad (5)$$

Figure 4.5-9a shows the closed-loop step response corresponding to this transfer function. The response is fast and well damped; the initial rate of rise is particularly fast because of the pitch acceleration component of the response. The rate limitations of the elevator actuator would modify this response slightly, but this is not considered important for the purposes of this example. The associated pitch-rate response (not illustrated) shows an overshoot of approximately 100%, but the normalized C^* response shown in Fig. 4.5-9b falls almost exactly in the middle of the level 1 envelope (see Section 4.3). The C^* values were computed by adding the component $12.4q$ to the normal acceleration output equation in the closed-loop Jacobian matrices. An initial time-response run was performed so that the steady-state value of C^* could be determined for use in normalizing the response.

The closed-loop transfer function obtained by applying the same feedback gains to the complete longitudinal dynamics (i.e., phugoid included) is

$$\frac{n_z}{r} = \frac{4.901(s + 0.900)(s + 3.175 \pm j6.925)(s + 0.01685)(s - 0.003139)}{(s + 20.28)(s + 0.9194)(s + 3.000 \pm j2.186)(s + 0.01637)(s - 0.003219)}. \quad (6)$$

Notice that this transfer function contains, to a very good approximation, the poles and zeros of (5), thereby justifying the use of the short-period approximation. In (6) the phugoid mode is degenerate (two real poles) and one pole is unstable, whereas in Section 4.4 a stable phugoid was achieved with the basic stability augmentation system. This is because the normal acceleration equation, (1), contains a component due to v_T , and this component is being fed back in a positive sense (positive δ_e gives positive v_T). The phugoid mode is almost canceled by the transfer function zeros in this case, and the unstable pole is very slow. Nevertheless, an unstable phugoid is undesirable.

The problem is avoided in practice by retaining some inner-loop alpha feedback and using less gain in the normal acceleration loop. An alternative possibility is to modify (4.5-1) to remove the gravity component from the normal acceleration feedback. The quantity A'_n given by [see (2.4-2)]

$$A'_n = A_n - g \cos \theta \cos \phi, \quad (7)$$

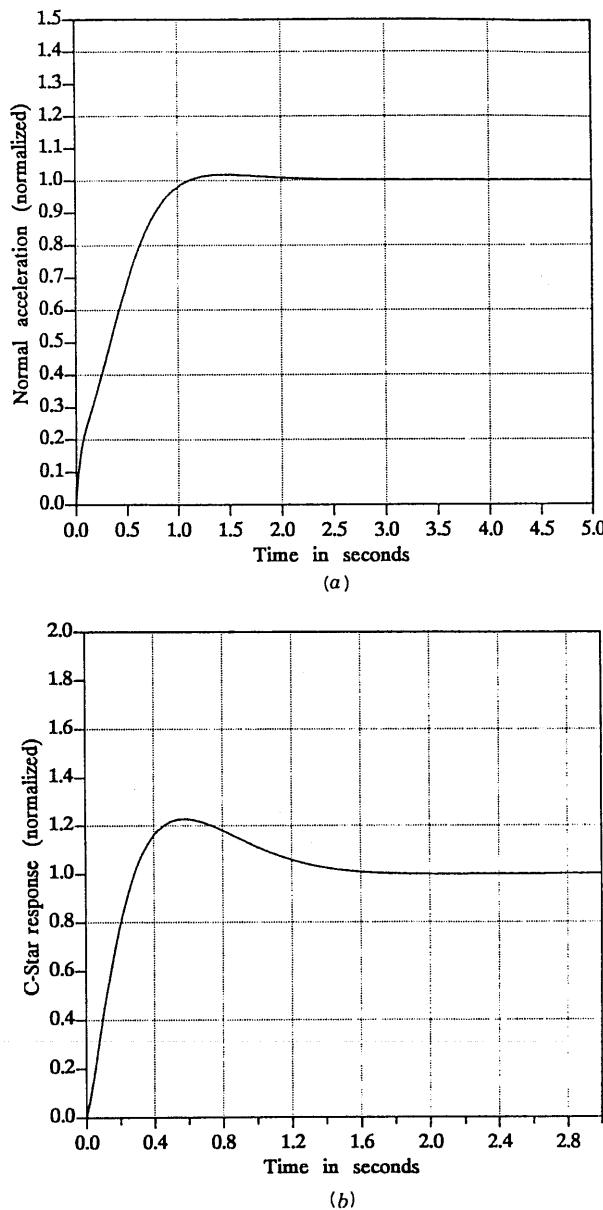


Fig. 4.5-9 (a) Normal acceleration step response and (b) C-star response.

can easily be calculated in the flight control computer if attitude information is provided. When (7) is linearized, the incremental normal acceleration contains a component due to θ that will be in the correct sense to provide a stabilizing effect on the phugoid mode (Problem 4.5-2). This scheme has the practical advantage that zero g 's now corresponds to any nonturning flight condition and, for example, the aircraft will hold a steady climb with no control stick deflection.

Lateral–Directional Control Augmentation

The roll/yaw stability augmentation system described in Section 4.4 is adequate for most aircraft, but for aircraft that must maneuver rapidly at high angles of attack, a more refined lateral–directional control augmentation system is required. The lateral aerodynamic control surfaces (ailerons and differential elevator) tend to cause the aircraft to roll about its longitudinal axis, and at high alpha, this can lead to some highly undesirable effects.

Consider the effect of a rapid 90° body-axis roll at high alpha. It is easy to visualize that the angle of attack will be converted immediately, and almost entirely, to a sideslip angle. This is referred to as *kinematic coupling* of alpha and beta. Because of this rapid elimination of the angle of attack, the body-axis roll is counterproductive. The most important purpose of a roll is to initiate a turn, which is then achieved by using angle of attack to produce the lift that will subsequently generate the required centripetal acceleration.

The sideslip created by kinematic coupling is referred to as *adverse sideslip* because it will tend to oppose the roll (remember that $C_{l\beta}$ is normally negative; a right roll will generate positive beta through kinematic coupling and hence a negative rolling moment). The sideslip will exist until the aircraft has yawed into the wind once more, and then if the angle of attack must be reestablished, the result will be an inefficient turn entry. Most modern fighters therefore use automatic control systems designed to roll the aircraft about the stability x -axis, thus maintaining the initial angle of attack.

Finally, large sideslip angles are undesirable for several important reasons. The effectiveness of the aerodynamic control surfaces may be greatly reduced; directional stability may be lost so that, in some cases, aircraft have been known to tumble (end over end). Even if directional stability is maintained, a large sideforce may develop which may possibly break the vertical tail.

Another important effect that may occur during a roll is *inertia coupling*. Suppose that the aircraft has been designed to roll around the stability x -axis, with no sideslip. Then the transformations in Section 2.3 can be used to determine the body-axes roll and yaw rates that result in a stability-axes roll rate, P_s , with zero yaw rate, R_s . The relevant equations are

$$P_s = P \cos \alpha + R \sin \alpha \quad (4.5-4a)$$

$$0 = R_s = -P \sin \alpha + R \cos \alpha \quad (4.5-4b)$$

or

$$R = P \tan \alpha. \quad (4.5-4c)$$

When alpha is positive R and P must have the same sign, and if alpha is large, body-axes yaw rates comparable to the body-axes roll rate must be generated. Therefore, in a rapid high-alpha roll, gyroscopic (inertia coupling) effects will generate a significant body-axes pitching moment. Equations (1.3-22) described the inertia-coupling effect in the case when the cross-products of inertia could be neglected, and using these simplified equations, the pitching moment, M_{IC} , due to inertia coupling is given by

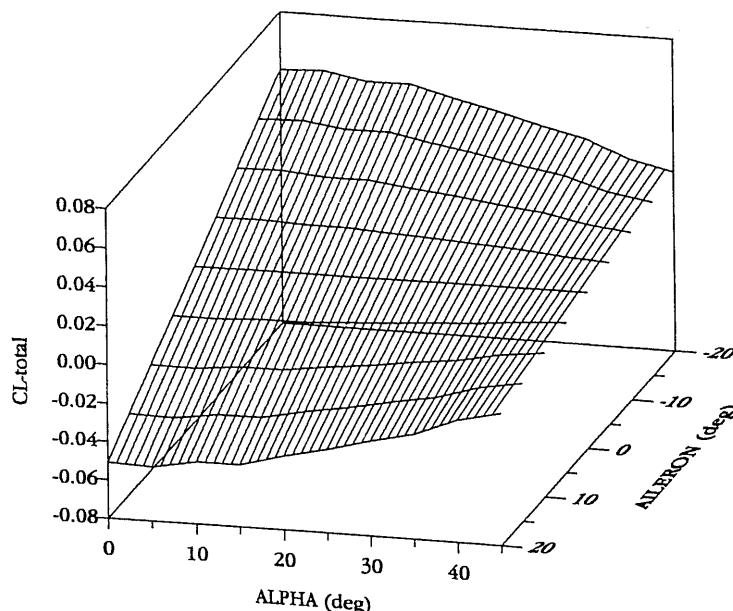
$$M_{IC} = \dot{Q}J_Y = (J_Z - J_X)PR. \quad (4.5-5)$$

For modern fighter aircraft with stubby wings and engine(s) on or near the longitudinal axis, the moment of inertia J_X is usually small compared to J_Z (while J_Z and J_Y are comparable in magnitude). Therefore, a rapid roll (right or left) about the stability x -axis, at large positive alpha, can produce a strong nose-up pitching moment. To avoid a "pitch departure," the pitch-axis control augmentation system must cause the horizontal tail to generate an opposing aerodynamic moment. At high alpha it may be difficult to obtain the necessary aerodynamic pitching moment because of the horizontal tail stalling. Even when adequate pitching moment is available, the required yawing moment may be unachievable because the rudder is blanketed by the wings. Conventional aircraft therefore have greatly degraded roll response at high alpha, and furthermore, the control systems must often be designed to limit the commanded roll rate to avoid a pitch departure.

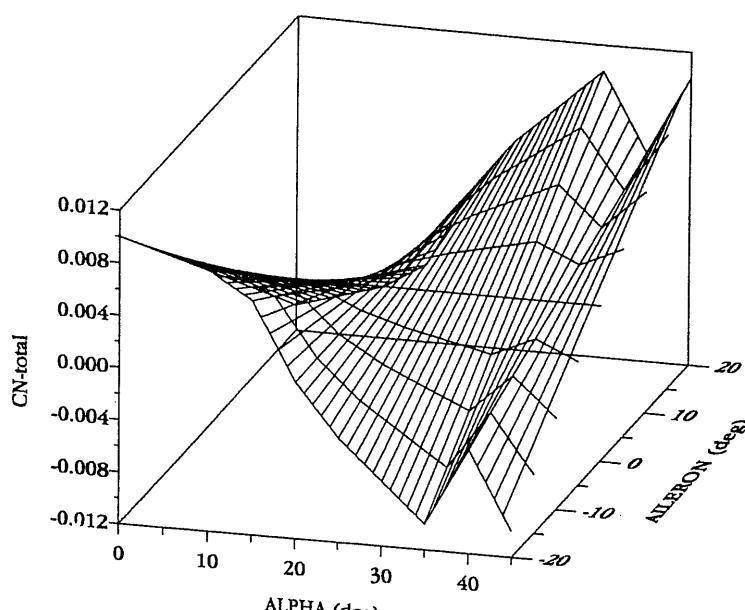
Figures 4.5-10 show some aerodynamic-data plots, obtained from our F-16 model, that are relevant to lateral-directional control at high alpha. The plot of rolling moment against aileron deflection and alpha (Fig. 4.5-10a) shows that the aileron effectiveness begins to decrease at about 5° angle of attack and has degraded by about 60% at 30° alpha. Sideslipping would cause a further reduction in effectiveness.

Figure 4.5-10b illustrates the yawing moment created by the ailerons. At low angles of attack this moment is *proverse*; that is, it tends to yaw the aircraft into the turn. At higher alpha the yawing moment becomes *adverse*, and the combination of rolling with adverse yaw will create sideslip unless the rudder is used to overcome the adverse yaw and coordinate the turn. Note that at high alpha, the yawing moment coefficient of the ailerons is comparable to the rolling moment coefficient.

The yawing and rolling moment coefficients for the rudder are plotted in Fig. 4.5-10c and d, respectively. They show that the effectiveness of the rudder begins to fall off above about 25° to 30° angle of attack. Overall, the plots show that the ailerons and rudder are not "pure" controls; there are strong cross-control effects between the lateral and directional dynamics.

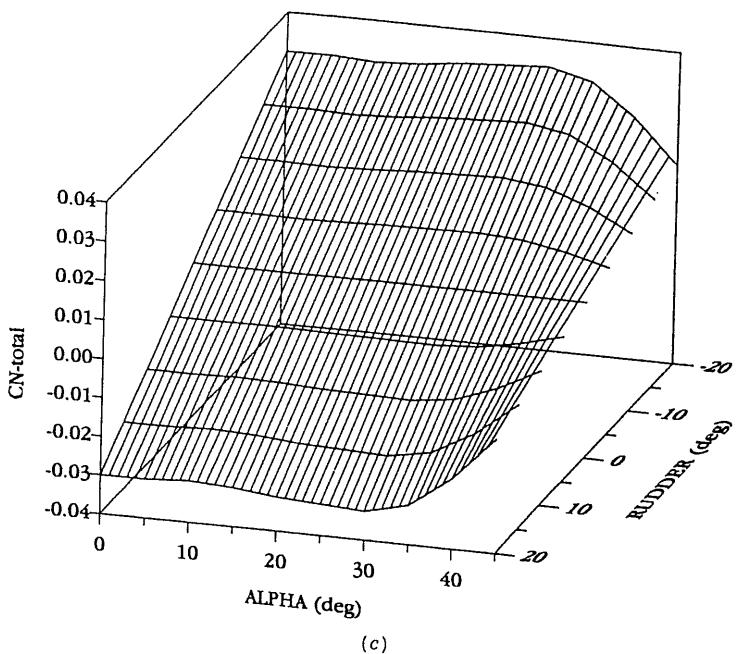


(a)

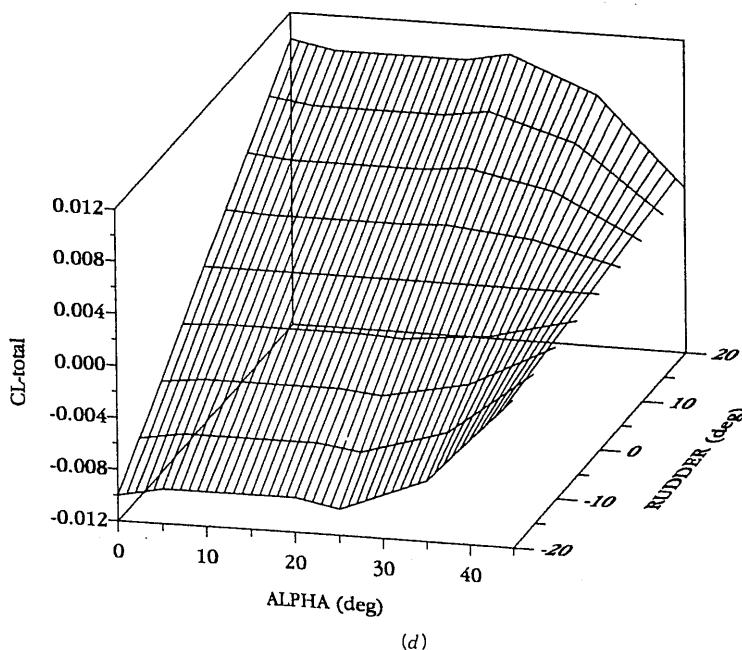


(b)

Fig. 4.5-10 (a) F-16 rolling moment versus ailerons and alpha. (b) F-16 yawing moment versus ailerons and alpha.



(c)



(d)

Fig. 4.5-10 (continued) (c) F-16 yawing moment versus rudder and alpha. (d) F-16 rolling moment versus rudder and alpha.

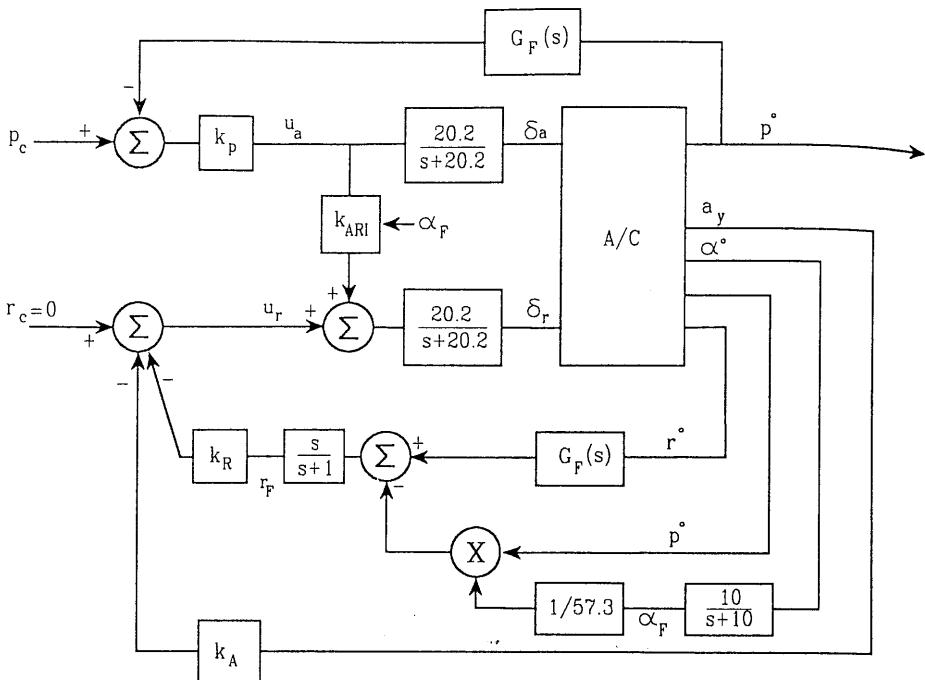


Fig. 4.5-11 Lateral-directional CAS.

Figure 4.5-11 illustrates the essential features of a lateral-directional CAS for a modern fighter aircraft; compensation networks, limiters, and so on, are added as necessary. The aileron-control channel is the same as that shown in Fig. 4.4-4 for the lateral-directional SAS, except that the aileron-actuator input now has a cross-connection to the rudder actuator via an alpha-dependent gain (also Mach dependent in general). This cross-connection, known as the *aileron-rudder interconnect* (ARI), may be implemented hydromechanically on some aircraft or electrically on others. Its purpose is to provide the component of yaw rate necessary to achieve a stability-axis roll.

The ARI "gain" must be determined, as a function of alpha and Mach number, to achieve the amount of yaw rate required to satisfy the constraint equation (4.5-4c). The gain is typically estimated from the known aerodynamic data, and adjusted using nonlinear simulation. We can avoid this by incorporating the constraint $R = P \tan \alpha$ in our steady-state trim program, and trimming the aircraft for a "steady-state" roll (see Section 3.4). Table 4.5-2 shows an abridged set of trim data for different roll rates and two different pitch rates; the angular units are all in degrees. The trim program has driven the lateral acceleration, A_y (along the body y -axis), essentially to zero (about 10^{-6} g's actually), with a small sideslip angle.

The table shows that angle of attack is almost independent of the roll rate, but it is not independent of pitch rate. Therefore, the second half of the table

TABLE 4.5-2. Trim Conditions for Determining ARI Gain^a

	<i>P</i>	α	β	A_y	ail	rdr	rdr/ail
$Q = 0$	10	2.12	-0.012	0	-0.813	0.269	-0.331
	20	2.12	-0.023	0	-1.63	0.537	-0.329
	45	2.15	-0.050	0	-3.66	1.20	-0.328
	60	2.17	-0.065	0	-4.88	1.60	-0.328
	90	2.22	-0.091	0	-7.32	2.38	-0.325
	120	2.28	-0.112	0	-9.76	3.14	-0.322
	180	2.39	-0.139	0	-14.6	4.63	-0.317
	10	6.53	0.012	0	-0.835	-0.0948	0.114
$Q = 5$	20	6.52	0.022	0	-1.67	-0.195	0.117
	45	6.46	0.046	0	-3.76	-0.445	0.118
	60	6.41	0.058	0	-5.01	-0.595	0.119
	90	6.28	0.071	0	-7.52	-0.897	0.119
	180	5.87	0.048	0	-15.0	-1.83	0.122

^a $c_g = 0.35\bar{c}$, $V_T = 502$, $h = 0$, $\bar{q} = 300$, $M = 0.450$.

is for a pitch rate of $5^\circ/\text{s}$ and serves to provide data for a higher alpha condition (6.5°). Pitch rates of 10, 15, and $20^\circ/\text{s}$ were used to provide additional data; maximum engine thrust is reached in between the last two conditions. The table indicates that for a stability-axis roll under the conditions shown (i.e., $M = 0.45$, etc.), the required ratio of rudder deflection to aileron deflection is -0.33 at $\alpha = 2.2^\circ$ and 0.12 at $\alpha = 6.5^\circ$. Using the additional data for other angles of attack showed that the ratio of rudder to aileron deflection, k_{ARI} , was a good fit to the straight line:

$$k_{ARI} = 0.13\alpha - 0.7 \quad (\alpha \text{ in degrees}). \quad (4.5-6)$$

In a practical design the effect of Mach number must also be determined, and a two-dimensional lookup table might be constructed for k_{ARI} . Because of time and space limitations, (4.5-6) will be used here, and the design example will not involve large variations of Mach number.

The ARI alone would be an open-loop attempt to achieve a stability-axis roll, and to eliminate this deficiency, feedback control is used to drive the lateral acceleration to zero (as in Table 4.5-2). Figure 4.5-11 shows how lateral acceleration is fed back and compared with a null reference input, and the error signal is used to drive the rudder actuator. This is also known as a *turn-coordination* scheme and can be used in autopilot systems to respond to radio-navigation steering signals or relieve the pilot of the need to coordinate turns.

Like the normal acceleration CAS, lateral acceleration feedback suffers from a wide variation of sensitivity. High values of feedback gain are needed at low speed, and this may cause problems with accelerometer noise. At low speed ($M < 0.3$) sideslip-angle feedback is normally used instead of lateral

acceleration, but has the disadvantage that a beta sensor is less reliable than an accelerometer.

The inner feedback loop in the rudder channel provides dutch roll damping by feeding back an approximation to the stability-axis yaw rate [equation (4.5-4b)] to the rudder. Thus the filtered alpha signal, converted to radians (as necessary), is used as an approximation to $\sin \alpha$, multiplied by the roll rate and subtracted from the yaw rate. The stability-axis yaw rate is washed out so that it operates only transiently and does not contribute to a control error when a steady yaw rate is present. Note that, according to (2.4-23), the yaw-rate feedback is equivalent to beta-dot feedback when phi and beta are zero or small.

When necessary the pilot can still sideslip the airplane, because rudder inputs are applied directly to the rudder actuator. The control system will tend to reject this disturbance input, so the desirable effect of limiting the sideslipping capability will be achieved.

A practical lateral-directional CAS, based on the concept above, will be a complex system involving gain scheduling (with angle of attack and dynamic pressure or Mach), multipliers and limiters, and discrete switching (to change the control laws automatically at the alpha limits). It is a particularly good illustration of the fact that aircraft control systems incorporate many nonlinear and time-varying effects, and that the "tuning" of a design is done by trial and error, using computer simulation as a tool, together with piloted simulation and flight tests. An example of a lateral-directional CAS design based on Fig. 4.5-11 will now be given.

Example 4.5-4: A Lateral-Directional CAS Design. There is an expedient alternative to augmenting the aircraft Jacobian matrices with the controller dynamics shown in Fig. 4.5-11. This is to simulate the controller and linearize the complete dynamics of controller plus aircraft. The controller subroutine is shown in Fig. 4.5-12. The elevator, aileron, and rudder actuators all use simple-lag models, with states X_1 , X_4 , and X_5 , respectively. The aircraft state vector is appended to the controller state vector as shown in the aircraft subroutine call (i.e., the aircraft states are now automatically numbered from seven onward). The pitch-axis controller uses the pitch-rate control law derived in Example 4.5-1, and the integrator and alpha-filter states are, respectively, X_3 and X_2 . Alternative lines of code are shown such that the roll-rate feedback loop can be broken and the command PCOM applied directly to the aileron actuator. Equation (4.5-6) is used for the ARI gain, and a 1.0-s washout time constant has been used (state X_6). An alternative line of code is shown for applying the yaw-rate command RCOM directly to the rudder actuator when linearizing the dynamics.

The nominal flight condition of Table 3.4-3 was used, and the six compensator states were included in the trim data file. The actuator states must be set to the trimmed values of the corresponding aircraft controls, and the alpha-filter state to the value of alpha in degrees. The other controller states can be set to zero since the rest of the controller is linear. Linearization of

```

SUBROUTINE FC(TIME,X,XD)
DIMENSION X(*), XD(*)
REAL M
COMMON/CONTROLS/THTL,EL,AIL,RDR,PCOM,QCOM,RCOM
COMMON/OUTPUT/AN,AY,AX,QBAR,M,ALPHA,BETA,PHID,THTAD,
& PD, QD, RD, RF
C
      EL = X(1)
      AIL= X(4)
      RDR= X(5)
      CALL F(TIME,X(7),XD(7))           ! aircraft dynamics
      XD(3) = QCOM - QD               ! error integrator
      U= -(1.5*X(3) - .5*QD - .08*X(2)) ! pitch control law
      XD(1)= 20.2*(U-X(1))           ! elevator actuator
      XD(2)= 10.0*( ALPHA - X(2) )   ! alpha filter
      UA = -.2*(PCOM-PD)             ! roll control law
C
      UA= PCOM
      XD(4)= 20.2*( UA - X(4) )       ! aileron actuators
      ARI = (0.13*X(2) - 0.7)*UA     ! ARI
      RS = RD - PD*X(2)/57.3         ! yaw-rate feedback
      XD(5)= RS - X(6)               ! washout
C
      RF= XD(6)
      ERR= RCOM + .8*XD(6) + 10.0*AY ! yaw control law
C
      ERR= RCOM
      XD(5)= 20.2*( ERR + ARI - X(5) ) ! rudder actuator
      RETURN
      END

```

Fig. 4.5-12 Pitch rate and lateral-directional controller subroutine.

the aircraft and compensator dynamics was performed for this flight condition, selecting only the lateral-directional states, the inputs PCOM and RCOM, and the outputs AY, PD, and RF (from the output common block of the aircraft model). The open-loop Jacobian matrices are given by

$$\begin{aligned}
A &= \begin{bmatrix} \delta_a & \delta_r & x_w & \beta & \phi & p & r \\ -20.20 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -20.20 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.0 & 0 & 0 & -2.1147 & 57.296 \\ 0.00029506 & 0.00080557 & 0 & -0.32201 & 0.064040 & 0.036382 & -0.99167 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0.036928 \\ -0.73331 & 0.13154 & 0 & -30.649 & 0 & -3.6784 & 0.66461 \\ -0.031865 & -0.062017 & 0 & 8.5395 & 0 & -0.025435 & -0.47637 \end{bmatrix} \\
B &= \begin{bmatrix} u_a & u_r \\ 20.2 & 0 \\ -8.5864 & 20.2 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \\
C &= \begin{bmatrix} 0.0046043 & 0.012571 & 0 & -5.0249 & 0 & -0.0081179 & 0.11932 \\ 0 & 0 & 0 & 0 & 0 & 57.296 & 0 \\ 0 & 0 & -1.0 & 0 & 0 & -2.1147 & 57.296 \end{bmatrix} \begin{matrix} a_y \\ p \\ r_F \end{matrix}
\end{aligned}$$

When the poles and zeros of the principal transfer functions are checked it is found, as expected, that the effect of the ARI is to move only the transfer function zeros. The open-loop transfer functions from actuator input to roll rate and to lateral acceleration, with the ARI connected, are given by

$$\frac{p}{u_a} = \frac{-913.4(s + 0.4018 \pm j2.945)(s - 0.002343)}{(s + 0.4235 \pm j3.063)(s + 3.615)(s + 0.01433)(s + 20.20)} \quad (1)$$

$$\frac{a_y}{u_r} = \frac{0.2539(s - 4.191)(s + 4.092)(s + 3.575)(s + 0.005030)}{(s + 0.4235 \pm j3.063)(s + 3.615)(s + 0.01433)(s + 20.20)}. \quad (2)$$

The dutch roll poles are very lightly damped, and are virtually canceled out of the roll-rate transfer function by a pair of zeros. The lateral acceleration transfer function contains an NMP zero, which is analogous to the NMP zero in the elevator-to-normal-acceleration transfer function. In this example no added compensation will be needed.

A root-locus plot for the roll-rate loop showed that the dutch roll poles moved to the adjacent zeros, and the spiral pole was essentially unaffected by the roll-rate feedback. The roll-subsidence pole joined with the actuator pole to form a high-frequency complex pair, whose damping decreased as the feedback gain was increased. Roll-rate gains close to those used in Example 4.4-3 were tried, and the gain $k_p = -0.2$ produced a damping ratio of about 0.7 for the complex pair. A very fast but nonoscillatory roll response is desired (the control system investigated by NASA, for this model, was command-limited to about $300^\circ/\text{s}$), so this value was tentatively used for the controller.

The rudder-input-to-lateral-acceleration transfer function contains the NMP zero and has positive static loop sensitivity. The zero-angle root locus is required for satisfactory root loci behavior, so the a_y feedback gain must be negative. Figure 4.5-13 shows the root-locus plot for the lateral acceleration feedback when the inner-loop feedback gain is $k_R = -0.8$. Increasing the lateral acceleration feedback causes the dutch roll poles to circle around in the left-half s -plane, before terminating in the right-half plane on the NMP zero and at infinity. Increasing the inner-loop yaw-rate feedback causes the dutch roll poles to circle farther to the left in the s -plane and allows more lateral-acceleration feedback to be used. However, using large amounts of lateral-acceleration feedback creates a slow real pole by pulling the washout pole back to the right (it was moved left by the rate feedback) and makes the dutch roll pole positions quite sensitive to gain changes.

A conservative choice, $k_R = -0.8$, $k_A = -10$, was made for the rate and acceleration feedback gains, and these are shown in the controller subrou-

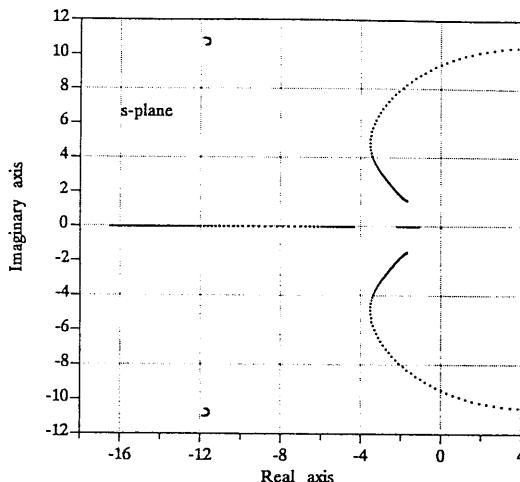


Fig. 4.5-13 Root-locus plot for lateral acceleration feedback.

tine. The relevant closed-loop transfer functions are

$$\frac{p}{p_c} = \frac{182.7(s + 13.10)(s + 2.428 \pm j2.243)(s + 1.538)(s - 0.002347)}{(s + 13.42)(s + 2.386 \pm j2.231)(s + 1.575)(s + 0.002116)(s + 11.78 \pm j10.96)} \quad (3)$$

$$\frac{a_y}{r_c} = \frac{0.2539(s - 4.157)(s + 4.00)(s + 11.92 \pm j10.58)(s + 1.00)(s - 0.0001965)}{(s + 13.42)(s + 2.386 \pm j2.231)(s + 1.575)(s + 0.002116)(s + 11.78 \pm j10.96)} \quad (4)$$

A number of poles and zeros can be canceled out of the transfer functions, and there is good decoupling between the two channels. The static loop sensitivity of the first transfer function has changed because the feedback gain k_p has been moved into the forward path, as shown in Fig. 4.5-11. Note that the dutch roll mode is satisfactory, and the spiral mode is stable but with an increased time constant.

A nonlinear simulation was used to evaluate this control system. The trajectory must be chosen to exercise the ARI, through high-alpha and fast roll rates, yet be easily preprogrammed for non-real-time simulation. A suitable trajectory is a pull-up into a vertical loop, with a 180° roll at the top of the loop, and continuing into a second vertical loop. The calling program and the preprogrammed commands are shown in Fig. 4.5-14. The alpha-filter and actuator states were initialized exactly with the correct initial conditions, but the other controller states were left uninitialized, so the first 5 s of the flight are used to let any transients die away. Full throttle and a 15°/s pitch rate command are applied at $t = 5$ s, and then a roll-rate command pulse is applied between 15 and 17 s. The desired roll rate is therefore 90°/s, but

```

PROGRAM FLY
EXTERNAL FC,D
COMMON/PARAM/XCG
XCG = .35
CALL TRESP(FC,D) ! generic time response pgm.
END

C
SUBROUTINE D(TIME,TS,X,XD,DFLAG)
DIMENSION X(*),XD(*)
COMMON/CONTROLS/THTL,EL,AIL,RDR,PCOM,QCOM,RCOM
LOGICAL DFLAG
IF (TIME .LT. 5.0) THEN
  QCOM= 0.0
  PCOM= 0.0
  RCOM= 0.0
ELSE IF (TIME .LT. 15.0) THEN
  QCOM= 15.0
  THTL= 1.0
ELSE IF (TIME .LT. 17.0) THEN
  PCOM= 150.0
ELSE
  PCOM = 0.0
END IF
RETURN
END

```

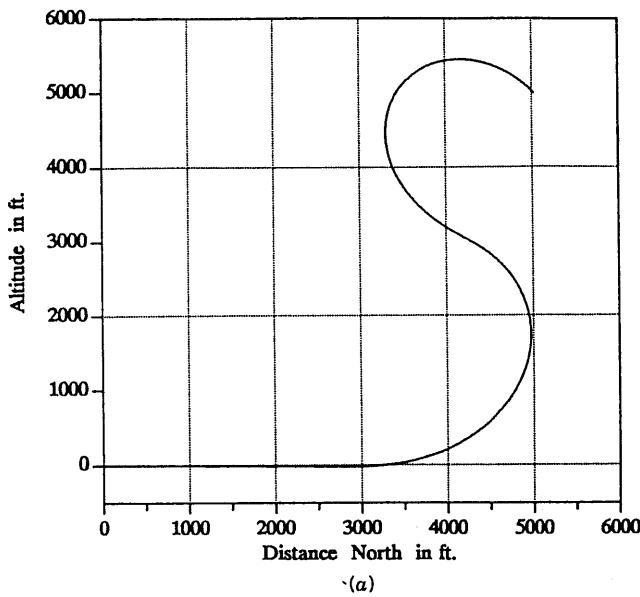
Fig. 4.5-14 Commands for the simulated S-trajectory.

because of the finite error and finite response time of the roll-rate loop, the rate command had to be adjusted by trial and error to achieve the 180° roll.

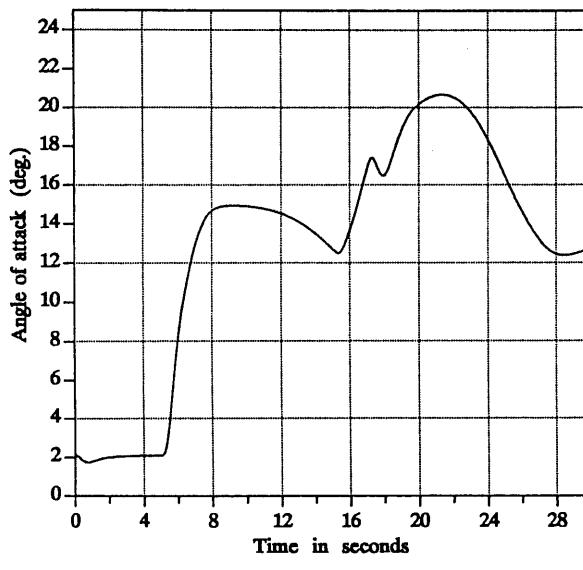
An elevation view of the trajectory is shown in Fig. 4.5-15a. The first loop corresponds to a normal acceleration of about $4g$, and the aircraft speed decreases roughly linearly from 500 ft/s at 5.0 s to 270 ft/s at 24 s (near the top of the second loop). Figure 4.5-15b and c show angle of attack, bank attitude, and pitch attitude. Alpha increases rapidly as the loop is started, remains roughly constant to provide the centripetal acceleration while the pitch attitude is between 45° and 90° , and then starts to fall off as gravity helps to provide the centripetal acceleration. During the second loop alpha rises to a larger peak, because the airspeed has dropped considerably by then.

The bank angle of 0° suddenly becomes a bank angle of 180° as the aircraft passes through the vertical attitude condition, and this wings-level attitude is held until the roll is started at 15 s. The attitude angles are computed by integrating the angular rates (state derivatives), not from trigonometric functions, so the bank angle may contain multiple 360° ambiguities, depending on how the angular rates behave.

Figure 4.5-15d shows the fast roll-rate response and the corresponding yaw rate that is generated by the ARI. The pitch rate undergoes a positive perturbation during the roll, and this is due to the nose-up pitching moment generated by inertia coupling. Figure 4.5-15e shows the positive increment in elevator deflection that is generated by the longitudinal control system to counteract the inertia-coupling moment. As stated earlier, a major problem

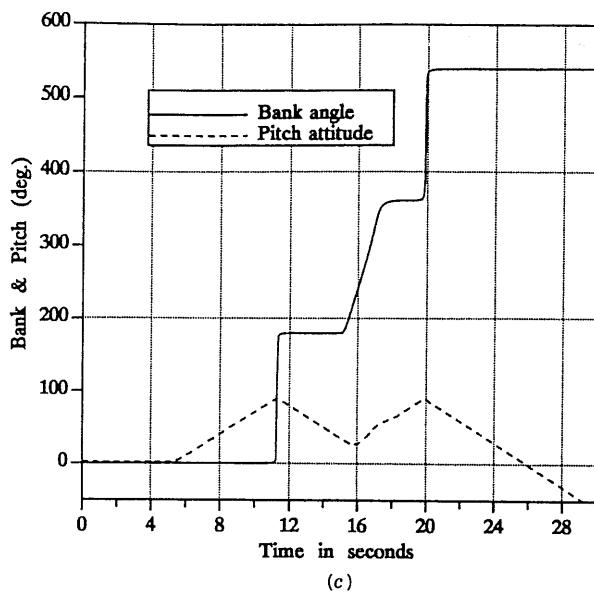


(a)

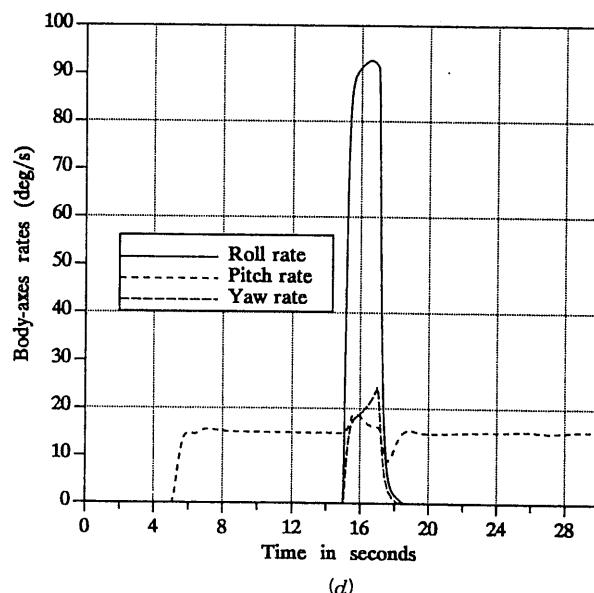


(b)

Fig. 4.5-15 (a) Elevation view of the simulated S-trajectory. (b) Angle of attack on the S-trajectory.

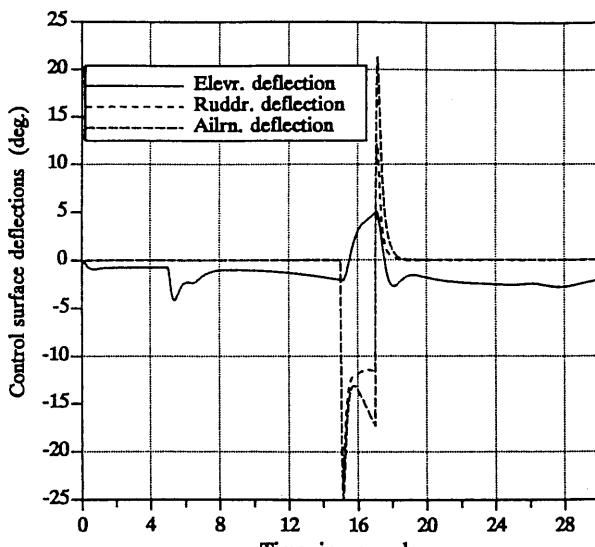


(c)

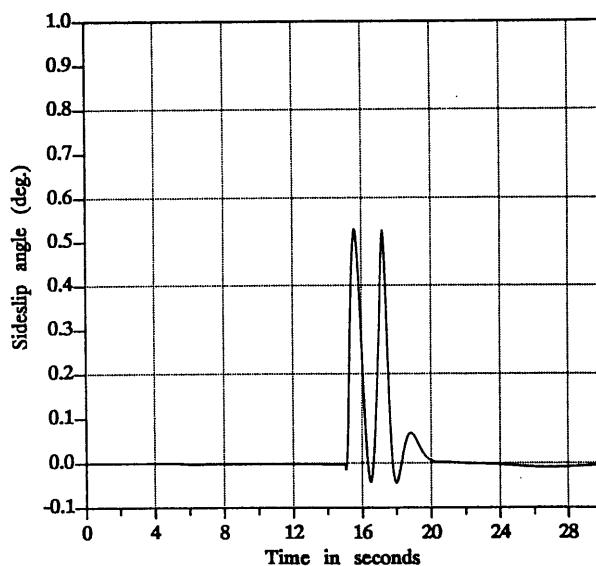


(d)

Fig. 4.5-15 (continued) (c) Bank and pitch angles on the S-trajectory. (d) Body-axes rates on the S-trajectory.



(e)



(f)

Fig. 4.5-15 (continued) (e) Control surface deflections on the S-trajectory. (f) Sideslip variation on the S-trajectory.

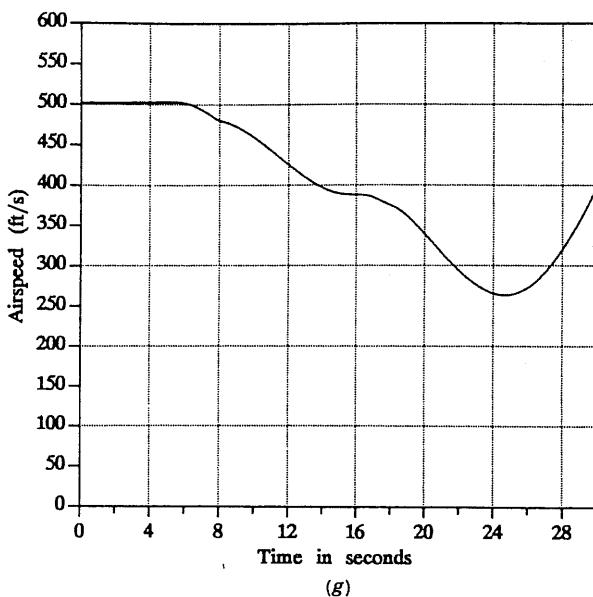


Fig. 4.5-15 (continued) (g) Speed variation on the S-trajectory.

when rolling rapidly at still higher angles of attack is that the longitudinal control surfaces may be unable to generate a large enough nose-down moment.

In this example the elevator deflections are quite small, but the aileron and rudder deflections are large. This is due to the combination of high demanded roll-rate and low aileron effectiveness (because of the high alpha and relatively low dynamic pressure). It is also partly due to the fact that while the rudder is generating the required yaw rate, it is also generating a rolling moment that opposes the aileron rolling moment. The large peak deflections are due to the instantaneous demand for the high roll rate. Note that the aileron and rudder deflection rates may have reached or exceeded the capabilities of their actuators; this concern is addressed in Section 4.7.

Figure 4.5-15f shows that the control system has done an excellent job of keeping the sideslip angle small during this demanding maneuver. The sideslip excursions are biased positively, that is, toward adverse beta. This is desirable in general; a combination of adverse and proverse beta tends to excite the dutch roll mode, and is avoided. The ARI gain is quite critical, and the values used are close to optimal. Larger values will produce a single negative and single positive beta excursion, that are more nearly symmetrical about zero but considerably larger in magnitude. Figure 4.5-15g shows the variation of airspeed with time. The maneuvers have caused the speed to fall continuously until the top of the second loop is passed, despite full throttle

being used throughout. The reduction in speed, by a factor of 2, will cause a fourfold reduction in dynamic pressure.

Finally, note that the performance is satisfactory for these flight conditions, but the design must be evaluated at other altitude/speed combinations. Gain scheduling with Mach number will probably be required, and a much more comprehensive simulation is necessary before the design can be considered practical. ■

4.6 AUTOPILOTS

Most of the flying-qualities specifications do not apply directly to autopilot design. In the case of pilot-relief autopilot modes, the autopilot must be designed to meet specifications on steady-state error and disturbance rejection, with less emphasis on dynamic response. In addition, special consideration must be given to the way in which the autopilot is engaged and disengaged, so that uncomfortable or dangerous transient motions are not produced. For example, the altitude-hold autopilot that we shall design could not be engaged directly at a few hundred feet below the commanded altitude. Otherwise, the result would be a very steep climb, leading to a stall if the engine thrust was not increased.

On the other hand, navigation-coupled autopilot modes must be designed to have a dynamic response that is appropriate to their function. For example, in an automatic terrain-following mode an autopilot must track a randomly changing input, of quite wide bandwidth, without significant overshoots in its response. A number of autopilot designs will now be illustrated using the transport-aircraft and F-16 dynamic models.

Pitch-Attitude Hold

This autopilot is normally used only when the aircraft is in wings-level flight. The controlled variable is θ ($\theta = \gamma + \alpha$), and the sensor is an attitude reference gyro (which provides an error signal proportional to the deviation from a preset orientation in inertial space). The controller does not hold the flight-path angle, γ , constant because the angle of attack changes with flight conditions. Thus, if thrust is increased, alpha will tend to decrease and the aircraft will climb; and as aircraft weight decreases (as fuel is burned), alpha will decrease, also causing a gradual climb. Similarly, a preset climb will gradually level out as decreasing air density causes alpha to increase. Because of these characteristics the pitch-attitude-hold autopilot is not very important in its own right. However, the same feedback configuration is used in the inner loops of other autopilots, such as altitude hold and automatic landing.

The block diagram of an attitude-hold autopilot is shown in Fig. 4.6-1. Dynamic compensation, $G_c(s)$, is necessary if a small steady-state error and

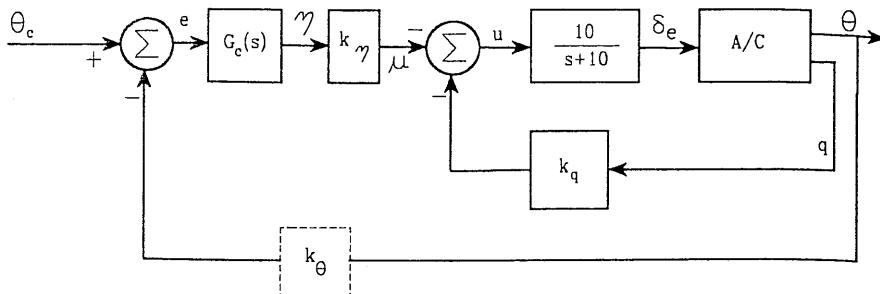


Fig. 4.6-1 Attitude-hold autopilot.

good transient response are required. Inner-loop rate feedback is used to provide additional design freedom and to promote good short-period damping. The design can be performed using the short-period approximation for the aircraft dynamics, and adding an integrator to obtain pitch from pitch rate. This approach is not very satisfactory because pitch attitude is one of the variables involved in the phugoid mode, and feedback of pitch attitude to the elevator will change both the short-period and phugoid modes. It will be seen in the following design examples that both modes are involved in the time response, and both modes influence the compensator design.

Altitude is also involved in the phugoid mode (provided that an atmosphere model is included in the dynamics), so the altitude state should be included in the dynamics if the effect of pitch-attitude feedback on the phugoid is to be determined accurately. Two design examples will be given, with and without a dynamic compensator, and these designs will be used later as parts of more complex autopilots. The first example will be for a high-altitude cruise condition, and the second for a landing condition.

Example 4.6-1: A Simple Pitch-Attitude-Hold Autopilot. This example will demonstrate the performance that can be achieved by adjusting the gains k_q and k_γ , in Fig. 4.6-1, with no dynamic compensation. The dynamics of the transport-aircraft model (Section 3.3) in a level-flight cruise condition at 25,000 ft, 500 ft/s true airspeed, and $x_{cg} = 0.25\bar{c}$ are given by

$$A = \begin{bmatrix} v_T & \alpha & \theta & q & h \\ -0.0082354 & 18.938 & -32.170 & 0.0 & 5.9022E-05 \\ -0.00025617 & -0.56761 & 0.0 & 1.0 & 2.2633E-06 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 1.3114E-05 & -1.4847 & 0.0 & -0.47599 & -1.4947E-07 \\ 0.0 & -500.00 & 500.00 & 0.0 & 0.0 \end{bmatrix} \quad (1)$$

$$B^T = [0 \ 0 \ 0 \ -0.019781 \ 0] \quad (\text{single input } \delta_e).$$

If the dynamics above are augmented with the 0.1-s simple-lag actuator model shown in the figure, and put into the state-space form needed to

determine the feedback gains, the result is

$$\begin{aligned}\dot{x} &= \begin{bmatrix} A & B \\ 0 & -10 \end{bmatrix}x + \begin{bmatrix} 0 \\ 10 \end{bmatrix}u \\ y &= \begin{bmatrix} \theta \\ q \end{bmatrix} = \begin{bmatrix} 0 & 0 & 57.29578 & 0 & 0 \\ 0 & 0 & 0 & -57.29578 & 0 \end{bmatrix}x \\ x^T &= [v_T \quad \alpha \quad \theta \quad q \quad h \quad \delta_e], \quad u = -Ky = -[k_\theta \quad k_q]y.\end{aligned}\quad (2)$$

The desired forward-path gain, k_η , is found from $k_\eta = -k_\theta$ (the closed-loop static loop sensitivity will then be different; see below).

The transfer function from δ_e to θ is found to be

$$\frac{\theta}{\delta_e} = \frac{-1.133(s + 0.5567)(s + 0.01897)(s + 1.666E-4)}{(s + 0.5234 \pm j1.217)(s + 0.002471 \pm j0.08988)(s + 1.892E-4)}. \quad (3)$$

All of the modes are stable, but the complex modes are quite lightly damped ($\zeta_{sp} = 0.395$, $\zeta_p = 0.027$) in this flight condition. The altitude pole is almost canceled by a zero, but omitting the altitude state will cause a noticeable error in the phugoid parameters. This transfer function shows that the configuration in Figure 4.6-1 will be a Type-0 control system (with no dynamic compensator added).

The effect of pitch-attitude feedback on transfer function (3) can be deduced from the root-locus rules. The altitude pole will move to the nearby zero, and the phugoid poles will move to the real axis and eventually terminate on the two remaining zeros. When the effect of the actuator pole is accounted for, the short-period poles must move toward the right-half plane (approaching 60° asymptotes). Thus the short-period mode becomes less well damped as the phugoid damping increases. Because the effect of the pitch attitude feedback is self-evident, it is convenient to fix k_θ and use a root-locus plot to investigate the effect of varying the pitch-rate feedback.

Figure 4.6-2 shows the root-locus plot when $k_\theta = -4.0$ (elevator degrees per degree of pitch). All of the poles except the short-period poles are on the real axis, and the damping of the short-period poles passes through a maximum as k_q is varied. The upper branch of the loci will move upward and to the right as k_θ is increased, thus reducing the maximum damping that can be attained.

The maximum short-period damping in Fig. 4.6-2 is more than adequate, and a gain ($k_q = -2.5$) corresponding to lower damping and reduced natural frequency ($\zeta = 0.64$, $\omega_n = 3.12$) was selected. Thus the gains $k_q = -2.5$ and $k_\eta = -k_\theta = 4$ give the closed-loop transfer function

$$\frac{\theta}{\delta_e} = \frac{45.33(s + 0.5567)(s + 0.01897)(s + 1.666E-4)}{(s + 1.999 \pm j2.389)(s + 6.646)(s + 0.3815)(s + 0.02522)(s + 1.718E-4)}. \quad (4)$$

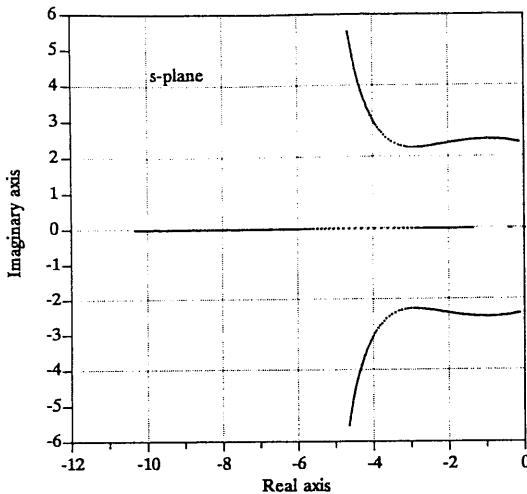


Fig. 4.6-2 Root-locus plot for the attitude-hold autopilot.

The static loop sensitivity of (4) follows from that of (3) when the actuator and k_n are accounted for. The altitude pole is almost canceled by a zero, but the cancellation of the degenerate phugoid poles is less exact and they are readily apparent in the step response. The step response is shown in Fig. 4.6-3; it eventually settles at about 0.77, thus exhibiting a significant steady-state error.

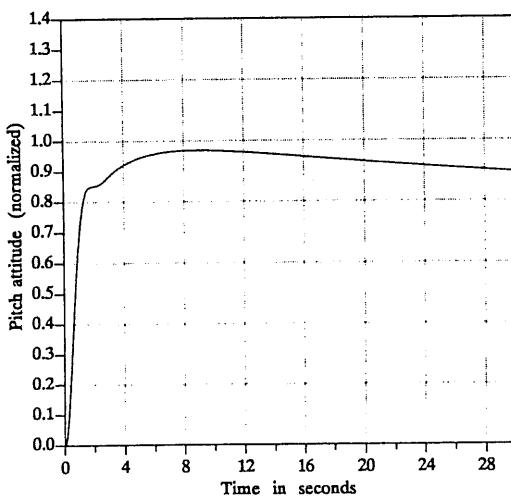


Fig. 4.6-3 Pitch-attitude autopilot step response.

This design has the disadvantage that as k_n is increased to reduce the steady-state error, large values of k_q (e.g., $k_q = 3^\circ$ of elevator per degree per second of pitch rate) must be used to obtain adequate damping of the short-period poles. This is likely to cause problems with rate-sensor noise or structural mode feedback. In the next example a PI compensator will be used to remove the steady-state error, and a phase-lead compensator will be used to improve the dynamic response. ■

Example 4.6-2: A Pitch-Attitude Hold with Dynamic Compensation.

When the transport aircraft model is trimmed with landing gear and flaps deployed, at $V_T = 250$ ft/s, $h = 50$ ft, $\gamma = -2.5^\circ$, and $x_{cg} = 0.25\bar{c}$, the dynamics are described by

$$A = \begin{bmatrix} v_T & \alpha & \theta & q \\ -3.8916E-02 & 1.8992E+01 & -3.2139E+01 & 0.0000E+00 \\ -1.0285E-03 & -6.4537E-01 & 5.6129E-03 & 1.0000E+00 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ 8.0847E-05 & -7.7287E-01 & -8.0979E-04 & -5.2900E-01 \end{bmatrix} \quad (1)$$

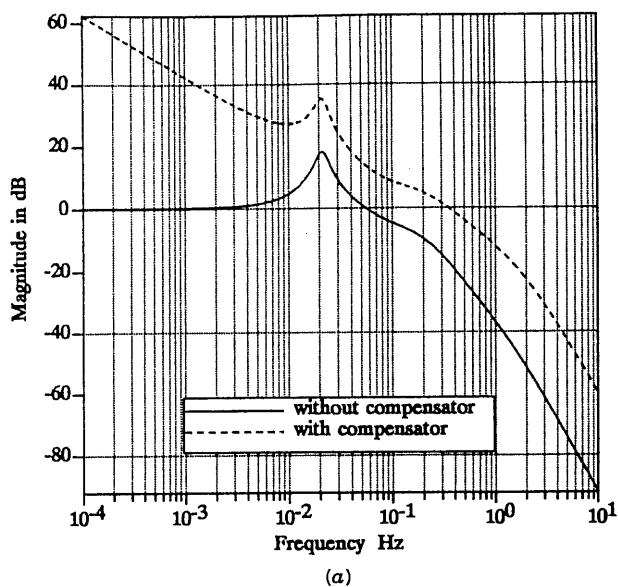
$$B^T = [0 \quad 0 \quad 0 \quad -0.010992] \quad (\delta_e).$$

For simplicity the altitude state has been omitted, since its effect on the design is negligible. The elevator actuator model from Example 4.6-1 was also incorporated into these dynamics.

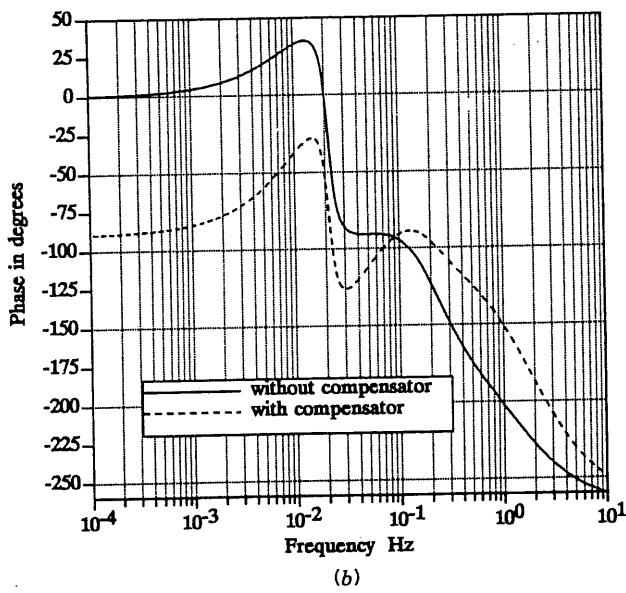
A pitch-rate feedback gain $k_q = -1$ results in a short-period damping ratio of $\zeta_{sp} = 0.74$; this is a more practical value of k_q than in Example 4.6-1 and corresponds to good short-period damping. The elevator-input-to-pitch-attitude transfer function is then

$$\frac{\theta}{\mu} = \frac{-6.298(s + 0.6112)(s + 0.07305)}{(s + 0.9442 \pm j0.8674)(s + 0.01836 \pm j0.1328)(s + 9.288)}. \quad (2)$$

These poles and zeros were placed in a data file, so that the frequency-response program could be used to design a phase-lead and proportional-plus-integral compensator for the pitch-attitude loop. Figure 4.6-4a and b show the frequency response of transfer function (2) alone, with the lightly damped phugoid mode ($\zeta = 0.14$, $\omega_n = 0.13$) very much in evidence. The PI compensator can be chosen to modify the frequency response in the range up



(a)



(b)

Fig. 4.6-4 Bode plots for the attitude-hold autopilot.

to the phugoid frequency, and to add very little lag at higher frequencies where the lead compensator is to be placed. Therefore, the PI zero should be placed near $s = -0.1$ ($\omega_c = 0.1$ rad/s). If the zero is moved right, the response will tend to be slowed down by poles trapped near the origin, and if moved left, the phugoid mode will become less damped because of the additional phase lag.

A lead compensator pole/zero ratio of 10 was chosen to obtain tight control of the pitch attitude, with later application to an automatic landing system in mind. The loop gain was raised progressively as the compensator was adjusted to optimize the gain and phase margins. The compensator

$$G_c = 40 \left(\frac{s + 0.2}{s} \right) \left(\frac{s + 1.4}{s + 14} \right) \quad (3)$$

gives a phase margin of 66.8° at 0.34 Hz and a gain margin of 21 dB at 1.76 Hz. The outer-loop gain and phase plots with this compensator are also shown in Fig. 4.6-4a and b.

The short-period and phugoid modes interact strongly in this autopilot, and frequency-domain design alone is inadequate to secure a good time response. A time response without large overshoots requires that the closed-loop poles be quite well damped ($\zeta \approx 0.8$). In addition, small residues in the slow poles require that the compensator gain be quite large, and the PI zero be chosen so that the closed-loop poles closely approach the zeros. The fine adjustment of the compensator above was arrived at by checking the closed-loop transfer function and the closed-loop step response, using the following augmented state equations.

If the A matrix in (1) is augmented with the elevator actuator and the compensator (3), then, referring to Fig. 4.6-1 and Table 3.2-1, the complete dynamics are given by

$$\dot{x} = \begin{bmatrix} & & & & 0 & 0 \\ & A & & B & 0 & 0 \\ & & & & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & -57 & 0 & 0 & 0 \\ 0 & 0 & -57 & 0 & 0 & 0.2 \\ \hline & & & & 0 & -14 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 10 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} r$$

$$y = \begin{bmatrix} 0 & 0 & -57 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & -12.6 \end{bmatrix}x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}r \quad (4)$$

$$u = - [k_q \quad k_\eta] y.$$

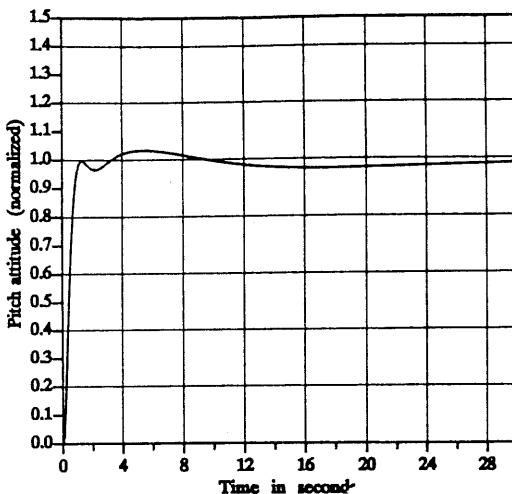


Fig. 4.6-5 Pitch-attitude autopilot step response.

These equations can now be used to close the loops and check the corresponding time response. When the gain is

$$K = [-1 \quad 40], \quad (5)$$

the closed-loop pitch-attitude transfer function is given by

$$\frac{\theta}{\theta_c} = \frac{251.9(s + 0.6112)(s + 0.07305)(s + 1.40)(s + 0.20)}{(s + 2.121 \pm j1.762)(s + 0.2717 \pm j0.1516)(s + 0.06335)(s + 4.170)(s + 16.19)}. \quad (6)$$

In the transfer function (6) the short-period mode has increased in frequency [compared to (2)], and the phugoid mode has increased in frequency and become more damped. Figure 4.6-5 shows the step response; the slow poles are readily apparent, but the response is within 4% of its final value in less than 2 s, and remains there. The steady-state error will eventually be removed by the error integrator. Unlike the previous design, this design requires only a moderate amount of pitch-rate feedback. ■

Altitude Hold / Mach Hold

Altitude hold is an important pilot-relief mode; it allows the aircraft to be held at a fixed altitude in an air-route corridor, to meet air-traffic control requirements. The sensed altitude is normally the *pressure altitude*, that is, altitude computed in the air-data computer from external pressure measure-

ments. In a modern passenger aircraft the altitude hold will typically hold the aircraft within ± 200 ft and provide a warning signal if the deviation exceeds ± 100 ft. The system will have *limited authority* over the horizontal control surfaces and will again warn the pilot if the control limits have been reached. These situations will often occur, for example, in rapidly rising air currents deflected upward by mountain ranges ("mountain waves"). A modern system may also have an "easy-on" or "fly-up, fly-down" feature that allows the autopilot to take the aircraft to an assigned altitude without exceeding certain rate-of-climb and pitch-attitude limits (e.g., 2000 to 3000 ft/min, 20° pitch attitude).

Mach-hold (at high speed) is similar to altitude hold in that it is used for cruise conditions, involves elevator control, and the same inner-loop feedback signals and mixture of fast and slow poles. When the Mach number is held constant the aircraft will climb slowly as fuel is burned. This promotes efficient cruising operation because the optimum altitude for fuel efficiency increases as aircraft weight decreases. A Mach-hold system could be designed in a very similar way to the following altitude-hold design example.

Example 4.6-3: An Altitude-Hold Autopilot Design. The altitude-hold configuration is shown in Fig. 4.6-6, where G_c is a compensator and G_F is the effective lag of the pressure-altitude measurement. In the interest of simplicity the altitude-sensor lag will be omitted from this example; the design procedure is unchanged by this omission. Again for simplicity, the basic pitch-attitude autopilot from Example 4.6-1 will be used to provide the inner loops of the design, and the compensator G_c will still allow good altitude control to be achieved. The first design goal will be to achieve a high loop gain, for good rejection of low-frequency (lf) altitude disturbances and small altitude error. Second, an altitude response that is deadbeat and relatively slow will be required for energy efficiency and passenger comfort.

Altitude is one of the state variables, and by adding an appropriate row to the C matrix in Example 4.6-1, the transfer function from the pitch-attitude command to altitude can be determined. The altitude feedback has a strong effect on the phugoid poles and a relatively weak effect on the short-period

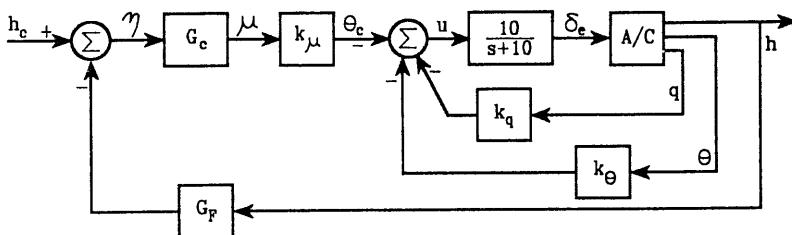


Fig. 4.6-6 Altitude-hold autopilot.

poles. Therefore, the damping of the short-period mode will initially be set close to the desired final value. Thus, based on the experience of Example 4.6-1, the pitch-rate and pitch-attitude feedback loops will be closed by gains $k_q = -2.5$ and $k_\theta = -3.0$. The altitude transfer function is then

$$\frac{h}{\theta_c} = \frac{-56.14(s + 0.002264)}{(s + 2.261 \pm j1.936)(s + 6.170)(s + 0.3333)(s + 0.02750)(s + 1.731E - 4)}, \quad (1)$$

with short-period damping ratio of 0.76.

Simple phase-lead compensation can be used to close the altitude loop. For example, the lead compensator

$$G_c = 1.0 \frac{s + 0.3}{s + 2.4} \quad (2)$$

gives gain and phase margins of 13 dB and 70° , respectively. Unfortunately, the lead compensator reduces the If loop gain. The transfer function (1) has an If gain of about 63.3 dB (or 1460) and the compensator (2) reduces this by 18 dB (i.e., $\frac{1}{8}$). The final loop gain of 45.3 dB (or 183) would allow a steady-state altitude error of 1 ft per 184 ft, a rather poor performance. The performance can be improved by adding a lag compensator that boosts the If gain, while adding negligible phase lag in the frequency range of the lead compensator. The same effect can be achieved by using a PI compensator to make the altitude control loop Type 1, and placing the PI zero close to the origin. A simple lag compensator has the advantage that it can be implemented with passive components (see Table 3.2-1), provided that the time constant is not too large. Modern electronics has diminished this advantage, but a lag compensator is still simpler and more reliable than a PI compensator, and its use will be illustrated here.

Practical considerations limit the maximum time constant of the passive lag compensator to about 100 s (pole at $s = -0.01$). If the compensator zero is chosen to give a large If gain increase, then it will be found that in the closed-loop transfer function, the slow poles (from $s = -0.0275$ and $s = -0.01$) have relatively large residues (i.e., do not cancel with zeros). If the lag compensator zero is placed near $s = -0.05$ (i.e., an If gain increase of 5), these slow poles will have a relatively small effect on the closed-loop time response. There will also be less phase lag in the frequency range where the lead compensator is to be added. Therefore, the lag-compensator zero was placed at $s = -0.05$. [The reader should sketch a root locus for (1), with the lag and lead compensators, and also repeat this design with the lag compensator zero at $s = -0.1$.]

The complete compensator will now be of the form

$$G_c = \frac{s + z_1}{s + p_1} \times \frac{s + z_2}{s + p_2} \times k \quad (z_1 = 0.05, p_1 = 0.01, p_2 > z_2), \quad (3)$$

where the gain factor k will become the feedback gain k_μ in Figure 4.6-6. A lead compensator pole/zero ratio of 8 was chosen, since this provides a large amount of lead (maximum lead 51°) without greatly accentuating any sensor noise. As in Example 4.6-2, the compensator was moved along the frequency axis until the best compromise between gain and phase margins was achieved, and if these were found to be excessive, the gain, k , was increased. A dead-beat step response was found to require a large phase margin, and the values

$$z_2 = 0.3, \quad p_2 = 2.4, \quad k = 1.0,$$

give gain and phase margins of 13.1 dB and 65.7°, respectively. This compensator will reduce the lf loop gain by a factor of $\frac{5}{8}$ (to 913), but it is still adequate. Figure 4.6-7a and b show the compensated and uncompensated Bode plots. The gain plots show that the compensator has smoothed out the gain variations, and the phase plots show that phase lead has been added only in the region where it is required.

The closed-loop time response of the compensated system must be checked, and this can be done as follows. Using two additional state variables, x_7 and x_8 , the method of Section 3.2 gives the following state and output equations for the compensator:

$$\begin{aligned} \dot{x}_7 &= x_8 \\ \dot{x}_8 &= \eta - (p_1 p_2)x_7 - (p_1 + p_2)x_8 \\ \mu &= u + (z_1 z_2 - p_1 p_2)x_7 + (z_1 + z_2 - p_1 - p_2)x_8. \end{aligned} \quad (4)$$

The respective input and output variables are η and μ , as used in Fig. 4.6-6. When these equations are added to the coefficient matrices used in Example 4.6-1, we obtain

$$\begin{aligned} \dot{x} &= \begin{bmatrix} A & B \\ \hline 0 & 0 & 0 & 0 & 0 & -10 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 10 \\ 0 \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} h_c \\ y &= \begin{bmatrix} \theta \\ q \\ \mu \end{bmatrix} = \begin{bmatrix} 0 & 0 & 57.29578 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 57.29578 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & \nu \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} h_c \end{aligned} \quad (5)$$

where

$$\nu = (z_1 z_2 - p_1 p_2), \quad \xi = (z_1 + z_2 - p_1 - p_2),$$

$$x^T = [v_T \quad \alpha \quad \theta \quad q \quad h \quad \delta_e \quad x_7 \quad x_8],$$

$$u = -Ky = -[k_\theta \quad k_q \quad k_\mu] y.$$

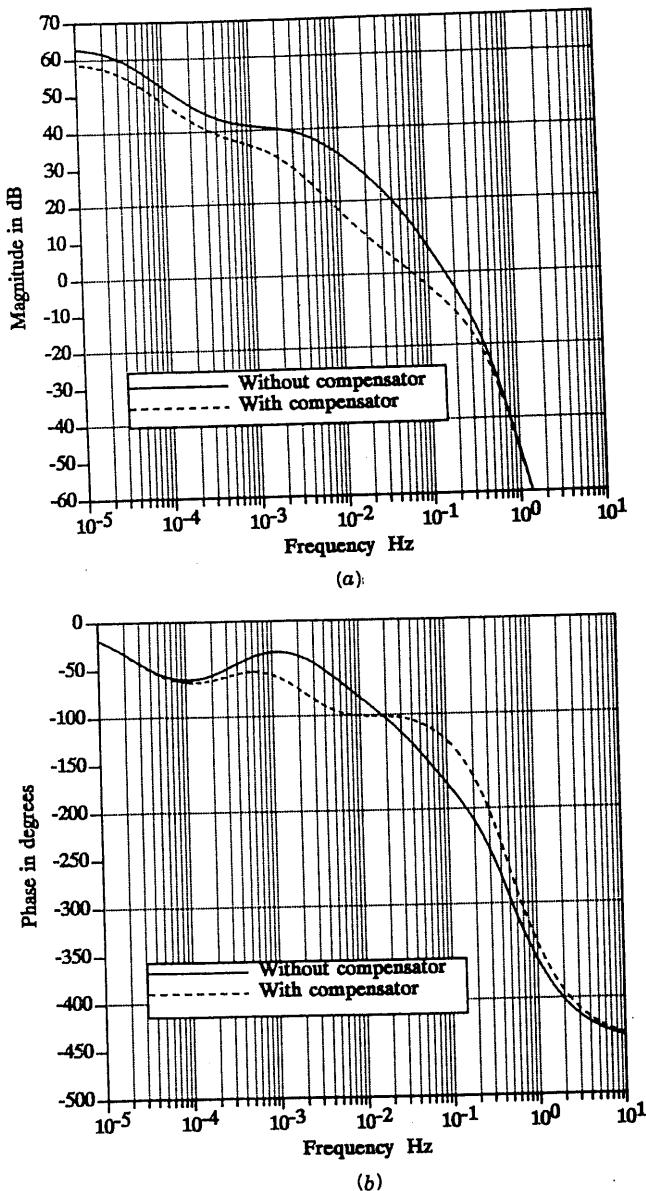


Fig. 4.6-7 Bode plots for the altitude-hold autopilot.

When the loops are closed with the gain

$$K = [-3 \quad -2.5 \quad 1] \quad (6)$$

the closed-loop coefficient matrices can be used with the time-response program to compute a step response, and with the pole-zero program to obtain a (factored) transfer function. Thus the closed-loop altitude transfer function is

$$\frac{h}{h_c} =$$

$$\frac{56.14(s + 0.30)(s + 0.050)(s + 0.002264)}{(s + 6.29)(s + 2.75 \pm j2.03)(s + 0.673 \pm j0.604)(s + 0.267)(s + 0.053)(s + 0.00224)} \quad (7)$$

A comparison with transfer function (1) shows that the fast poles have not moved significantly, the three slowest poles essentially cancel out of the transfer function, and a new complex pair has been created.

The step response, obtained by numerical integration from the closed-loop coefficient matrices, is shown in Fig. 4.6-8. The effect of the slow poles is visible as a small, slowly decaying, displacement from the final value. The

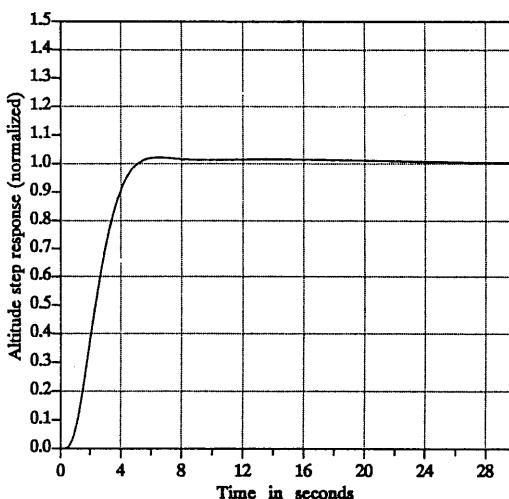


Fig. 4.6-8 Step response of the altitude-hold autopilot.

steady-state error will be negligible because of the high value of the If loop gain. The response is essentially deadbeat and is considerably slower than a pitch-axis response. It could be slowed down further by reduction of the loop gain or by using additional lag compensation, but is considered to be satisfactory. As pointed out earlier, it is obvious that this autopilot would not normally be directly engaged with a large altitude error.

The reader may wish to consider repeating this design for cruising conditions at, say, 35,000 ft, to determine the need for scheduling of the controller gains. ■

Automatic Landing Systems

In Section 4.1 we referred to the need for automatic control in situations where controlling the trajectory of an air vehicle was too difficult a task for a human pilot. A particular case of this is the landing phase in conditions such as bad weather or limited visibility. Landing in limited visibility may be achieved by providing the pilot with instruments to determine the aircraft position relative to a reference trajectory, but a landing in more difficult conditions requires full automatic control with the pilot playing only a supervisory role.

Automatic control of the trajectory requires simultaneous control of engine thrust and pitch attitude because, for example, using only the elevator to attempt to gain altitude will result in a loss of speed and an eventual stall. If the landing speed is such that the aircraft is on the "back side" of the power curve (see Section 3.4), the throttles control altitude and the elevator controls airspeed (increased power causes a gain in altitude, *down elevator* causes a gain in speed).

An aircraft is normally *reconfigured* for landing and takeoff by deploying flaps and wing leading-edge devices (slats) so that the wing effectively has more camber. The increased camber provides more lift at low speed, and increased drag; the wing is therefore optimized for a low-speed landing. The reconfiguration has the effect of moving the minimum of the power curve to lower speed. Thus most aircraft do not operate on the back side of the power curve, although naval aviators are routinely taught to fly in this regime for aircraft-carrier operations. The reconfigured wing and extended landing gear produce a strong nose-down moment, which in turn leads to a trim with a large amount of "up" elevator. We shall see this effect in our transport-aircraft model in the following example, and these conditions play a role in determining the elevator size and deflection limit during the aircraft design.

A typical automatic landing system uses a radio beam directed upward from the ground at 2.5° to 3.5° , with equipment onboard the aircraft to measure the angular deviation from the beam and compute the perpendicular displacement of the aircraft from the *glide path*. Additional equipment is used to provide azimuth information, so that the aircraft can be lined up with the runway. The glide path will usually be intercepted at about 1200 to 1500

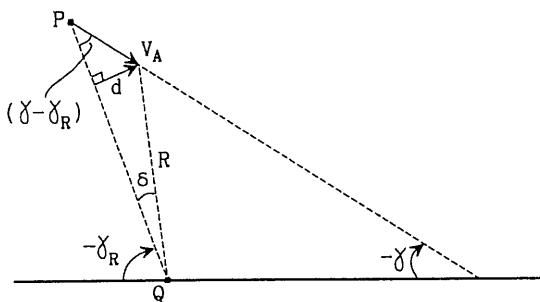


Fig. 4.6-9 Geometry of the glide-path control system.

ft altitude, with an airspeed of 145 to 150 knots (245 to 253 ft/s), and then the automatic control system will be engaged. The aircraft will descend with a constant flight-path angle (γ), a constant airspeed, and a pitch attitude of between about -5° and 5° (possibly higher for noncommercial aircraft).

Figure 4.6-9 shows an elevation view of a descending trajectory with velocity V_A (in the NED frame) and flight-path angle γ . The reference trajectory has an angle γ_R , and the radio beam equipment is at the position Q . Assume that the aircraft passes through the radio beam at point P and time t_1 , and that the descent is too gradual (as shown). The resulting positive glide path deviation that builds up is denoted by d . The automatic landing equipment measures the angular deviation δ and the range R and calculates d from

$$d = R \sin \delta. \quad (4.6-1)$$

An onboard automatic control system is used to maneuver the aircraft so that d is driven back to zero.

To design a control system we must relate d to the aircraft trajectory. The geometry of the figure shows that the derivative of d is given by

$$\dot{d} = V_A \sin(\gamma - \gamma_R). \quad (4.6-2)$$

Therefore, $d(t)$ can be derived by integrating this equation with the aircraft state equations, with the initial condition $d(t_1) = 0$ applied at the time t_1 at which the aircraft intersects the glide path. Note that when d is computed from (4.6-1), the sensitivity of d to flight-path changes will depend on the range R . This effect will be assumed to be compensated for in the onboard computer, so that an automatic control system can be designed for some nominal value of the range. The design of the longitudinal control system for automatic landing will now be presented as an example.

Example 4.6-4: Longitudinal Control for Automatic Landing. Figure 4.6-10 is a block diagram of the auto-land control system. The transport aircraft model in the landing configuration will be used. The throttle servo and engine response will be modeled by a single 5-s lag, and the elevator servo by a 0.1-s lag, as shown; sensor lags have been neglected. The compensators that must be designed are G_1 and G_2 , and the pitch-attitude autopilot (including compensator G_3) will be taken from Example 4.6-2.

Equation (4.6-2) was added to the transport aircraft model, with γ_R as a model input and d as an additional state. The model was trimmed with gear down and landing flap settings for the conditions $V_T = 250 \text{ ft/s}$, $\gamma = -2.5^\circ$, $x_{cg} = 0.25\bar{c}$, and $h = 750 \text{ ft}$ (halfway down the glide slope). The A and B Jacobian matrices for this flight condition are

$$A = \begin{bmatrix} v_T & \alpha & \theta & q & h & d \\ -0.038580 & 18.984 & -32.139 & 0 & 1.3233E-4 & 0 \\ -0.0010280 & -0.63253 & 0.0056129 & 1.0 & 3.7553E-6 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 7.8601E-5 & -0.75905 & -0.00079341 & -0.51830 & -3.0808E-7 & 0 \\ -0.043620 & -249.76 & 249.76 & 0 & 0 & 0 \\ 0 & -250.00 & 250.00 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

$$B = \begin{bmatrix} \delta_{th} & \delta_e \\ 10.100 & 0 \\ -1.5446E-4 & 0 \\ 0 & 0 \\ 0.024656 & -0.010770 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \quad (2)$$

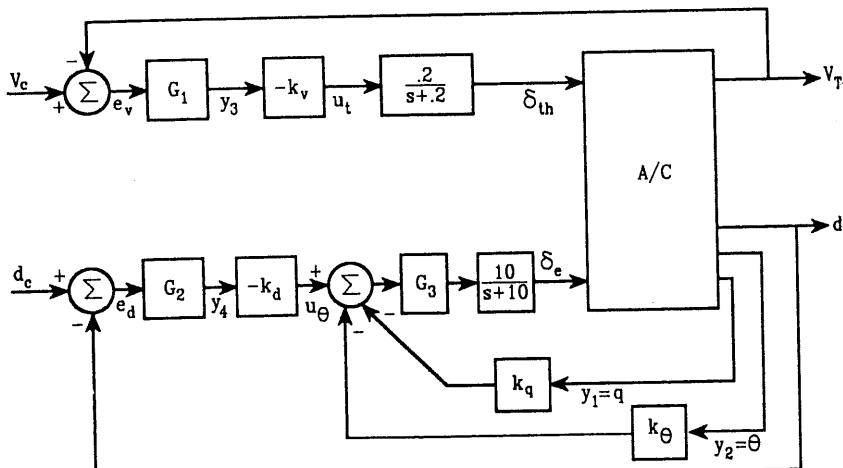


Fig. 4.6-10 Longitudinal control system for automatic landing.

```

SUBROUTINE FC(TIME,X,XD)
DIMENSION X(*), XD(*)
COMMON/CONTROLS/THTL,EL,GAMREF,VCOM,THCOM,DCOM
COMMON/OUTPUT/AN,D2,D3,QBAR,AMACH,VT,ALPHA,THETAD, QD,H,HDOT
C
      EL      = X(8)                                ! set the elevator
      CALL    F(TIME,X,XD)                          ! call aircraft model
      XD(9)   = THCOM - THETAD                     ! pitch error
      PI     = XD(9) + .2*X(9)                      ! prop. + integral signal
      XD(10)= PI - 14.0*X(10)                      ! phase-lead
      PL     = PI - 12.6*X(10)                      ! phase-lead output
      UE     = 1.0*QD - 40.0*PL                      ! inner-loop rate feedback
      XD(8)   = 10.0*(UE-X(8))                     ! elevator actuator
      RETURN
      END

```

Fig. 4.6-11 Pitch-attitude controller subroutine.

A complete autopilot design can be accomplished much more quickly if rather than augmenting the coefficient matrices with controller dynamics at each stage, the linearization program is used to determine coefficient matrices for a controller subroutine plus the aircraft model. A subroutine that implements the pitch-attitude autopilot is given in Fig. 4.6-11.

The same controller parameters are used as in Example 4.6-2 because there is negligible change in the aircraft dynamics (only the altitude has changed, from 50 ft to 750 ft). Controller states x_8 through x_{10} must be added to the trim data file before linearizing the complete dynamics. The elevator state x_8 must be set to the trimmed elevator setting, but the other controller states do not directly affect any nonlinear dynamics and can have zero initial conditions. By selecting control input five (THCOM) and output eight (THETAD) with the linearization program, the closed-loop pitch attitude transfer function can be checked against Example 4.6-2. Note that the altitude state is now included in the dynamics.

The d control loop cannot hold the required trajectory without closing the speed (auto-throttle) loop, but the speed loop can function independently of the d loop. Therefore, the speed loop should logically be closed first. A transfer function from throttle to speed can be obtained by using the input THTL and output VT. After canceling some very close pole-zero pairs, this transfer function is given by

$$\frac{v_T}{\delta_{th}} \approx \frac{10.10(s + 0.2736 \pm j0.1116)(s + 0.001484)}{(s + 0.2674 \pm j0.1552)(s + 0.0002005)(s + 0.06449)}. \quad (3)$$

Transfer function (3) can now be used to perform frequency-domain design of a speed-loop compensator.

When the throttle servo and engine lag model is added to the set of poles and zeros in (3), the Bode gain and phase plots (not shown) indicate a low-frequency gain of about 60 dB, infinite gain margin, and a phase margin

```

SUBROUTINE FC(TIME,X,XD)
DIMENSION X(*), XD(*)
COMMON/CONTROLS/THTL, EL, GAMREF, VCOM, THCOM, DCOM
COMMON/OUTPUT/AN, D2, D3, QBAR, AMACH, VT, ALPHA, THETAD, QD, H, HDOT
C
EL      = X(8)
THTL   = X(11)
CALL    F(TIME,X,XD)
C
XD(9) = THCOM - THETAD           ! pitch controller
PI     = XD(9) + .2*X(9)
XD(10)= PI - 14.0*X(10)
PL     = PI - 12.6*X(10)
UE     = 1.0*QD - 40.0*PL
XD(8) = 10.0*(UE-X(8))
C
EV     = VCOM - VT              ! speed controller
XD(12)= EV - 10.0*X(12)         ! phase lead
UT     = 10.0*(EV - 9.0*X(12))  ! phase lead
XD(11)= 0.2*(UT - X(11))       ! throttle servo + engine lag
RETURN
END

```

Fig. 4.6-12 Attitude and speed controller routines for auto-land.

of about 10° . A finite (but small) steady-state speed error is acceptable, so a Type 0 speed control can be used. Therefore, a conventional phase-lead compensator was chosen to improve the phase margin of this loop. The compensator

$$G_1(s) = \frac{10(s+1)}{(s+10)} \quad (4)$$

gives a phase margin of about 60° and retains the same If loop gain. This compensator was considered good enough for our purposes, and the speed control loop can now be added to the controller subroutine in Fig. 4.6-11.

Figure 4.6-12 shows the updated controller routine. The additional states x_{11}, x_{12} must be added to the trim file, with x_{11} initialized to the steady-state throttle setting, and the coefficient matrices for the d/u_θ transfer function can then be determined. After removing some canceling pole-zero pairs, the transfer function is found to be

$$\frac{d}{u_\theta} = \frac{675.2(s+1.40)(s+0.20)}{(s+2.021 \pm j1.770)(s+0.2725 \pm j0.1114)(s+4.409)(s+16.16)(s+0.001475)} \quad (5)$$

This transfer function has a relative degree of 5 (an asymptotic phase lag of 450°), and for satisfactory performance, phase-lead compensation can be anticipated. The pole at $s \approx -0.0015$ is the equivalent of the altitude pole that we observed in the attitude and altitude hold examples, and will again become a pure integration if the atmosphere model is disabled. Because the

pole is very close to the origin, and the low-frequency gain is high (over 60 dB), the d loop might be expected to approximate Type I behavior. This is not the case; the inner-loop pitch attitude controller normally has a finite input, and this will require a corresponding error signal in the d loop. The steady-state error will be exacerbated if we add a lead compensator (that cuts the If gain relative to the hf gain), and steady-state d errors of a few feet can be expected. Therefore, proportional plus integral compensation, together with a lead compensator, will be used.

PI and lead compensators were synthesized, and the design

$$G_2(s) = 1.0 \times \frac{(s + 0.18)(s + 0.5)}{s(s + 5)} \quad (6)$$

was derived. When the loop is closed the integrator pole and the pole from $s = -0.0015$ will circle to the left in the s -plane, to terminate eventually on the PI zero and the zero at $s = -0.20$. Therefore, one can attempt to speed up the response by moving the PI zero to the left, or by moving the PI zero to the right so that a closed-loop pole approaches it very closely (thus ensuring a small residue in the slow pole). The position $s = -0.18$, for the PI zero, was obtained by examining the closed-loop poles and zeros and the step response. The lead compensator was adjusted to obtain a compromise between the gain and phase margins, and these were, respectively, 14.7 dB and 51.6° . Figures 4.6-13a and b show the Bode plots; note that the pole at $s = -0.0015$ causes

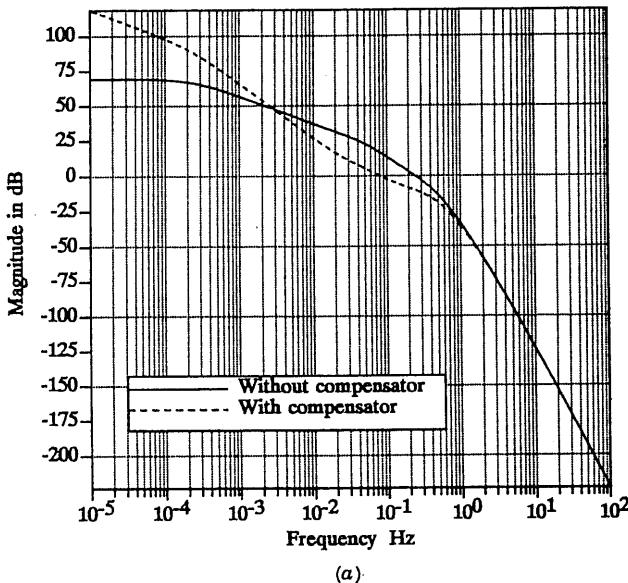


Fig. 4.6-13 (a) Bode magnitude plot for auto-land d -loop controller.

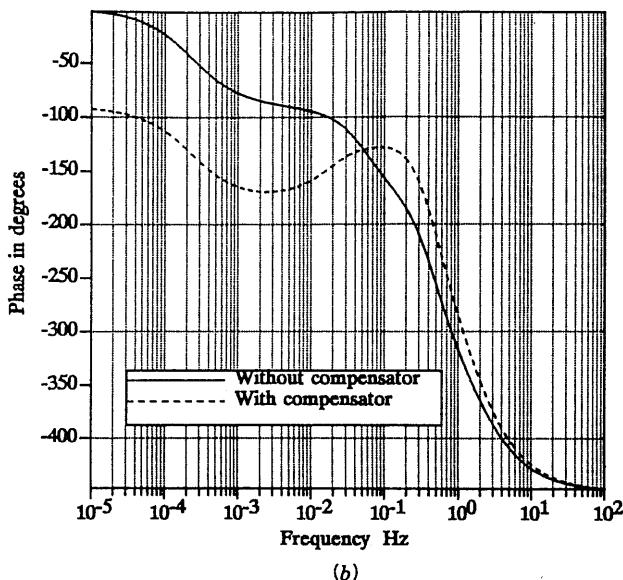


Fig. 4.6-13 (b) Bode phase plot for auto-land d -loop controller.

the low-frequency phase lag to approach -180° (like a Type II system) before the lead compensation begins to take effect.

Figure 4.6-14 shows the complete controller subroutine, with this compensator added and the d -loop closed. When the complete closed-loop controller and aircraft are linearized, and some close poles and zeros are canceled, the principal closed-loop transfer functions are

$$\frac{v_T}{v_c} \approx \frac{20.20(s + 1)}{(s + 7.627)(s + 1.280 \pm j0.9480)} \quad (7)$$

$$\frac{d}{d_c} \approx \frac{677.0(s + 1.40)(s + 0.50)(s + 0.20)(s + 0.180)}{(s + 16.2)(s + 5.16 \pm j1.65)(s + 1.38 \pm j1.69)(s + 0.292 \pm j0.348)(s + 0.179 \pm j0.0764)} \quad (8)$$

Note that in the d transfer function, the slowest pair of complex poles is close to terminating on the zeros at $s = -0.18$ and $s = -0.20$. The step responses could be evaluated by a linear simulation using the closed-loop state equations, but these are not particularly relevant to the auto-land function. Instead, a nonlinear simulation of the glide-path descent will be illustrated.

Figure 4.6-15 shows a driver program, FLY, used to call the generic time-response program, TRESP, which in turn calls the controller FC (and

```

SUBROUTINE FC(TIME,X,XD)
DIMENSION X(*), XD(*)
COMMON/CONTROLS/THTL, EL, GAMREF, VCOM, THCOM, DCOM
COMMON/OUTPUT/AN,D2,D3,QBAR,AMACH,VT,ALPHA,THETAD,QD,H,HDOT
C
EL      = X(8)
THTL   = X(11)
CALL    F(TIME,X,XD)
C
XD(13)= DCOM - X(7)          ! d controller
DPI    = XD(13) + .18*X(13)  ! PI compensator
XD(14)= DPI - 5.0*X(14)     ! Phase lead
THCOM = DPI - 4.5*X(14)      ! pitch command
C
XD(9) = THCOM - THETAD       ! pitch controller
PI    = XD(9) + .2*X(9)
XD(10)= PI - 14.0*X(10)
PL    = PI - 12.6*X(10)
UE    = 1.0*QD - 40.0*PL
XD(8) = 10.0*(UE-X(8))
C
EV    = VCOM - VT           ! speed controller
XD(12)= EV - 10.0*X(12)
UT    = 10.0*(EV - 9.0*X(12))
XD(11)= 0.2*(UT - X(11))
RETURN
END

```

Fig. 4.6-14 Auto-land *d*-loop controller subroutine.

```

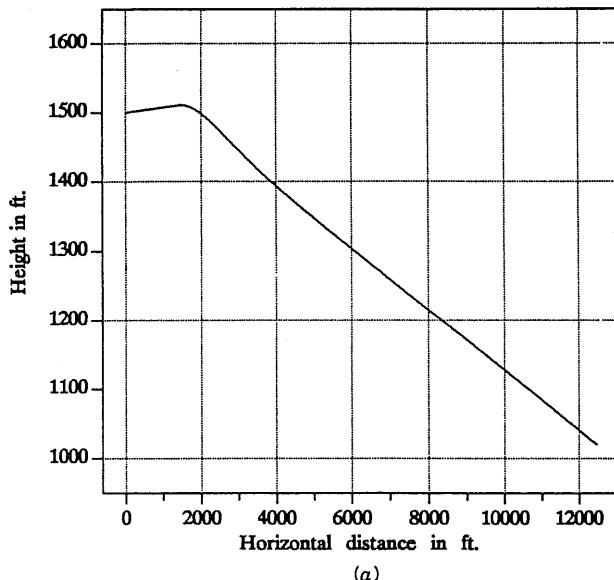
PROGRAM FLY
EXTERNAL FC,D
COMMON/PARAM/XCG,LAND
LAND= 1
XCG = .25
CALL TRESP(FC,D)
END
C
C
SUBROUTINE D(TIME,TS,X,XD,DFLAG)
LOGICAL DFLAG
DIMENSION X(*), XD(*)
COMMON/CONTROLS/THTL, EL, GAMREF, VCOM, THCOM, DCOM
COMMON/OUTPUT/AN,D2,D3,QBAR,AMACH,VT,ALPHA,THETAD,QD,H,HDOT
C
IF (TIME .GE. 5.0) THEN
GAMREF= -2.5
DCOM = 0.0
VCOM= 250.0
ELSE
GAMREF= THETAD - ALPHA
DCOM = 0.0
VCOM = 250.0
END IF
RETURN
END

```

Fig. 4.6-15 Driver programs for glide-path simulation.

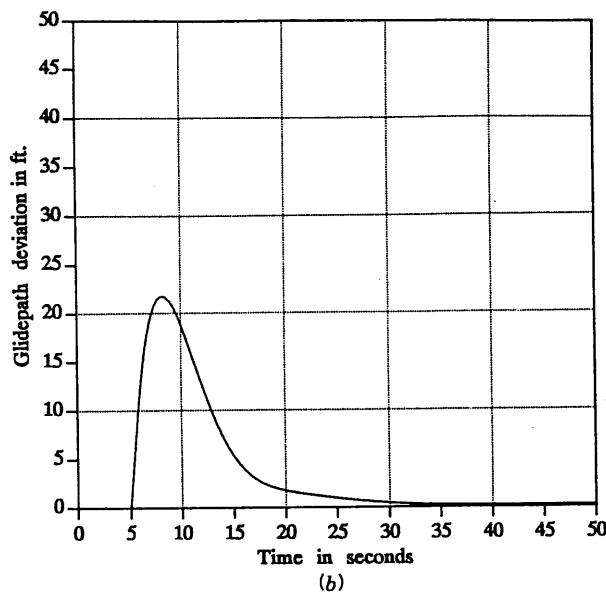
therefore the airplane F) and the discrete time command routine D. A steady-state trim condition was obtained by flying the controller and aircraft in level flight at 1500 ft, with $V_c = 250$ ft/s and $\gamma_R = 0^\circ$, until all of the transients due to improper initialization had died out. The state vector was then filed and used to initialize a glide-path simulation. As shown in subroutine D, the glide path is assumed to be intercepted 5 s after the start of the simulation; a fly-down command, $\gamma_R = -2.5^\circ$, is then received and this will start a buildup of the deviation, d , from the glide path. When the correct flight-path angle is achieved, the rate of change of d will become zero, but d itself will not become zero until the controller has driven the aircraft back to the imaginary line in the sky representing the glide path.

Figure 4.6-16a shows the aircraft trajectory obtained from the nonlinear simulation; only 50 s of flight were simulated because steady-state conditions were well established by that time. Figure 4.6-16b shows the deviation from the glide path; note that the behavior of the initial d error will be different depending on the initial conditions existing in the d -loop compensator at the time of glide-path acquisition. Figure 4.6-16c shows the behavior of alpha, pitch attitude, and elevator deflection, and Fig. 4.6-16d and e show, respectively, the corresponding variation of throttle position and airspeed. It is evident that the airplane is driven onto the glide path quickly and smoothly, without large excursions in pitch attitude. Airspeed is held very nearly constant and the throttle is changed smoothly and gently. The elevator shows some rapid excursions, which could be reduced by slowing down the d loop, with very little loss of performance.

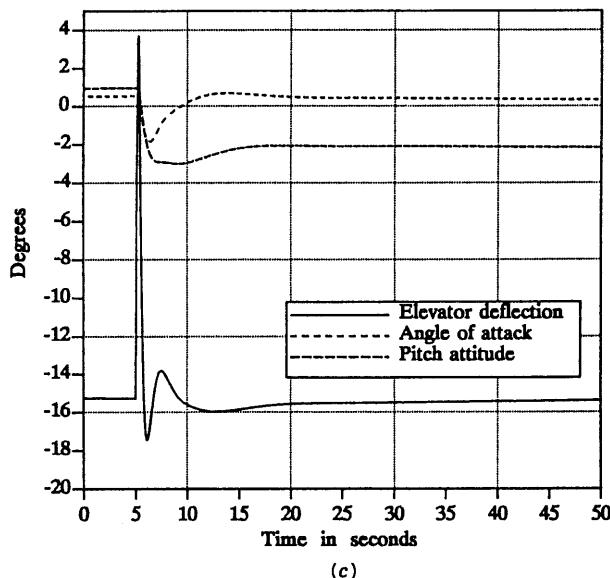


(a)

Fig. 4.6-16 (a) Simulated trajectory with glide-path controller.

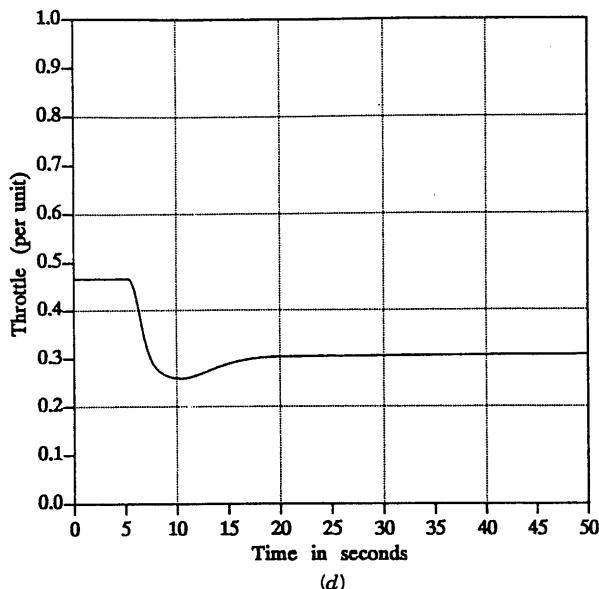


(b)

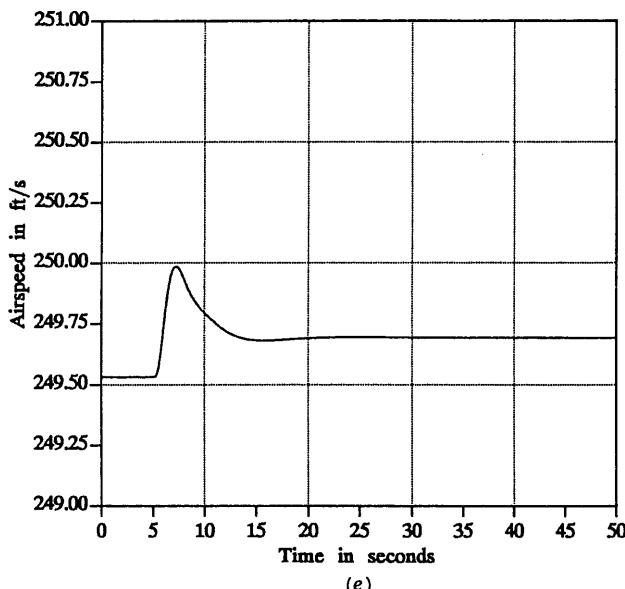


(c)

Fig. 4.6-16 (continued) (b) Glide-path deviation during acquisition. (c) Alpha, theta, and elevator variations during glide-path acquisition.



(d)



(e)

Fig. 4.6-16 (continued) (d) Throttle response during glide-path acquisition. (e) Airspeed variation during glide-path acquisition.

It should be noted that this control system is still a long way from a practical design. A practical design must take into account the imperfections of the various sensors (noise, bias, delay, etc.), and must be robust to changes in aircraft weight, wind conditions, and so on.

The final component of this design is the automatic "flare" control that makes the aircraft begin to level out as the altitude approaches zero, and touch down with an acceptably small rate of descent. This is described in the following subsection. ■

Flare Control

At an altitude between 20 and 70 ft above the end of the runway the automatic landing system must start to reduce the rate of descent of the aircraft, achieve the correct pitch attitude for landing, and begin to reduce the airspeed. This portion of the trajectory is called the landing flare, and the geometry of the flare is illustrated in Fig. 4.6-17. On the glide path the aircraft is descending at a rate of 10 ft/s or greater and will hit the ground hard, within 5 to 10 s, if the flare is not executed. The rate of descent must be reduced to less than about 2.0 ft/s. This should occur as the altitude reaches zero, to allow a gentle but positive landing within a reasonable distance from the start of the runway. The pitch attitude angle increases by a few degrees at the start of the flare, and depending on the type of aircraft, the pitch attitude at touchdown will usually be in the range of 0 to 5°. The principal sensor is a radar altimeter, and satisfactory performance requires tight control of the aircraft altitude. This is illustrated in the following design example.

Example 4.6-5: Automatic Flare Control. Modern digital-computer-based flight control systems allow functions such as throttle reduction to be programmed as a function of altitude. This indirectly controls the attitude of the aircraft during the flare. We shall simply assume that in Fig. 4.6-10, the speed

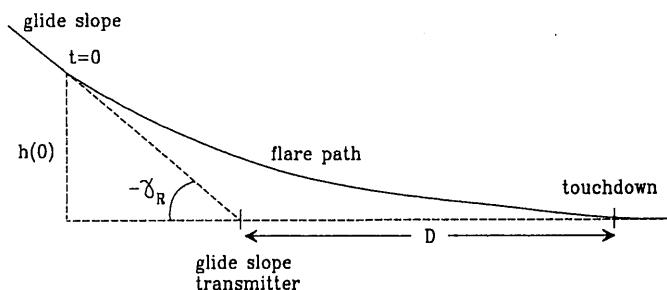


Fig. 4.6-17 Automatic landing flare path geometry.

loop continues to operate with the same command input, and the d loop is replaced by an altitude control loop when the flare is initiated.

Using the controller routine from Example 4.6-4, the d loop was removed and the controller and aircraft were linearized under the glide-path flight conditions, at an altitude of 50 ft (this is not critical). The h/u_θ transfer function, after removing some canceling pole-zero pairs, was found to be

$$\frac{h}{u_\theta} \approx \frac{705.1(s + 1.40)(s + 0.20)}{(s + 16.19)(s + 4.20)(s + 2.117 \pm j1.801)(s + 0.2761 \pm j0.1094)(s + 0.001440)}. \quad (1)$$

This transfer function is almost the same as the d transfer function [equation (5)] in Example 4.6-4, and the same compensator can be considered. However, the flare requires a very tight control loop, and the compensator was redesigned. The compensator gain was boosted by a factor of 4 and the phase lead adjusted to optimize the gain and phase margins. This led to the compensator

$$G_2(s) = 4.0 \times \frac{(s + 0.18)(s + 1.0)}{s(s + 10.0)}, \quad (2)$$

which gives gain and phase margins of 7.8 dB and 33° , respectively. The performance with these rather small margins will be illustrated; in practice, the effects of worst-case aircraft weight, runway altitudes, and wind gusts would be considered and a compromise made between performance and a more complex controller design.

As a first attempt at producing a controlled flare, assume that the altitude control loop is to be driven by a command that decays exponentially from h_0 ft to zero. Theoretically, it will take infinite time to reach zero altitude, but as a practical assumption, the exponential decay can be assumed to be complete in four time constants. A constraint on the exponential flare command is that it should start the flare with the same rate of change of altitude as the vertical velocity on the glide path. Thus we have the following equations and constraints for the exponential model that generates the flare command:

$$h = h_0 e^{-t/\tau}; \quad \text{therefore, } \dot{h} = -\frac{h}{\tau} \quad (\text{model}) \quad (3)$$

$$h(0) = -\tau \dot{h}(0) = \tau V_T \sin \gamma_R \quad (\text{constraint}). \quad (4)$$

The flight-path angle is very shallow throughout the flare, so the horizontal component of the aircraft's velocity, $V_T \cos \gamma$, is approximately equal to V_T and will also be assumed constant (the speed-loop command will be constant). Therefore, the horizontal distance traveled during four time con-

stants is approximately $4\tau V_T$. Figure 4.6-17 shows that the total horizontal distance, from the start of the flare to touchdown, is given by

$$4\tau V_T = (h_0/\tan \gamma) + D = \tau V_T \cos \gamma + D. \quad (5)$$

Since γ is a small angle, $\cos \gamma \approx 1.0$, and this equation leads to

$$\tau = D/3V_T. \quad (6)$$

If, for example, the distance D from the glide-slope transmitter to touchdown is 2000 ft and $V_T = 250$ ft/s, (6) and (4) give

$$\tau = 2.667 \text{ s}, \quad h_0 = 29.1 \text{ ft}. \quad (7)$$

Figure 4.6-18 shows how the altitude control and flare commands can be incorporated in the glide-path controller. The figure shows the way in which

```

SUBROUTINE FC(TIME,X,XD)
DIMENSION X(*), XD(*)
COMMON/CONTROLS/THTL, EL, GAMREF, VCOM, THCOM, DCOM
COMMON/OUTPUT/AN, D2, D3, QBAR, AMACH, VT, ALPHA, THETAD, QD, H, HDOT, HC, EH
C
EL      = X(8)
THTL   = X(11)
CALL    F(TIME,X,XD)
C
IF (H .GT. 29.08) THEN
  ERR   = DCOM - X(7)                      ! d controller
  XD(14)= ERR - 5.0*X(14)
  DPL   = 1.0*(ERR - 4.5*X(14) )
  SAVE  = TIME
ELSE
  HC    = 29.08*EXP( -(TIME-SAVE)/2.6667) ! flare alt. command
  ERR   = HC - H                           ! command following error
  ERR   = -HDOT -.375*H                     ! model-following control
  XD(14)= ERR - 40.0*X(14)                  ! model-foll. phase-lead
  DPL   = 10.0*(ERR - 36.0*X(14) )          ! phase lead
  XD(14)= ERR - 10.0*X(14)                  ! command foll. phase-lead
  DPL   = 4.0*(ERR - 9.0*X(14) )             ! phase lead
END IF
C
XD(13) = DPL                                ! error integrator
THCOM  = XD(13) + .18*X(13)                  ! PI control
XD(9)  = THCOM - THETAD                      ! pitch autopilot
PI     = XD(9) + .2*X(9)
XD(10)= PI - 14.0*X(10)
PL     = PI - 12.6*X(10)
UE     = 1.0*QD - 40.0*PL
XD(8)  = 10.0*(UE-X(8))
C
EV     = VCOM - VT                          ! speed controller
XD(12)= EV - 10.0*X(12)
UT     = 10.0*(EV - 9.0*X(12))
XD(11)= 0.2*(UT - X(11))
RETURN
END

```

Fig. 4.6-18 Auto-land controller with flare control added.

the flare control is initialized to ensure a smooth transition into the flare. Note also that the d controller and the flare controller share the same PI compensator, and the integrator tends to smooth out the commands into the pitch autopilot during the transition. These concerns about transient-free transitions are important in the practical implementation of aircraft controllers. The code labeled "model following" is temporarily disabled, and will be explained later in this example.

A nonlinear simulation of this controller, with the transport aircraft model, was started from an initial state corresponding to an altitude of approximately 100 ft on the glide-path simulation of Example 4.6-4. Figure 4.6-19a shows the trajectory, with the flare command automatically initiated when the altitude reaches 29.1 ft. At first the aircraft trajectory begins to diverge from the flare command because the command model contains derivatives of all orders. Later, as the rate of change of the command decreases, the trajectory begins to follow the command more closely, and the performance is quite good. The flare starts at about 6.2 s into the simulation and touchdown occurs at about 13.5 s, during which time the aircraft covers about 1800 ft. Figure 4.6-19b shows the vertical component of the aircraft velocity; at touchdown this is about -0.49 ft/s, and this constitutes a very smooth landing. Figure 4.6-19c shows the variation of pitch attitude during the flare; this is quite smooth and the aircraft is essentially level for approximately 5 s before touchdown.

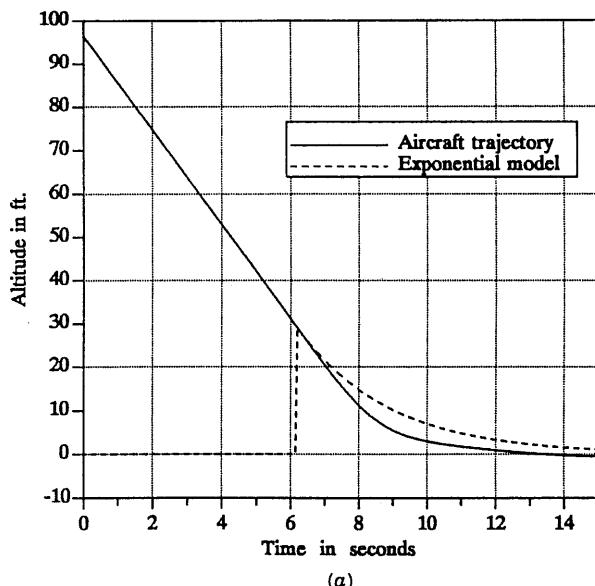


Fig. 4.6-19 (a) Flare trajectory with an exponential command.

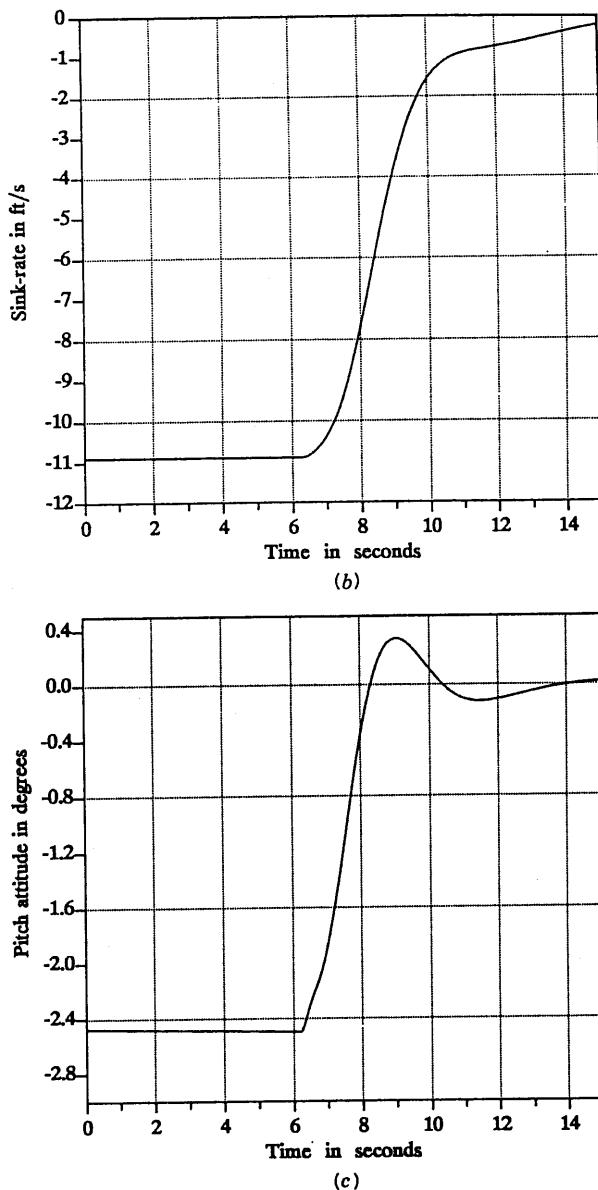


Fig. 4.6-19 (continued) (b) Sink rate during the exponential flare. (c) Pitch attitude variation during the flare.

An alternative flare control scheme will now be considered; this is a form of *implicit model following control*. The control law is indicated by the in-line comments in Fig. 4.6-18 and replaces the command-following error in the line immediately preceding it. The error signal is given by

$$e = -\left(\dot{h} + \frac{h}{\tau}\right). \quad (8)$$

If the forward path of the control system is of wide bandwidth and high gain, the error signal will remain very small during the flare. Therefore, the altitude will be constrained to satisfy the differential equation

$$\dot{h} + \frac{h}{\tau} = 0, \quad (9)$$

whose solution is

$$h = h(0)e^{-t/\tau}. \quad (10)$$

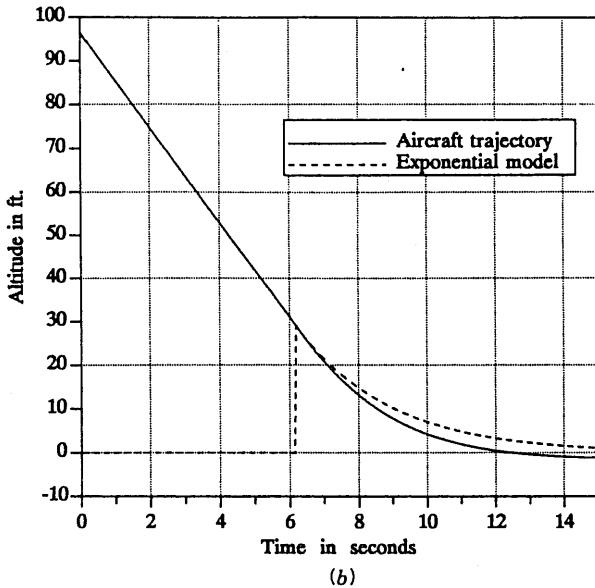
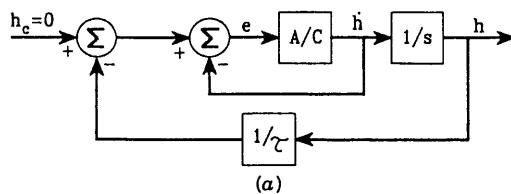
Equation (10) is the "implicit model," and it provides the exponential flare that we wished to achieve.

Another viewpoint on the implicit model following is shown in Fig. 4.6-20a, in which the feedback satisfies (8). The \dot{h} feedback provides an inner-loop \dot{h} control system; this is followed by the integration that produces h , and an outer-loop feedback law $u = -h/\tau$. If the dynamics of the \dot{h} control system can be neglected, this constitutes negative feedback around an integrator, through a gain $1/\tau$. The result is a simple lag transfer function, which has the exponential response (10) to its initial conditions.

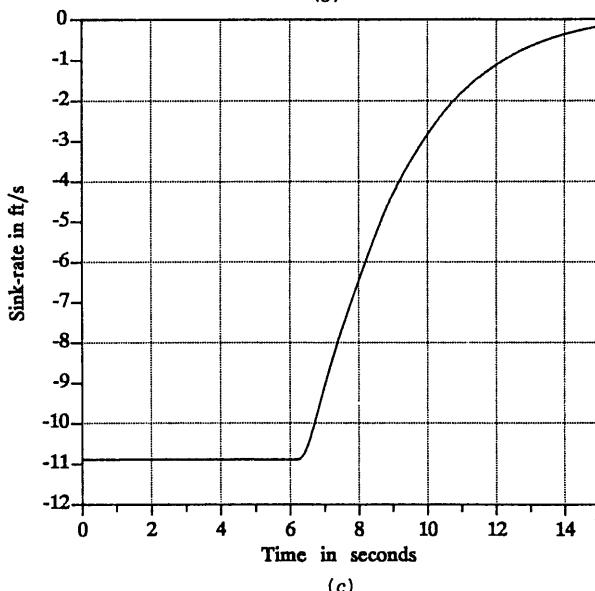
The model-following controller was designed by developing the compensator:

$$G_2(s) = 10.0 \times \frac{(s + 0.18)(s + 4.0)}{s(s + 40.0)}, \quad (11)$$

for the h -dot loop. The simulation results for implicit model-following flare control are shown in Fig. 4.6-20b, c, and d. Figure 4.6-20b shows that the aircraft touches down sooner, at about 12.5 s, or approximately 1600 ft from the start of the flare. The error characteristics are noticeably different from the command-following case; the error does not increase so alarmingly at the start of the flare, and the model could be followed more closely if the h -dot dynamics could be made faster.



(b)



(c)

Fig. 4.6-20 (a) Implicit model-following control. (b) Flare trajectory with implicit model-following control. (c) Sink rate with implicit model-following control.

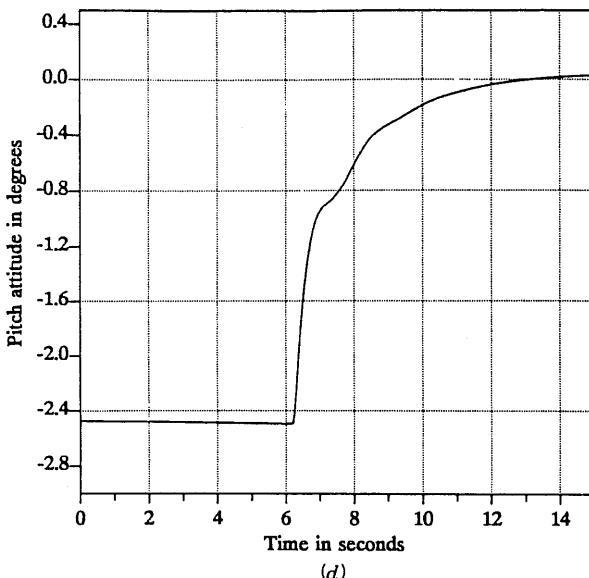


Fig. 4.6-20 (continued) (d) Pitch attitude with implicit model-following control.

Figure 4.6-20d shows that there is a smoother variation of pitch attitude, while Fig. 4.6-20c shows that the landing is less gentle (sink rate at touchdown is 0.86 ft/s) but still good. This controller is used in practice, rather than command following, because of the more desirable error characteristics. No allowance has been made for "ground effect" in this example. There is a significant increase in lift coefficient as the aircraft approaches within one wingspan of the ground; this tends to make the aircraft float along the runway and must be taken into account in the flare control design. ■

Bank-Angle Hold Autopilots

In its simplest form, as a wing leveler, the bank-angle hold autopilot has a history going back to the experiments of Elmer Sperry (see Section 4.1). A sensor incorporating an attitude reference, such as a gyroscope, is used to sense deviations from a reference attitude in the aircraft $y-z$ plane. Feedback of the deviation signal to the ailerons can then be used to control the bank (or roll) attitude of the aircraft. The bank-angle feedback gives the aircraft positive stiffness in roll and stabilizes the spiral mode. Thus the bank-angle hold autopilot provides a pilot-relief function for long flights and eliminates the danger of the pilot being caught unaware by a coordinated spiral motion toward the ground.

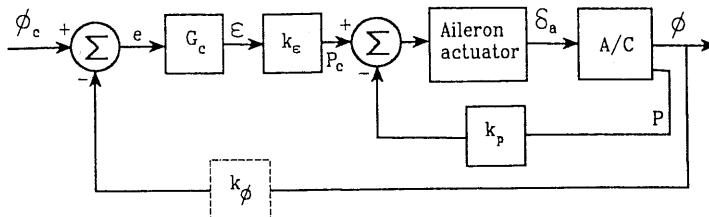


Fig. 4.6-21 Bank-angle hold autopilot.

If the aircraft is held at some attitude other than wings level, additional control systems must be used to control sideslip and pitch rate, so that a coordinated turning motion is produced. Depending on the commanded pitch rate, the aircraft may gain or lose altitude in a turn. If a means of varying the bank-attitude reference is provided, the aircraft can be steered in any direction by a single control. These control systems can provide the inner loops for other autopilots that allow an aircraft to fly on a fixed compass heading, or home on a radio-navigational beam in the presence of cross-winds. Such systems will be described later.

Figure 4.6-21 shows a block diagram of a bank-angle hold autopilot. High-performance aircraft virtually always have available a roll-rate gyro for use by a SAS or CAS, and this can be used to provide inner-loop rate damping for the autopilot. If the roll-rate gyro is not available, then for good performance, a compensator is needed in the bank-angle error path. There is usually no requirement for precise tracking of bank-angle commands, so Type 0 bank angle control can be used. By the same token, the *velocity error* due to straight roll-rate feedback (i.e., no washout) is not important, particularly since the roll rate is not usually sustained for very long.

If the aircraft has strong roll-yaw coupling, the bank-angle-to-ailerons feedback must be considered as part of a multivariable design, as was done in Sections 4.4 and 4.5. This is often not the case, and in the lateral transfer function, the poles associated with the directional controls are approximately canceled by zeros. The transfer function for the bank-angle loop is then determined by the roll-subsidence pole, the spiral pole, and the actuator and compensator (if any) poles. If roll-rate feedback is used, in conjunction with the bank-angle feedback, there is good control over the position of the closed-loop poles and quite large amounts of feedback can be used. A bank-angle hold autopilot design will now be illustrated by applying bank-angle feedback around the lateral-directional CAS from Example 4.5-4.

Example 4.6-6: A Bank-Angle-Hold Autopilot. This example will use the controller subroutine from Example 4.5-4 and linearize the complete dynamics (controller plus aircraft) with bank angle as an output, under the same

flight conditions. The closed-loop transfer function from the roll-rate command to the bank angle is found to be

$$\frac{\phi}{p_c} = \frac{182.7(s + 13.09)(s + 2.429 \pm j2.241)(s + 1.540)}{(s + 13.42)(s + 2.386 \pm j2.231)(s + 1.575)(s + 0.002116)(s + 11.78 \pm j10.96)}, \quad (1)$$

or, approximately,

$$\frac{\phi}{p_c} = \frac{182.7}{(s + 11.78 \pm j10.96)(s + 0.002116)}. \quad (2)$$

In this transfer function the complex pole pair arose from the actuator pole and the roll-subsidence pole, and the real pole is the spiral pole. When the bank-angle feedback loop is closed around (1), the root-locus plot is as might be expected from (2); the spiral pole moves to the left and the complex poles move to the right. The root-locus plot is shown in Fig. 4.6-22.

A bank-angle feedback gain of $k_\phi = 5.0$ gave the complex poles a damping ratio of $\zeta = 0.71$ (at $s = -8.88 \pm j8.93$), and the real pole was at $s = -5.4$. This gain k_ϕ was moved to the forward path ($k_e = k_\phi$) to allow unity feedback. The bank-angle control loop is well damped but unrealistically fast. The commanded attitude will be tightly controlled in the steady state, but the aileron actuators will be driven into rate limiting if abrupt bank-angle commands are applied. This control system will be used in the next subsection in a nonlinear simulation.

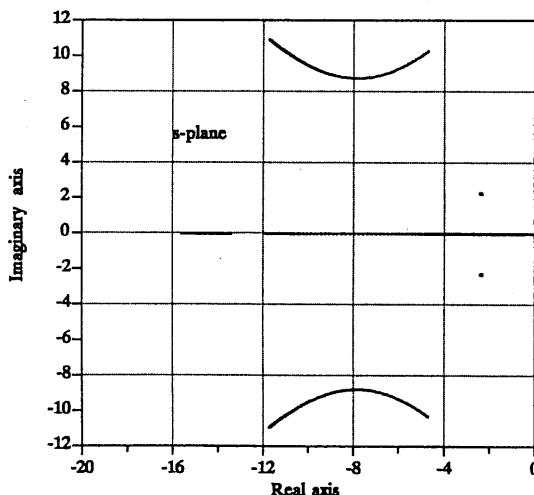


Fig. 4.6-22 Root-locus plot for the bank-angle hold autopilot. ■

Turn Coordination and Turn Compensation

The basic requirement for a coordinated turn is zero lateral acceleration (and also zero acceleration along the body x -axis). The sideslip angle may not be exactly zero in this condition because of some small asymmetry in the aircraft, such as asymmetric thrust or the effects of the angular momentum of spinning rotors. The relative merits of lateral acceleration or sideslip feedback for turn coordination were pointed out in Section 4.5, and the lateral-directional CAS was designed with lateral acceleration feedback.

There is, of course, an additional requirement for turning motion, and this is a nonzero pitch rate. The pitch rate required for a constant-altitude turn can be readily calculated. When the kinematic equations (1.4-15) are inverted, we have

$$Q = \dot{\theta} \cos \phi + \dot{\psi} \sin \phi \cos \theta \quad (4.6-3)$$

$$R = -\dot{\theta} \sin \phi + \dot{\psi} \cos \phi \cos \theta. \quad (4.6-4)$$

In a constant-altitude turn, the condition $\dot{\theta} = 0$ applies (see Section 2.5), so the pitch rate is related to the yaw rate by

$$Q = R \tan \phi. \quad (4.6-5)$$

This pitch rate can be calculated in the flight control computer, using the signals from the yaw-rate gyro and bank-angle sensor, and fed to the pitch rate control system as a command.

Equation (4.6-5) provides "turn compensation" that allows the aircraft to be maneuvered, through coordinated turns, by applying commands to the bank-angle control system. This is open-loop control of the pitch rate, and because the required pitch rate does not change rapidly with flight-path angle (near to $\gamma = 0$), we shall see that it does not provide perfect turn compensation in a specific example. An alternative method of providing turn compensation is to use altitude feedback to the elevator. In general, this would be expected to be a slower but more precise system. An example of turn compensation using (4.6-5) will now be given, by adding to the lateral-directional control system of Example 4.5-4 the bank-angle control of Example 4.6-6.

Example 4.6-7: A Bank-Angle-Steering Control System. The controller routine for the lateral-directional CAS is easily modified to include the bank-angle feedback and the turn compensation; and the controller and command routines are shown in Fig. 4.6-23. The same initial condition data used for the lateral-directional CAS was used with this controller, and a simulation was flown using the discrete-time commands shown in Fig. 4.6-23. Figure 4.6-24a shows the ground track of the aircraft in response to these commands. The altitude decreased by about 600 ft during the 30-s simulation

```

SUBROUTINE FC(TIME,X,XD)
DIMENSION X(*), XD(*)
REAL M
COMMON/CONTROLS/THTL,EL,AIL,RDR,BANK,QCOM,RCOM
COMMON/OUTPUT/AN,AY,AX,QBAR,M,ALPHA,BETA,PHID,THTAD,
& PD, QD, RD, RF
C
      EL = X(1)
      AIL= X(4)
      RDR= X(5)
      CALL F(TIME,X(7),XD(7))
      QCOM= RD*TAN(PHID/57.29578)           ! turn compensation
      XD(3)= QCOM - QD
      U= -( 1.5*X(3) - .5*QD - .08*X(2) )
      XD(1)= 20.2*(U-X(1))
      XD(2)= 10.0*( ALPHA - X(2) )
      UA = -1.0*(BANK-PHID-.2*PD)          ! bank angle control law
      XD(4)= 20.2*( UA - X(4) )
      ARI = (0.13*X(2) - 0.7)*UA
      RS = RD - PD*X(2)/57.3
      XD(6)= RS - X(6)
      ERR= RCOM + .8*XD(6) + 10.0*AY
      XD(5)= 20.2*( ERR + ARI - X(5) )
      RETURN
      END
C
      SUBROUTINE D(TIME,TS,X,XD,DFLAG)
      DIMENSION X(*),XD(*)
      COMMON/CONTROLS/THTL,EL,AIL,RDR,BANK,QCOM,RCOM
      LOGICAL DFLAG
C
      IF (TIME .GE. 20.0) THEN
        BANK= -70.0
      ELSE IF (TIME .GE. 5.0) THEN
        BANK= 70.0
      ELSE
        BANK = 0.0
        QCOM = 0.0
        RCOM = 0.0
      END IF
      RETURN
      END

```

Fig. 4.6-23 Code for simulating bank-angle steering.

because of the open-loop turn compensation and the finite control error of the lateral-directional control systems. The speed decreased by about 200 ft/s during the simulation, because of the maneuvers.

Figures 4.6-24*b*, *c*, and *d* show, respectively, the fast well-damped bank-angle response, the angle of attack, and the sideslip angle. The aileron and rudder deflections are shown in Fig. 4.6-24*e* and *f*; these show short-duration deflections that are well beyond the limits of the control surfaces and raise the following important points.

First, the simulation results may be unrealistic if the control-surface rate and deflection limits are not modeled (see Section 4.7). Second, control system limiting will be caused by the abrupt large-amplitude commands and the high gains that have been used in the bank-angle control. This is not

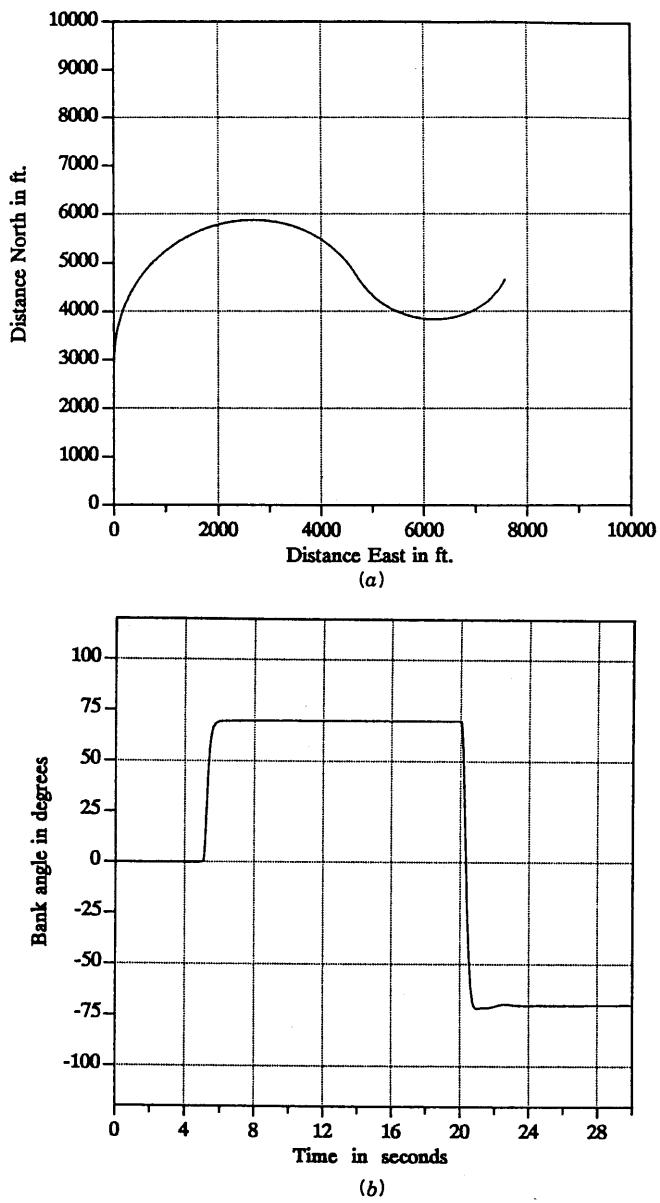
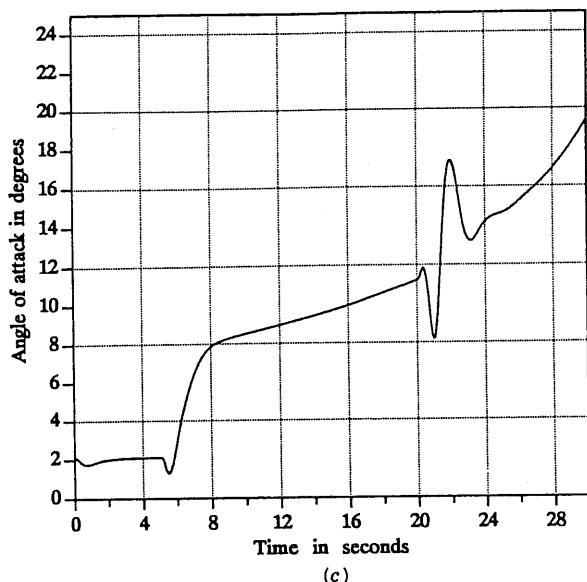
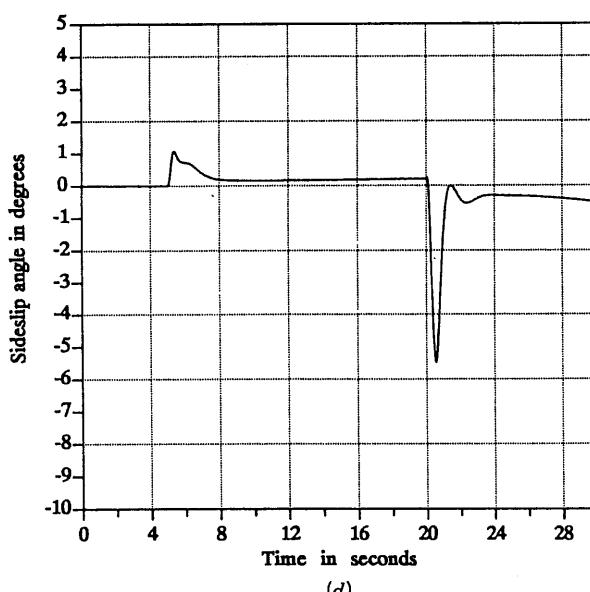


Fig. 4.6-24 (a) Ground track during bank-angle steering. (b) Bank-angle variation during simulation.

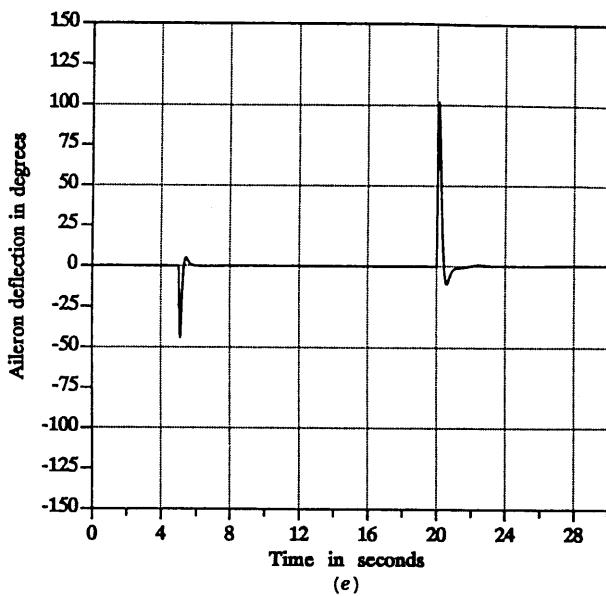


(c)

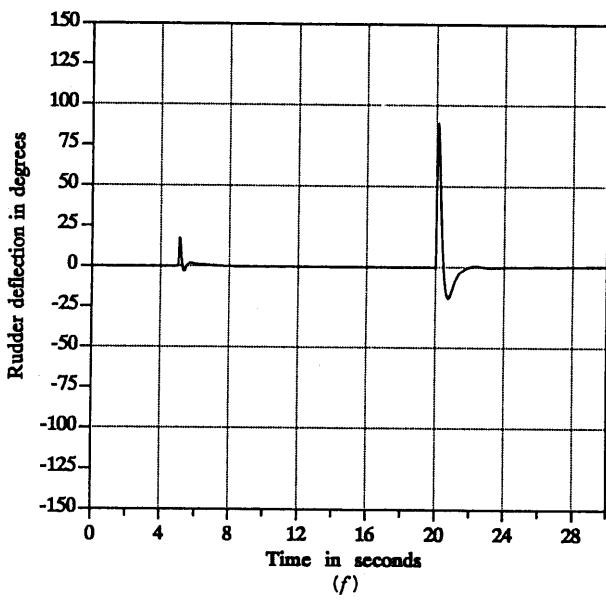


(d)

Fig. 4.6-24 (continued) (c) Angle-of-attack variation during simulation. (d) Sideslip variation during simulation.



(e)



(f)

Fig. 4.6-24 (continued) (e) Control deflections during simulation. (f) Control deflections during simulation.

necessarily a problem if the system response is still acceptable, since the fastest possible roll response may be desired, and the high gains also provide a small control error for low-amplitude inputs. Third, the airplane flying qualities are the most important consideration, and in this situation the stick prefilter and the maximum roll rate of the airframe will play a major part in determining the pilot's opinion of the roll performance. ■

Autopilot Navigational Modes

Automatic navigation is an important autopilot function for both military and civil aircraft, and the most important systems will be briefly summarized. A *heading-hold* autopilot is designed to hold the aircraft on a given compass heading. The conventional method of implementing this autopilot is to close an additional yaw-angle feedback loop around the bank-angle control system (including turn compensation) that was illustrated in the preceding subsection. Figure 4.6-25 shows the arrangement. The transfer function relating heading angle to bank angle is the linearized equation obtained from (3.4-7). The transfer function can equally well be obtained by means of the linearization program. Note that the transfer function gain is inversely proportional to speed. An investigation of the root locus for the heading angle loop, and the effects of flight conditions, will be left to the reader (see also Blakelock [1965] and Roskam [1979]).

A *VOR-hold* (VHF omni range) autopilot is an autopilot designed to home on an omnidirectional radio beacon. The heading-angle hold system (including turn compensation, etc.) is used to implement this autopilot, and Fig. 4.6-26 shows how this is done. The transfer function derived from the geometry of the beam following is similar to that derived for the automatic-landing longitudinal control system. The system normally requires proportional plus integral compensation, and possibly lead compensation also. Again, it is left to the reader to investigate further [Blakelock, 1965; Roskam, 1979].

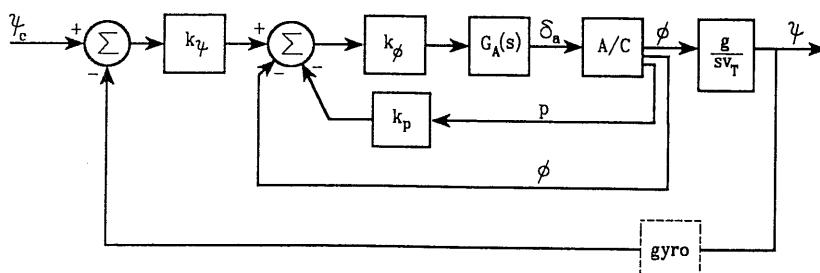


Fig. 4.6-25 Heading-hold autopilot.

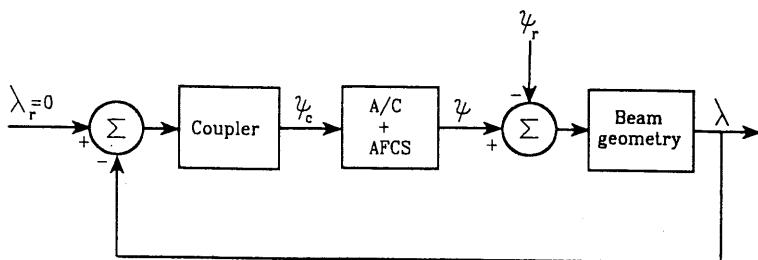


Fig. 4.6-26 VOR-hold autopilot.

A specialized military autopilot that is particularly interesting is a *terrain-following terrain-avoidance* (TFTA) autopilot. This system uses the aircraft's radar or a separate radar carried underneath the aircraft (as in the LANTIRN system). The radar provides guidance commands to fly at constant height (e.g., 100 to 400 ft) above the Earth's surface, at high speed. The fly-up, fly-down commands are usually applied to a *g-command* control system as described in Section 4.5, and the lateral-directional guidance commands are applied to a bank-angle steering control system as in Example 4.6-7.

4.7 NONLINEAR SIMULATION OF LIMITERS

The linear designs illustrated in previous sections are only the first stage in the design of complete aircraft control systems. At the second stage, nonlinear control-system elements must be modeled, the performance evaluated by simulation, and adjustments made to the design. We shall neglect effects that are dependent on the specific implementation, such as stiction and dead-zone effects in a hydraulic element. Instead, the generic effects that have a major effect on performance will be considered. These include the rate and deflection limits of control surface actuators, and limiters that are deliberately introduced to avoid exceeding airframe limits.

It is evident from the design examples that have been presented that if the actuator poles are not well to the left in the *s*-plane, it may be difficult to achieve satisfactory damping or natural frequency for the aircraft modes. Phase-lead compensation may be introduced in an effort to overcome this problem. However, since phase lead is always accompanied by a relative increase in high-frequency gain, this may cause the control actuators to reach their rate limits during fast transients. Therefore, it is important to model rate limiting and to perform a nonlinear simulation to see if the performance is significantly different from that of the linear model. If this is the case, a

redesigned control system with a faster (also heavier and more expensive) actuator may be required.

It is equally important to model actuator deflection limits. At low dynamic pressures large control surface deflections are needed, and the performance may become limited by control power rather than structural or human limits. If the deflection limits are not modeled, the low-speed performance cannot be evaluated accurately from a simulation. In designs where multiple control surfaces are involved it may be possible to modify the control system so that the deflections of different control surfaces are more nearly balanced.

Many aircraft control systems require *command limiters* to prevent the pilot from entering a dangerous flight condition. In the case of fighter aircraft, combat tactics may be influenced by knowledge of the conditions under which the control system limiters of opposing aircraft come into play. In Example 4.5-4 inertia coupling effects were described, and the need to limit the commanded roll rate at high alpha was explained. This is an example of the need for an alpha-dependent limiter on the pilot's roll command. We shall now extend this example to show how rate and deflection limits may be modeled, and a command limiter may be used.

Example 4.7-1: Simulation of a Controller with Limiters. Figure 4.7-1 shows how the control actuator rate and deflection limits may be added to the controller subroutine used in Example 4.5-4. The in-line comments make the controller listing largely self-explanatory. When a state variable reaches a limit, a nonzero derivative is allowed only if it is in the direction that takes the state variable off the limit. Also, integrators in the control loops are limited in the same manner, at the current output levels, once the actuators reach their limits.

The controller, with limiters, was used in a time-history simulation similar to that of Example 4.5-4, and the discrete-time commands are shown in Fig. 4.7-2. The integration step size was reduced to 1 ms in this simulation, to capture the action of the limiters accurately. This can make the execution quite slow on a personal computer, especially if a large number of variables are recorded [an IBM PS/2 50Z was used for the example, but a very much faster PS/2-70 (25 MHz) was used later to try out variations on the example].

A pull-up at $20^\circ/\text{s}$ is simulated, so that the angle of attack attains quite large values. When alpha reaches about 23° (Fig. 4.7-3a) a large roll-rate command is applied; this causes the aileron and rudder actuators to saturate almost immediately (Fig. 4.7-3b). The directional controls are then unable to control the sideslip tightly, and some of the angle of attack is rapidly converted to sideslip (Fig. 4.7-3c) through kinematic coupling. As the roll and yaw rates build up (Fig. 4.7-3d), the inertia coupling moment becomes strong. The elevator deflection then goes from a small negative value to its positive limit, as it tries to oppose the inertia coupling moment. Figure 4.7-3e shows

```

SUBROUTINE FC(TIME,X,XD)
DIMENSION X(*), XD(*)
REAL M
COMMON/CONTROLS/THTL,EL,AIL,RDR,QCOM,PCOM,RCOM
COMMON/OUTPUT/AN,AY,AX,QBAR,M,ALPHA,BETA,PHID,THTAD,PD,QD,
& RD,RF
DATA ERL,EDL,ARL,ADL,RRL,RDL/60.0,25.0,80.0,21.5,120.0,30.0/
C
EL = X(1)
AIL= X(4)
RDR= X(5)
CALL F(TIME,X(7),XD(7))
XD(2)= 10.0*( ALPHA - X(2) )
XD(3)= QCOM - QD
U= -( 1.5*X(3) - .5*QD - .08*X(2) )
XD(1)= 20.2*(U-X(1))
C
IF( ABS(XD(1)) .GT. ERL) THEN
  XD(1)= SIGN(ERL, XD(1)) ! Elevator rate limit
  XD(3)= 0.0
END IF
IF(X(1) .GT. EDL) THEN
  X(1)= EDL ! Elevator +deflection limit
  IF(XD(1) .GT. 0.0) XD(1)= 0.0 ! Stop integrating positively
  IF(XD(3) .LT. 0.0) XD(3)= 0.0 ! clamp error integrator
ELSE IF (X(1) .LT. -EDL) THEN
  X(1)= -EDL ! Elevator -deflection limit
  IF(XD(1) .LT. 0.0) XD(1)= 0.0 ! stop integrating negatively
  IF(XD(3) .GT. 0.0) XD(3)= 0.0 ! clamp error integrator
ELSE
  CONTINUE
END IF
C
UA = -.2*(PCOM-PD)
XD(4)= 20.2*( UA - X(4) )
IF( ABS(XD(4)) .GT. ARL) THEN
  XD(4)= SIGN(ARL, XD(4)) ! Aileron rate limit
END IF
IF(X(4) .GT. ADL) THEN
  X(4)= ADL ! Aileron deflection limit
  IF(XD(4) .GT. 0.0) XD(4)= 0.0
ELSE IF (X(4) .LT. -ADL) THEN
  X(4)= -ADL
  IF(XD(4) .LT. 0.0) XD(4)= 0.0
ELSE
  CONTINUE
END IF
C
TEMP= UA
IF (ABS(TEMP) .GT. ADL) THEN
  TEMP= SIGN(ADL,TEMP) ! limit ARI to aileron limit
END IF
ARI = (0.13*X(2) - 0.7)*TEMP
RS = RD - PD*X(2)/57.3
XD(6)= RS - X(6)
ERR= RCOM + .8*XD(6) + 10.0*AY
XD(5)= 20.2*( ERR + ARI - X(5) )
C
IF( ABS(XD(5)) .GT. RRL) THEN
  XD(5)= SIGN(RRL, XD(5)) ! Rudder rate limit
END IF
IF(X(5) .GT. RDL) THEN
  X(5)= RDL
  IF(XD(5) .GT. 0.0) XD(5)= 0.0
ELSE IF (X(5) .LT. -RDL) THEN
  X(5)= -RDL
  IF(XD(5) .LT. 0.0) XD(5)= 0.0
ELSE
  CONTINUE
END IF
RETURN
END

```

Fig. 4.7-1 Code for simulating actuator limiting.

```

SUBROUTINE D(TIME,TS,X,XP,DFLAG)
PARAMETER (LL=20)
DIMENSION X(*),XP(*)
COMMON/CONTROLS/THTL,EL,AIL,RDR,QCOM,PCOM,YAWCOM
COMMON/OUTPUT/OP(LL)
LOGICAL DFLAG

C
IF (TIME .LT. 5.0) THEN
  QCOM= 0.0
  PCOM= 0.0
  YAWCOM= 0.0
ELSE IF (TIME .LT. 12.0) THEN
  QCOM= 20.0
  THTL= 1.0
ELSE IF (TIME .LT. 16.0) THEN
  QCOM= 0.0
  PCOM= 300.0
ELSE
  PCOM = 0.0
  QCOM = 0.0
END IF
RETURN
END

```

Fig. 4.7-2 Discrete-time commands for limiter simulation.

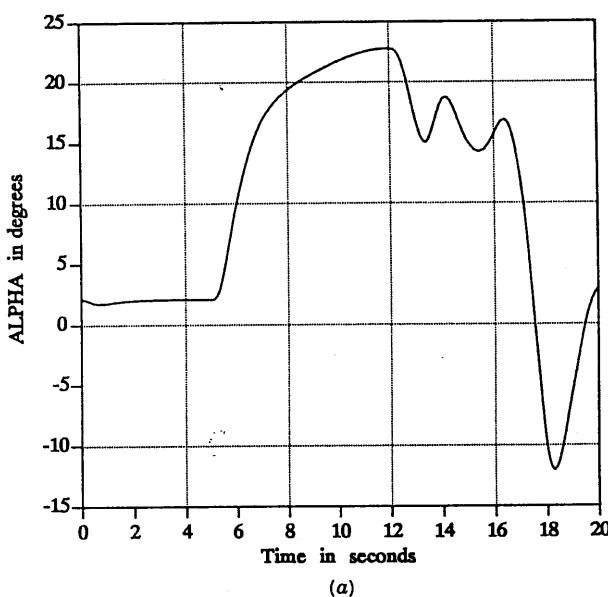
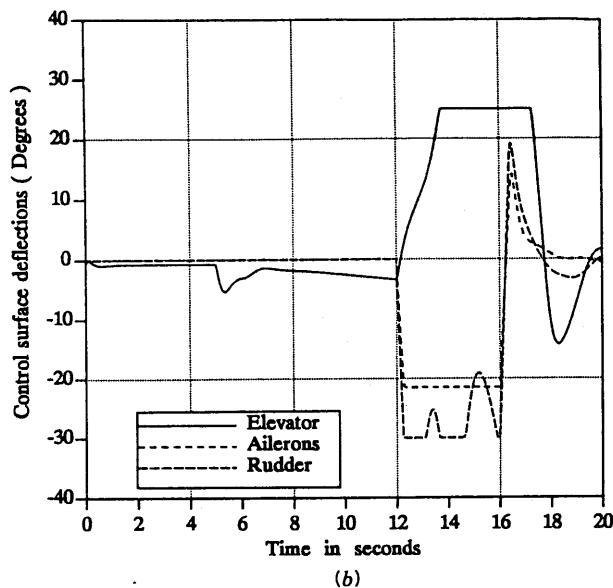
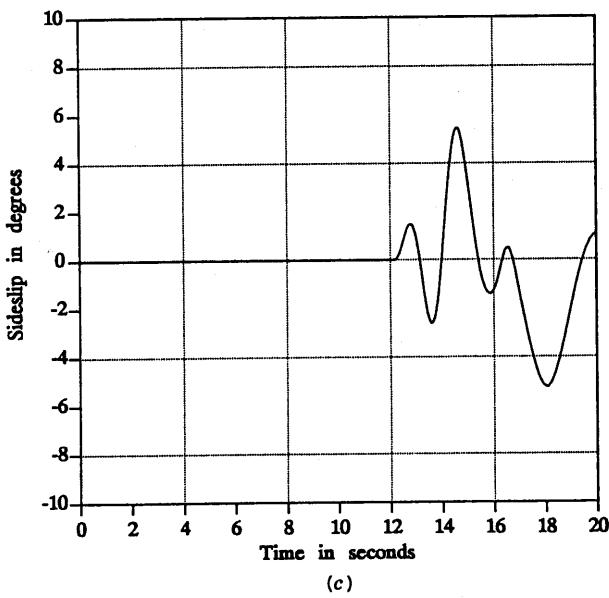


Fig. 4.7-3 (a) Alpha variation during simulation.

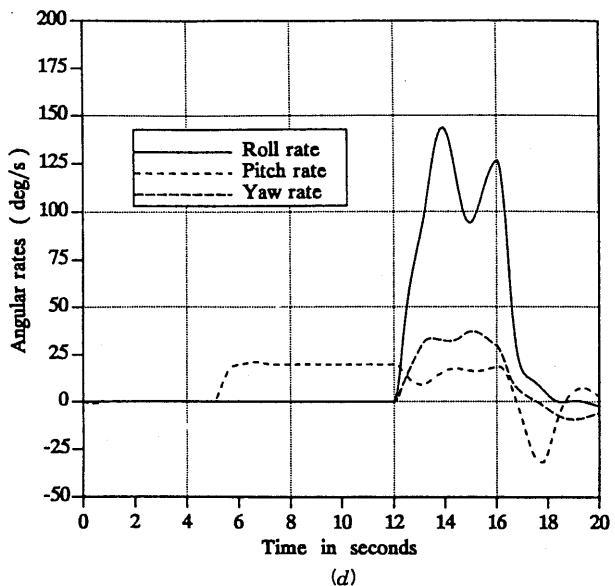


(b)

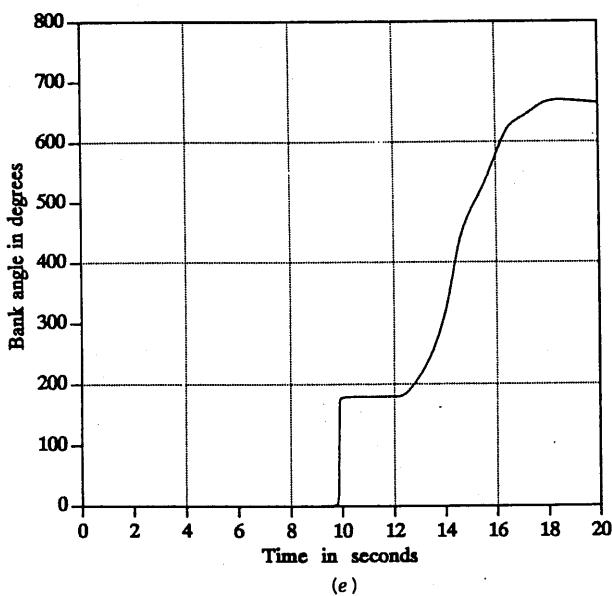


(c)

Fig. 4.7-3 (continued) (b) Control surface deflections during simulation. (c) Sideslip variation during simulation.

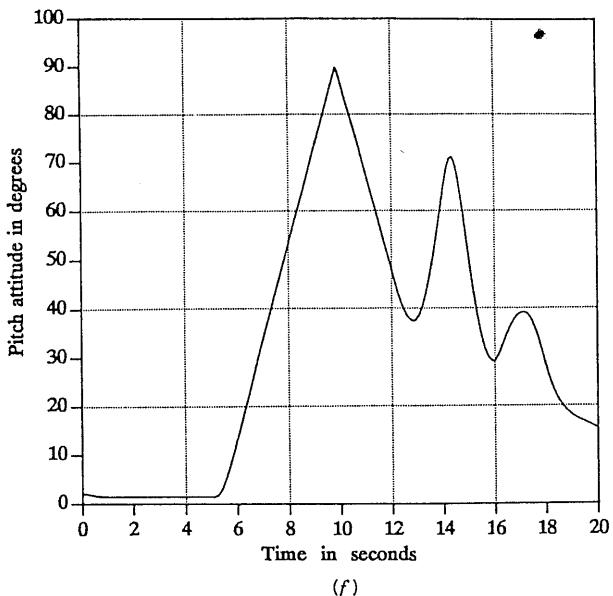


(d)

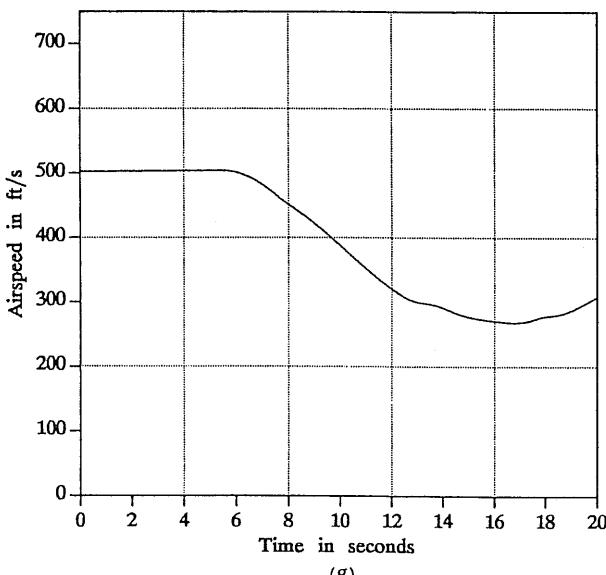


(e)

Fig. 4.7-3 (continued) (d) Aircraft angular rates during simulation. (e) Bank-angle variation during simulation.



(f)



(g)

Fig. 4.7-3 (continued) (f) Pitch-attitude variation during simulation. (g) Airspeed variation during simulation.

the bank-angle variation, Figure 4.7-3*f* shows pitch attitude, and Fig. 4.7-3*g* shows the decrease in airspeed during the maneuvers.

In the flight condition illustrated the available roll and yaw rates are insufficient to cause a pitch departure (due to inertia coupling), but the elevator saturation means that there is no longitudinal control available for 2 or 3 s. For this aircraft, pitch departure appears to be a problem only at very low dynamic pressure and high alpha, and more details can be found in Nguyen et al. [1979]. These types of problems are usually solved by using command limiters to limit the roll rate that the pilot can command or the angle of attack that can be reached through the longitudinal controls. The limiting values must be made functions of the flight conditions, and the design process is a lengthy one involving much nonlinear simulation. ■

4.8 SUMMARY

In this chapter we have described the effect of flight conditions on the aircraft modes, presented some background in flying qualities and controls design criteria, and described the purpose and design requirements of a large number of commonly used control systems. The design examples are quite realistic, having been performed on nonlinear aircraft models that are quite accurately representative of two very different types of aircraft.

For the reader who wishes to assemble the software tools, enough data have been presented to provide some check cases, and a very large number of variations and extensions of the designs can be performed. In the case of student classes, a number of competing designs can be evaluated. Time and space limitations have not allowed the control designs to be gain-scheduled over the aircraft envelope and to be evaluated thoroughly in terms of the flying qualities requirements, but the necessary capabilities have been developed.

The state-space formulation of modern control has provided an exceptionally convenient framework for the software and the use of classical design techniques. It should be evident that a primary requirement for successful design of aircraft control systems is an understanding of the physics of flight, and that interpreting the results of simulations is a vital aspect of this. Classical control theory fits extremely well into this picture because it relates very closely to the physics of the problems and usually provides clues to the modifications needed to make the design successful. In the following chapters modern design techniques will be introduced. These techniques will come into their own in situations that we found difficult to handle up to this point, such as shaping the closed-loop time response when a number of poles and zeros all contribute significantly.

REFERENCES

- AFWAL-TR-84-3105, *Proc. Aeroservoelast. Spec. Meet.*, Oct. 2-3, 1984, Vols. I and II, USAF Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio.
- Ashkenas, I. L., "Some open and closed-loop aspects of airplane lateral-directional handling qualities," *AGARD Rep. 533*, 1966.
- Ashkenas, I. L., "Twenty-five years of handling qualities research," *AIAA J. Aircraft*, vol. 21, no. 5, May 1984, pp. 289-301.
- Bischoff, D. E., "Longitudinal equivalent systems analysis of Navy tactical aircraft," AIAA paper 81-1775, *AIAA Guid. Control Conf.*, Albuquerque, N.M., Aug. 19-21, 1981.
- Blakelock, J. H., *Automatic Control of Aircraft and Missiles*. New York: Wiley, 1965.
- Bryan, G. H., *Stability in Aviation*. London: Macmillan, 1911.
- Chalk, C. R., "Fixed-base simulator investigation of the effects of L_a and true speed on pilot opinion rating of longitudinal flying qualities," *USAF ASD-TDR-63-399*, Wright-Patterson AFB, Ohio, Nov. 1963.
- Chalk, C. R., T. P. Neal, and T. M. Harris, *Background Information and User Guide for MIL-F-8785B (ASG)*, Cornell Aeronautical Laboratories, Inc., Buffalo, N.Y., Aug. 1969 (AFFDL-TR-69-72; AD860856).
- Cooper, G. E., and R. P. Harper, Jr., "The use of pilot rating in the evaluation of aircraft handling qualities," *NASA, Tech. Note D-5153*, NASA, Washington, D.C., 1969.
- Craig, S. J., and I. L. Ashkenas, Background Data and Recommended Revisions for MIL-8785B (ASG), "Military Specifications—Flying Qualities of Piloted Airplanes," *Technical Report TR-189-1*, Systems Technology Inc., Hawthorne, California, March, 1971.
- Etkin, B., *Dynamics of Atmospheric Flight*. New York: Wiley, 1972.
- Gentry, T. A., "Guidance for the use of equivalent systems with MIL-F-8785C," paper 82-1355, *AIAA Atmos. Flight Mech. Conf.*, San Diego, 1982.
- George, F. L., and D. J. Moorhouse, "Relationship of the flying qualities specification to task performance," *AFFTC/NASA-Dryden/AIAA Workshop Flight Testing to Identify Pilot Workload and Pilot Dynamics*, Edwards AFB, Calif., Jan. 19-21, 1982.
- Gerlach, O. H., "Developments in mathematical models of human pilot behavior," *Aeronaut. J.*, July 1977, pp. 293-305.
- Hodgkinson, J., "Equivalent systems approach for flying qualities specification," MCAIR 79-017, presented at *SAE Aerosp. Control Guid. Syst. Comm. Meet.*, Denver, Colo., Mar. 7-9, 1979.
- Kisslinger, R. L., and M. J. Wendle, "Survivable flight control system interim report," No. 1, "Studies analysis and approach, supplement for control criteria studies," *Tech. Rep. AFFDL-TR-71-20*, suppl. 1, Wright-Patterson AFB, Ohio, May 1971.
- Kleinman, D. L., S. Baron, and W. H. Levison, "An optimal control model of human response. Part I: Theory and validation; Part II: Prediction of human performance in a complex task," *Automatica*, vol. 6, 1970, p. 357.
- McRuer, D. T., D. Graham, E. Krendel, and W. Reisinier, Jr., "Human pilot dynamics in compensatory systems," *AFFDL-TR-65-15*, Wright-Patterson AFB, Ohio, 1965.

- McRuer, D., I. Ashkenas, and D. Graham, *Aircraft Dynamics and Automatic Control*. Princeton, N.J.: Princeton University Press, 1973.
- MIL-F-8785C, "U.S. Dept. of Defense Military Specification: Flying qualities of piloted airplanes," Nov. 5, 1980.
- MIL-F-9490D, "Flight control systems: Design, installation, and test of, piloted aircraft, general specification for," USAF, June 1975.
- Moorhouse, D. J., and R. J. Woodcock, "Background information and user guide for MIL-F-8785C," *AFWAL-TR-81-3109 (AD-A119-421)*, Wright-Patterson AFB, Ohio, July 1982.
- Neal, T. P., and R. E. Smith, "An in-flight investigation to develop control system design criteria for fighter airplanes," *AFFDL-TR-70-74*, Vols. I and II, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, Dec. 1970.
- Nguyen, L. T., et al., "Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability," *NASA Tech. Paper 1538*, NASA, Washington, D.C., Dec. 1979.
- O'Hara, F., "Handling criteria," *J. Roy. Aeronaut. Soc.*, vol. 71, no. 676, 1967, pp. 271-291.
- Onstott, E. D., and W. H. Faulkner, "Production evaluation and specification of closed-loop and multi-axis flying qualities," *AFFDL-TR-78-3*, Wright-Patterson AFB, Ohio, Feb. 1978.
- Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls*. Lawrence, Kans.: Roskam Aviation and Engineering Corp., 1979.
- Rynaski, E. G., "Flying qualities in the time domain," AIAA paper 85-1849, *AIAA Guid., Navig. Control Conf.*, Snowmass, Colo., Aug. 1985.
- Sheridan, T. B., and W. R. Ferrell, *Man-Machine Systems: Information, Control, and Decision Models of Human Operator Performance*. Cambridge, Mass.: MIT Press, 1974.
- Shomber, H. A., and W. M. Gertsen, "Longitudinal handling qualities criteria: an evaluation," *J. Aircraft*, vol. 4, no. 4, 1967, pp. 371-376.
- Tobie, H. N., E. M. Elliot, and L. G. Malcom, "A new longitudinal handling qualities criterion," *Proc. National Aerospace Electronics Conf.*, Dayton, Ohio, 16-18 May 1966, pp. 93-99.
- Toles, R. D., "Application of digital multi-input/multi-output control law with command path blended gains for the pitch axis task-tailored air combat mode of the AFITI/F-16," *Proc. 1985 Am. Control Conf.*, Boston, June 1985, p. 1475.

PROBLEMS

Section 4.2

- 4.2-1** Use the results of Section 4.2 to write a program that will calculate the damping and natural frequency of the phugoid and short period modes from the dimensionless longitudinal stability derivatives. Use it to check the results given in Table 4.2-1.

- 4.2-2** Write a program to determine the lateral directional modes from the appropriate dimensionless derivatives, and use it to check the results of Example 4.2-1. Determine both the approximate values from the equations in Section 4.2 and accurate values from the eigenvalues of the coefficient matrices.

Section 4.4

- 4.4-1** Trim the F-16 model for level flight at 30,000 ft and 820 ft/s; the dynamic pressure and angle of attack will then be the same as those of the nominal sea-level condition in Table 3.4-3. Linearize the dynamics and then attempt to find α and q feedback gains that will yield closed-loop longitudinal dynamics close to those obtained in Example 4.4-1.
- 4.4-2** Using the same flight conditions as in Problem 4.4-1, linearize the lateral-directional dynamics of the F-16 and determine the modes. Find suitable feedback gains for basic lateral-directional stability augmentation, as in Example 4.4-3. Compare the roll-rate response to an aileron doublet with the response shown in Example 4.4-3.
- 4.4-3** Repeat Example 4.4-3 with a washout time constant of 0.5 s and determine the effect on the dutch roll damping.

Section 4.5

- 4.5-1** In Example 4.5-2 the pitch-rate response is degraded when the flight conditions are different from the nominal design conditions. Repeat the pitch-CAS design for a few other values of dynamic pressure and determine if the pitch response shown in the example can be improved by scheduling one or both of the controller gains with dynamic pressure.
- 4.5-2** Repeat the design of the normal-acceleration CAS in Example 4.5-3, but with the accelerometer placed at the aircraft cg. Attempt to obtain a step response that is fast but well damped and similar to that shown in Fig. 4.5-9a.
- 4.5-3** Determine whether the alpha filter has any significant detrimental effect on the performance of the lateral-directional CAS in Example 4.5-4. Also, remove the ARI, retune the design, and compare the performance with and without the ARI.

Section 4.6

- 4.6-1** Redesign the pitch-attitude hold in Example 4.6-1, using the short-period approximation, with an additional integrator to produce pitch from pitch rate. Evaluate the design with a step response simulation performed on the full dynamics.
- 4.6-2** Redesign the pitch-attitude hold in Example 4.6-2 with the PI zero placed at $s = -0.1$; compare the step response with that given in the text. Can this design be performed using the short-period approximation?
- 4.6-3** Design a Mach-hold autopilot using the same dynamics as in Example 4.6-3. Design for a deadbeat step response with small steady-state error.
- 4.6-4** Redesign the d -loop of the glide-slope controller in Example 4.6-4 with the PI zero at $s = -0.1$. Attempt also to design the lead compensator so that the elevator is less active during the acquisition of the glide slope. Compare the simulation results with those given in the text.
- 4.6-5** Investigate the effect of changing the position of the PI zero on the performance of the implicit model-following flare control.
- 4.6-6** Modify Example 4.6-6 to use PI control and lead compensation (if necessary) so that zero steady-state bank-angle error is achieved. Use this controller to fly the bank-angle-steering simulation in Example 4.6-7, and determine whether the altitude remains more nearly constant.

Section 4.7

- 4.7-1** Include control surface rate and deflection limits (as in Example 4.7-1) in the simulation used with Problem 4.6-6, and determine the effects on performance.

CHAPTER 5

MODERN DESIGN TECHNIQUES

5.1 INTRODUCTION

Modern control theory has made a significant impact on the aircraft industry in recent years. Bryson (e.g., [1985], [Ly et al., 1985]) pioneered in applying it to aircraft control. Boeing [Gangsaas et al., 1986] has implemented control systems designed using modern techniques, for instance in the Boeing 767 autopilot. Honeywell has promoted modern robust design [Doyle and Stein, 1981]. Linear quadratic methods were used by General Dynamics in the control system of the AFTI/F-16 [AFWAL-TR-84-3008, 1984].

Therefore, in aircraft control systems design it is essential to have an understanding of modern control theory. Unfortunately, the traditional modern design techniques based on state-variable feedback that are available in current texts are not suitable for aircraft controls design. This is due to several things, one of which is their dependence on selecting large numbers of design parameters—namely, the performance index weighting matrices. Any design method for aircraft controls should eliminate the need for this trial-and-error selection. Thus the techniques in Ly et al. [1985], Gangsaas et al. [1986], Davison and Ferguson [1981], and Moerder and Calise [1985] all rely on modified design techniques that use output feedback or order-reduction techniques in conjunction with the minimization of a nonstandard performance index.

In the remainder of the book we focus on modern design techniques that are suitable for use in aircraft control. Included are such approaches as eigenstructure assignment, model following, LQG/LTR, and LQ output-feedback design. Each of these techniques has its proponents, and each has

its advantages and disadvantages, as we shall attempt to demonstrate. We shall focus on output-feedback design, with performance criteria that are more general than the usual integral-quadratic form. Using this approach it is straightforward to design controllers that have a sensible structure from the point of view of the experience within the aircraft industry, without the trial-and-error selection of a large number of design parameters.

Our strategy in the next chapters will be different than in the first part of the book due to the different character of the material to be covered. We shall first develop each modern design technique and then present examples showing how it is used in aircraft controls. In several instances we shall consider the same examples presented in Chapter 4; this will afford an opportunity to contrast the classical and modern approaches to design. We now discuss some basic philosophy of modern controls design.

Limitations of Classical Control

In Chapter 4 we showed how to design aircraft control systems using classical control techniques. The essence of classical design was *successive loop closure* guided by a good deal of intuition and experience that assisted in selecting the control system structure. For instance, we knew it was desirable to provide inner rate-feedback loops around a plant to reduce the effect of plant parameter variations. In conjunction with this we used standard compensator structures designed to approximate derivative action to stabilize the system, or integral action to eliminate steady-state error.

The one-loop-at-a-time design approach was aided by such tools as root locus, Bode and Nyquist plots, and so on, that enabled us to visualize how the system dynamics were being modified. However, the design procedure became increasingly difficult as more loops were added and did not guarantee success when the dynamics were multivariable: that is, when there were multiple inputs, multiple outputs, or multiple feedback loops.

Philosophy of Modern Control

Two concepts are central to modern control system design. The first is that *the design is based directly on the state-variable model*, which contains more information about the system than the input-output (black box) description. The state-variable model was introduced into system theory, along with matrix algebra, by R. Kalman [1958, 1960]. Since we have already seen how to extract state equations from the nonlinear aircraft dynamics and used them for analysis, we are at this point in a good position to use them for controls design.

The second central concept is *the expression of performance specifications in terms of a mathematically precise performance criterion* which then yields *matrix equations for the control gains*. These matrix equations are solved using readily available computer software. The classical successive-loop-closure

approach means that the control gains are selected individually. In complete contrast, solving matrix equations allows *all the control gains to be computed simultaneously* so that all loops are closed at the same time. This simultaneous design means that we will have greater insight into the design freedom than is possible when the system has more than one input and/or output, or multiple control loops. Moreover, using modern control theory we are able to design control systems more quickly and directly than when using classical techniques.

As in classical control, we are able with modern techniques to select the structure of the control system using the intuition developed in the aircraft industry. Thus it is straightforward to include washout circuits, integral control, and so on. The key to this is the use of *output feedback* design techniques, introduced in Section 5.3 and used throughout the chapter.

The modern control formulation means that the trial and error of one-loop-at-a-time design disappears. Instead, the fundamental engineering decision is *the selection of a suitable performance criterion*. Let us now discuss some important design problems and their associated performance criteria.

Fundamental Design Problems

Pole-Placement / Eigenvector Assignment. Modern control techniques are available for assigning the poles in multi-input/multi-output (MIMO) systems to desired locations *in one step* by solving equations for the feedback gains. These are called *pole-placement techniques*. Once we move away from classical one-loop-at-a-time design and obtain the capability to compute all the feedback gains simultaneously, it will become clear that in the MIMO case *it is possible to do more than simply assign the poles*. In fact, the closed-loop eigenvectors may also be selected within limits.

Desirable pole locations for aircraft design may be found in the military flying qualities specifications (see Section 4.3). However, while discussing flying-qualities requirements, we noted that the time response depends not only on the pole locations—it also depends on the zeros of the individual single-input/single-output (SISO) transfer functions, or equivalently on the eigenvectors (see the discussion on system modes in Section 3.7). Thus the capability of modern controls design to select both the closed-loop poles and eigenvectors is relevant in aircraft design.

In this design approach, the performance criterion is to achieve specified pole locations and eigenvectors. We shall discuss pole-placement/eigenstructure assignment in Section 5.2.

Regulator Problem. A fundamental design problem is the *regulator problem*, where it is necessary to regulate the outputs of the system to zero while ensuring that they exhibit desirable time-response characteristics. The regulator problem is important in the design of stability augmentation systems and autopilots.

Stable regulation of systems implies closed-loop stability, but using modern control we may do more than simply ensure stability. To exercise our design freedom, we select as our performance criterion an *integral-squared performance index (PI)* similar to those used in classical design [D'Azzo and Houpis, 1988]. That is, the squares of the states and inputs are integrated to obtain the PI. The control gains that minimize the PI are found by solving matrix equations using computer programs. Note that if the integral of the squares of the states is made small, then in some sense the states themselves are forced to stay near zero. Selecting different weighting coefficients in the PI for the various state components results in different time-domain behavior in the closed-loop system. Thus modern control regulator design is fundamentally a *time-domain design technique* useful in shaping the closed-loop response. This is in contrast to classical controls, where most techniques are in the frequency domain.

We shall discuss the regulator problem in Section 5.3. A deficiency of the traditional approach to modern regulator design using state feedback and the standard quadratic PI is that, to obtain suitable responses, one must select a large number of design parameters—namely, the PI weighting matrices. To avoid such trial-and-error approaches, we use modified PIs (Section 5.5) that are more suitable for aircraft control system design.

Tracker Problem. Another fundamental design problem is the *tracker problem*, where it is desired for an aircraft to follow or track a command signal. The command may either be constant or time varying. This is also referred to as the *servodesign problem*. The tracker problem is important in the design of control augmentation systems, where, for instance, the command signal may be a desired pitch rate or normal acceleration command. The tracker problem also relies on the selection of an integral-squared PI. However, now it is desired to keep the outputs not at zero, but near the reference command signals.

The modern control technique we shall use for tracker design is not the standard one—it has been modified in several respects to make it more suitable for the purpose of aircraft controls design. Specifically, we select a general PI that can easily be modified to attain different performance objectives. The result is a convenient technique for aircraft controls design that does not involve the trial-and-error tuning of large numbers of parameters. This design approach is described in Sections 5.4 and 5.5.

Model Following. An important approach to controls design is *model following*, where it is desired for the aircraft to perform like an ideal model with desirable flying qualities. We have seen in Chapter 4 that one way to specify good flying qualities is to prescribe a low-order model (e.g., with one zero and a complex pole pair) whose response the closed-loop system should match. In this design technique the performance criterion is some measure of the difference between the model and controlled aircraft responses. We

cover model-following design in Section 5.6, showing how to design controllers that make the aircraft behave like the model.

As another application, we show that model-following design offers a very straightforward approach to the design of an automatic flare control system. Note that in flare control, it is desired for the aircraft to follow an exponential path to a smooth touchdown—here, the model is just the desired trajectory.

Robust Design. It is important to incorporate notions from classical control theory into modern design. Particularly vital is the frequency-domain approach to robustness analysis. However, it is well known that in a multivariable system, individual gain and phase margins between different pairs of inputs and outputs mean little from the point of view of overall robustness. Therefore, in Chapter 6 we generalize frequency-domain robustness analysis techniques to MIMO systems using the concept of the *singular value*. There, we also present the linear quadratic regulator/loop-transfer recovery (LQG/LTR) technique, which has gained recent popularity in aircraft controls design.

Observers, Kalman Filter, and Regulators. In Chapter 6 also we cover the design of observers and the Kalman filter, which are dynamical systems that estimate the full state from measurements of the outputs. By using feedback of these state estimates in conjunction with the observer dynamics, we are able to design a dynamic linear quadratic regulator, which is just a compensator similar to those obtained using classical techniques. In the modern approach, however, a convenient design method for *multivariable systems* is achieved by solving matrix equations to guarantee specified performance.

Digital Control. The control systems of modern aircraft are implemented on digital microprocessors. Examples are the F-15E, F18, and late models of the F-16. The advantages of digital control include the ability to implement complicated multiloop control systems, reprogram the controller gains and structure (e.g., for gain scheduling), obtain redundancy for failure tolerance, and use digital signal processing techniques to filter the control signals.

Digital controls design introduces some new problems, such as the need to account for the delays associated with the sampling and hold processes and control computation. Also important is the development of design techniques that overcome the drawbacks associated with z -plane design, where the need arises for extreme accuracy in placing the poles within the unit circle. Implementation problems include accounting for actuator saturation and the effects of computer finite word length, roundoff error, overflow, and so on. We discuss the basics of digital control in Chapter 7.

5.2 ASSIGNMENT OF CLOSED-LOOP DYNAMICS

Classical design techniques such as root locus and Bode analysis are directly applicable only for SISO systems. Using such techniques a single feedback gain may be selected to place the closed-loop poles to guarantee desirable time responses and robustness qualities. In the case of multiple inputs and outputs, or multiple control loops, the classical techniques require successive closures of individual loops and involve a significant amount of trial and error.

In the MIMO case it is possible to do more than simply place the poles. This extra freedom is difficult to appreciate from the point of view of classical control theory due to the successive SISO design approach. In this section we want to show how modern control theory can be used in the multivariable case to place the poles as well as to *take advantage of the extra freedom arising from multiple inputs* to assign the closed-loop eigenvectors. This is important since as we saw in Sections 3.7 and 4.3, the time response of a multivariable system depends not only on the poles but also on the zeros of the individual SISO transfer functions, or equivalently on the eigenvectors.

The *eigenstructure assignment* technique discussed in this section offers good insight and is especially useful for the design of *decoupling controllers*, as we shall show in an example. As far as obtaining suitable time responses for multivariable systems goes, linear quadratic approaches like those in Sections 5.3 to 5.7 and Section 6.5 have generally been found more appropriate in the aircraft industry.

We shall now discuss some basic feedback concepts from the point of view of modern control theory.

State Feedback and Output Feedback

We have shown that the linearized equations of motion of an aircraft may be written in the state-space form

$$\dot{x} = Ax + Bu \quad (5.2-1)$$

$$y = Cx, \quad (5.2-2)$$

with $x(t) \in \mathbf{R}^n$ the state, $u(t) \in \mathbf{R}^m$ the control input, and $y(t) \in \mathbf{R}^p$ the measured output.

Let us select a feedback control input of the form

$$u = -Kx + v, \quad (5.2-3)$$

where $v(t)$ is an auxiliary input which might be, for instance, the pilot's command and K is an $m \times n$ gain matrix to be determined. This is called a *state-variable feedback* since all of the state components are fed back. The feedback gain K is an $m \times n$ matrix of scalar control gains.

Substituting the control into the state equation yields the closed-loop system

$$\dot{x} = (A - BK)x + Bu. \quad (5.2-4)$$

The closed-loop plant matrix is $(A - BK)$, and we would like to select the feedback gain K for good closed-loop performance.

It is a fundamental result of modern control theory that if the system is *controllable*, all of the closed-loop poles may be assigned to desired locations by selection of K . Controllability means that the control input $u(t)$ independently affects all the system modes. It can be tested for by examining the controllability matrix [Kailath, 1980]

$$U = [B \ AB \ A^2B \ \cdots \ A^{n-1}B]. \quad (5.2-5)$$

The system is controllable if U has full rank of n , that is, if U has n linearly independent columns. This is equivalent to the nonsingularity of the gramian UU^T , which is a square $n \times n$ matrix whose determinant can be evaluated.

In the next subsection we shall see that if there is more than one control loop, corresponding to more than one control gain, we cannot only place the poles, but also to a certain extent select the eigenvectors.

Unfortunately, in aircraft control systems it is usually not possible or economically feasible to measure all the states accurately. It is possible to design a dynamic observer or Kalman filter to provide estimates $\hat{x}(t)$ of the states $x(t)$, and then use *feedback of the estimates* by modifying (5.2-3) to read $u = -K\hat{x} + v$. Indeed, we do discuss this approach in Section 6.4, since we need it to cover the LQG/LTR robust design technique in Section 6.5. However, since the aircraft dynamics are nonlinear, all the parameters of any linear observer would need to be gain scheduled. This is inconvenient if the order of the observer is large.

Therefore, to obtain realistic aircraft control schemes, we should feed back not the entire state $x(t)$, but only the measurable outputs $y(t)$. The *output feedback* control law is

$$u = -Ky + v = -KCx + v \quad (5.2-6)$$

which on substitution into (5.2-1) yields the closed-loop system

$$\dot{x} = (A - BKC)x + Bu. \quad (5.2-7)$$

Now the closed-loop plant matrix is $(A - BKC)$. The output feedback matrix K is an $m \times p$ matrix of scalar gains. Thus since p is generally less than n , there are fewer scalar control gains to select in output-feedback design than in state-feedback design.

An important advantage of output feedback, as we shall see, is that it allows us to *incorporate a compensator of desired form into the feedback*

system. In aircraft control, there is a wealth of experience that often dictates the form of the control system. For instance, a washout filter may be required, or a PI controller may be needed for zero steady-state error.

Unlike the state feedback case, there is no convenient test to determine for a given system if the closed-loop poles may be independently assigned using output feedback. Pole placement using output feedback is more difficult to accomplish than using state feedback. The basic thrust of this chapter is to investigate the selection of the output feedback gain matrix K to obtain desirable closed-loop characteristics. Note that this will involve more than simply placing the poles, since desirable time responses depend on the poles as well as the zeros of the individual SISO transfer functions. These zeros can also be influenced using feedback if there is more than one input and output.

The gain matrix K is $m \times n$ for state feedback and $m \times p$ for output feedback. In the MIMO case there could be many gain elements, each corresponding to a feedback path. In classical control theory the individual gains must be separately selected using trial-and-error successive loop closure design. By contrast, using modern controls design *all the elements of K are selected simultaneously*. Thus all the feedback loops in a complicated control system can be closed at the same time with a modern control approach.

We shall now discuss the selection of K to yield desired closed-loop poles and eigenvectors—that is, to assign the closed-loop eigenstructure. Both state feedback and output feedback will be considered. First, let us recall the importance of the eigenvectors in the system response.

Modal Decomposition

We have discussed the importance of the system modes in Sections 3.7 and 4.2. Let us now carry that discussion a bit further. Let λ_i be an eigenvalue with (right) eigenvector v_i and left eigenvector w_i , so that

$$Av_i = \lambda_i v_i, \quad w_i^T A = \lambda_i w_i^T. \quad (5.2-8)$$

Since $y = Cx$, we may use the results of Section 3.7 to write the output as

$$y(t) = \sum_{i=1}^n (w_i^T x_0) C v_i e^{\lambda_i t} + \sum_{i=1}^n C v_i \int_0^t e^{\lambda_i(t-\tau)} w_i^T B u(\tau) d\tau. \quad (5.2-9)$$

The initial condition is $x(0) = x_0$. This equation is valid when the Jordan form of matrix A is diagonal.

From this equation we may note that Cv_i is a direction in the output space associated with λ_i , while the influence of the control input $u(t)$ on eigenvalue λ_i is determined by $w_i^T B$.

If $Cv_i = 0$, motion in the direction v_i cannot be observed in the output and we say that λ_i is *unobservable*. If $w_i^T B = 0$, the control input $u(t)$ can

never contribute to the motion in the direction v_i and we say that λ_i is *uncontrollable*.

Clearly, we may affect the coupling between the inputs, states, and outputs by selecting the vectors v_i and w_i in the closed-loop system; that is, we can influence the *numerators* of the individual SISO transfer functions as well as the poles. To see this clearly, examine the transfer function

$$H(s) = C(sI - A)^{-1}B. \quad (5.2-10)$$

Let the Jordan matrix J and matrix of eigenvectors M be

$$J = \text{diag}\{\lambda_i\}, \quad M = [v_1 \ v_2 \ \cdots \ v_n]^T, \quad (5.2-11)$$

so that

$$M^{-1} = [w_1 \ w_2 \ \cdots \ w_n]^T. \quad (5.2-12)$$

Now recall that $A = MJM^{-1}$ and use the fact that $(QP)^{-1} = P^{-1}Q^{-1}$ for any two compatible square matrices P and Q , to write (5.2-10) as

$$H(s) = CM(sI - J)^{-1}M^{-1}B,$$

or, since $(sI - J)^{-1}$ is diagonal with elements like $1/(s - \lambda_i)$, as

$$H(s) = \sum_{i=1}^n \frac{Cv_i w_i^T B}{s - \lambda_i}. \quad (5.2-13)$$

This equation gives the partial fraction expansion of $H(s)$ in terms of the eigenstructure of A when A is diagonalizable.

Several things may be said at this point. First, if λ_i is unobservable or uncontrollable, its contribution to the partial fraction expansion of $H(s)$ is zero. In this case we say that the state-space description (5.2-1), (5.2-2) is *not minimal*. Second, the terms Cv_i and $w_i^T B$ determine the residues of the poles, and hence the *zeros* of the individual SISO transfer functions in the $p \times m$ matrix $H(s)$. By proposing a technique for selecting the closed-loop eigenvectors by feedback, we are therefore proposing a method of shaping the time response that goes beyond what is possible using only pole placement.

In the next subsections we shall show how to design feedback gains that achieve desired closed-loop eigenvectors using both state feedback and output feedback. Meanwhile, in the following example we give some insight on desirable eigenvectors from the point of view of aircraft behavior, recalling some results from Sections 3.7 and 4.2. We also show that the eigenvectors may be selected to obtain *decoupling between the system modes*.

Example 5.2-1: Selecting Eigenvectors for Decoupling. In Sections 3.7 and 4.2 we studied the aircraft longitudinal and lateral modes, showing which

states are involved in each one. To ensure that the controlled aircraft exhibits suitable flying qualities, we should take care to design the control system so that this basic modal structure is preserved [Sobel and Shapiro, 1985; Andry et al., 1983].

In this example we idealize the findings of Sections 3.7 and 4.2 a bit. That is, we make more categorical statements about the mode couplings in order to obtain concrete design objectives.

a. *Longitudinal Axis.* Assuming that the state equations are augmented by a simple lag elevator actuator model, in the linearized perturbed longitudinal equations of an aircraft the state can be taken as

$$x = \begin{bmatrix} \alpha \\ q \\ \theta \\ v_T \\ \delta_e \end{bmatrix}. \quad (1)$$

The states are angle of attack α , pitch rate q , pitch angle θ , total velocity v_T , and elevator actuator state δ_e , the elevator deflection. We have ordered the states this way to make the upcoming discussion clearer [see (2)].

The short-period mode is due primarily to a coupling of energy between α and q . The phugoid mode is due primarily to a coupling between θ and v_T . It is desirable for the forward velocity to be unaffected by short-period oscillations, while the phugoid oscillations should have no influence on angle of attack.

To achieve this behavior, we could select the closed-loop eigenvectors as

$$v_1 = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \\ x \end{bmatrix} + j \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \\ x \end{bmatrix}, \quad v_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ x \end{bmatrix} + j \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \\ x \end{bmatrix} \quad (2)$$

with v_2 and v_4 the complex conjugates of v_1 and v_3 , respectively. Components whose values we do not care about are denoted by x . Then one oscillatory mode, the one with directions specified by v_1 and v_2 , will involve the first two components of the state vector but will not inject energy into components 3 and 4. This will be a "good" short-period mode. Similarly, the phugoid mode, described by the eigenvectors v_3 and v_4 , will involve components 3 and 4, but will not affect components 1 and 2.

A good design will have these closed-loop eigenvectors. The closed-loop poles should also be specified to attain the desired frequency and damping of the short-period and phugoid modes. The former will be defined by λ_1 and λ_2 , while the latter will be determined by λ_3 and λ_4 .

It may be necessary to modify the A matrix to a form that involves the nondimensional time to make the modal coupling of the eigenvectors more apparent [McRuer et al., 1973]. That is, the modified A matrix should have the eigenvectors in (2).

b. Lateral Axis. In the linearized perturbed lateral equations of an aircraft the state can be taken as

$$x = \begin{bmatrix} r \\ \beta \\ p \\ \phi \\ \delta_r \\ \delta_a \end{bmatrix}, \quad (3)$$

where we have assumed first-order lags for the rudder and aileron actuators. The states are yaw rate r , sideslip angle β , roll rate p , bank angle ϕ , and the actuator states δ_r , the rudder deflection, and δ_a , the aileron deflection.

Roll commands should not excite the dutch roll mode. Thus let us associate r and β in the closed-loop system with the dutch roll mode (see Section 4.2). Then the roll subsidence mode, which involves p , should not influence r and β . Similarly, the dutch roll oscillation should have no effect on roll rate or bank angle. Desirable eigenvectors to achieve this decoupling between modes are given by

$$v_1 = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \\ x \\ x \end{bmatrix} + j \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \\ x \\ x \end{bmatrix}, \quad v_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ x \\ x \\ x \end{bmatrix} \quad (4)$$

with v_2 the complex conjugate of v_1 and x denoting entries whose values we are not concerned about. The closed-loop poles should also be selected for desirable time response: λ_1 and λ_2 for the dutch roll and λ_3 for the roll subsidence mode. The desired closed-loop spiral mode may be selected as λ_4 . ■

Eigenstructure Assignment by Full State Feedback

Now that we have seen what the eigenvectors mean from the point of view of the aircraft behavior, we shall discuss the assignment of both the closed-loop poles and eigenvectors, first by full state feedback and then in the next subsection by output feedback. This represents an extension of classical control theory in several ways. First, we are able to deal in a natural fashion with MIMO systems, *selecting all the control gains simultaneously for suitable*

performance. Second, we shall be able to use the extra freedom in systems with more than one input and output to assign the eigenvectors as well as the poles, thus directly influencing the *zeros* of the individual SISO transfer functions. Third, we shall be able to address the problem of *decoupling of the modes* through considerations like those in Example 5.2-1.

Matrix Equation for Eigenstructure Assignment. For ease of presentation we shall assume that B and C have full rank m and p , respectively. Our discussion will be based on the polynomial matrix

$$C(s) = [sI - A \quad B] \quad (5.2-14)$$

with s a complex variable and A and B the system plant and input matrices.

In this subsection we follow [Moore, 1976] and consider full state feedback of the form

$$u = -Kx. \quad (5.2-15)$$

Under the influence of this control input the closed-loop system becomes

$$\dot{x} = (A - BK)x. \quad (5.2-16)$$

To select K so that a desired eigenvalue λ_i and associated eigenvector v_i are assigned to the closed-loop system, suppose that we can find a vector $u_i \in \mathbf{R}^m$ to satisfy the equation

$$[\lambda_i I - A \quad B] \begin{bmatrix} v_i \\ u_i \end{bmatrix} = 0. \quad (5.2-17)$$

Now, choose the feedback gain K to satisfy

$$Kv_i = u_i. \quad (5.2-18)$$

Using the last two equations, we may write

$$0 = (\lambda_i I - A)v_i + Bu_i \quad (5.2-19)$$

$$0 = [\lambda_i I - (A - BK)]v_i, \quad (5.2-20)$$

so that according to (5.2-8), v_i is assigned as a closed-loop eigenvector for eigenvalue λ_i .

As Cv_i was shown in the preceding subsection to be a direction in the output space \mathbf{R}^p associated with v_i , so u_i is the associated direction in the input space \mathbf{R}^m . That is, motions of $u(t)$ in the direction of u_i will cause motions of $x(t)$ in the direction of v_i , resulting in motions of $y(t)$ in the direction of Cv_i .

To complete the picture, suppose that n eigenvalues λ_i and associated eigenvectors v_i are chosen, and that in each case we have found a vector u_i that satisfies (5.2-17). Then we may define K by

$$K[v_1 \ v_2 \ \cdots \ v_n] = [u_1 \ u_2 \ \cdots \ u_n] \quad (5.2-21)$$

or by appropriate definition of the matrices V and U ,

$$KV = U. \quad (5.2-22)$$

Then, for each value of $i = 1, \dots, n$, (5.2-20) will hold, so that each λ_i will be assigned as a closed-loop pole with associated eigenvector v_i . This is the design technique for eigenstructure assignment using full state feedback. It remains only to discuss a few points.

Design Considerations. Since, by definition, the closed-loop eigenvectors must be linearly independent, it is necessary to select v_i as linearly independent vectors. Then (5.2-22) may be solved for K to give

$$K = UV^{-1} \quad (5.2-23)$$

Another issue is that the closed-loop system and feedback gain must be real and not complex. Thus if a complex closed-loop pole λ_i is selected, it is also necessary to select as a closed-loop pole its complex conjugate λ_i^* . Moreover, if v_i is to be the closed-loop eigenvector associated with a complex pole λ_i , then in order for (5.2-21) to have a real solution K , it is necessary to select v_i^* (i.e., the complex conjugate of v_i) as the eigenvector for λ_i^* .

To see that under these circumstances (5.2-22) indeed has a real solution K , note first that if u_i solves (5.2-19) for a given λ_i and v_i , then u_i^* solves the equation for their complex conjugates. Therefore, if $u_i = u_R + ju_I$ and $v_i = v_R + jv_I$, then to assign the desired eigenstructure K must satisfy

$$K[v_R + jv_I \ v_R - jv_I] = [u_R + ju_I \ u_R - ju_I]. \quad (5.2-24)$$

Postmultiplying both sides of this equation by

$$M = \frac{1}{2} \begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix},$$

it is seen that this equation is equivalent to the real equation

$$K[v_R \ v_I] = [u_R \ u_I], \quad (5.2-25)$$

which clearly has as a solution a real gain matrix K . Thus, if v_i is complex, then to obtain a real value for K it is only necessary to use not v_i and v_i^*

(respectively u_i and u_i^*) in (5.2-21) but the real and imaginary parts of v_i (respectively u_i).

Finally, we must investigate the conditions for existence of a solution to (5.2-17). It is unfortunately not usually possible to specify independently an arbitrary λ_i and v_i and obtain a solution u_i to this equation. Indeed, assuming that λ_i is not an open-loop pole, we have

$$v_i = -(\lambda_i I - A)^{-1} B u_i. \quad (5.2-26)$$

Thus, for the existence of a solution u_i , the desired v_i must be a linear combination of the m columns of the linear operator

$$L_i = (\lambda_i I - A)^{-1} B. \quad (5.2-27)$$

Since B has full rank m by assumption, the matrix L_i also has rank m . Thus v_i must lie in an m -dimensional subspace of \mathbf{R}^n that depends on the choice of λ_i . This means that we have m degrees of freedom in selecting the closed-loop eigenvector v_i once λ_i has been selected.

This last point is the crucial difference between classical SISO design and multivariable eigenstructure assignment. If $m = 1$, which corresponds to the single-input case, then eigenvector v_i has only one degree of freedom once the desired eigenvalue λ_i has been selected; that is, there is no additional freedom to choose the eigenvector. However, in the multi-input case where $m > 1$, we can have additional freedom to specify the internal structure of the closed-loop system by selecting m degrees of freedom of v_i arbitrarily. In the preceding subsection we have seen the importance of this in terms of design performance.

The successive-loop-closure approach of classical control, where only one feedback gain is selected at a time, obscures the extra design freedom arising from multiple inputs. In modern control, where all gains are selected simultaneously, this freedom is clearly revealed.

Design Procedure. The following design procedure for eigenstructure assignment is suggested. For a desired closed-loop pole/vector pair of λ_i and v_i^d , solve the equation

$$\begin{bmatrix} \lambda_i I - A & B \\ D & 0 \end{bmatrix} \begin{bmatrix} v_i \\ u_i \end{bmatrix} = \begin{bmatrix} 0 \\ v_i^d \end{bmatrix} \quad (5.2-28)$$

for u_i and the achievable eigenvector v_i . Repeat for $i = 1, \dots, n$ to select n closed-loop poles. If the v_i 's are not linearly independent, modify the choices for λ_i and/or v_i^d and repeat. Finally, determine the required state-variable feedback gain K using (5.2-23).

The design matrix D may be chosen for several different design objectives:

1. $D = I$. This is the case where the desired vectors v_i^d are eigenvectors (as in Example 5.2-1).
2. $D = C$. This is the case where the desired vectors are directions in the output space \mathbf{R}^p , so that we desire $Cv_i = v_i^d$.
3. If certain components of v_i are of no concern (see Example 5.2-1), the corresponding columns of D should be selected as zero. The remaining columns should be selected as columns from the $d \times d$ identity matrix, with d the number of rows of D . The elements of v_i that they multiply should be as specified by the elements of v_i^d . We illustrate further in Example 5.2-3.

We have seen that (5.2-28) may not have an exact solution v_i, u_i . It is necessary to find a solution so that (1) the first n equations hold exactly [i.e., (5.2-19) must hold exactly], and (2) the second block equation $Dv_i = v_i^d$ holds as closely as possible (then our design objectives are most closely matched). Subroutine LLBQF in the IMSL library (IMSL) allows us to do this. It gives a least-squares solution to the second equation in the sense that $\|Dv_i - v_i^d\|^2$ is minimized over all possible v_i for which there exists a u_i that satisfies (5.2-19) (where $\|w\|$ is the Euclidean norm of vector w).

An interactive design technique is suggested wherein:

1. Given the desired λ_i and v_i^d , (5.2-28) is solved for the pair v_i, u_i meeting the requirements above.
2. The achievable eigenvector v_i is compared with the desired eigenvector and if it is unsatisfactory either v_i^d or λ_i may be modified and step 1 repeated.

Eigenstructure Assignment by Output Feedback

In an aircraft control system, all of the states are not generally available for measurement. Instead, only selected outputs are available for control purposes. It is not difficult to modify the eigenstructure assignment technique so that the admissible controls are of the form

$$u = -Ky \quad (5.2-29)$$

with output $y(t) \in \mathbf{R}^p$ given by (5.2-2) [Srinathkumar, 1978]. In this case, we shall show that p eigenvalues may easily be assigned, with m degrees of freedom in the choice of the associated eigenvectors.

Matrix Equation for Eigenstructure Assignment. In the case of output feedback the closed-loop system is

$$\dot{x} = (A - BKC)x \quad (5.2-30)$$

and it is only necessary to replace (5.2-18) by

$$KCv_i = u_i. \quad (5.2-31)$$

Then, according to (5.2-17),

$$0 = (\lambda_i I - (A - BKC))v_i, \quad (5.2-32)$$

so that v_i is assigned as a closed-loop eigenvector for eigenvalue λ_i . In this case, (5.2-21) for K is replaced by

$$KC[v_1 \ v_2 \ \cdots \ v_r] = [u_1 \ u_2 \ \cdots \ u_r], \quad (5.2-33)$$

where r is the number of closed-loop eigenvalues selected.

If $r = p$ and the vectors Cv_i are linearly independent, we may define

$$V = [v_1 \ v_2 \ \cdots \ v_r], \quad U = [u_1 \ u_2 \ \cdots \ u_r] \quad (5.2-34)$$

and solve for K using

$$K = U(CV)^{-1}. \quad (5.2-35)$$

Thus it is clear what is lost by using incomplete state information for feedback purposes, for we can in general no longer assign n poles arbitrarily.

Extensions. What we have demonstrated is a technique for assigning by output feedback p closed-loop poles, with m degrees of freedom in specifying the components of each associated closed-loop eigenvector (where m is the number of inputs).

If $m \leq p$, so that the number of inputs is greater than or equal to the number of outputs, the technique just presented is suitable. However, if $m > p$, we must use the technique on the "dual" system. That is, (A, B, C) is replaced by (A^T, C^T, B^T) and the design is performed to find K^T . In this case we may assign m closed-loop poles, with p degrees of freedom in assigning the associated eigenvectors.

A problem with eigenstructure assignment using output feedback is that it is not possible to tell what happens to the $n - p$ poles that are not assigned. Indeed, some of them may become unstable, even though the original plant was stable. If this occurs, or if some closed-loop poles are too lightly damped, the design should be repeated using different values for λ_i or v_i^d . Generally, it is found that if one does not ask for too much in terms of modifying the original plant behavior, that is, if most of the desired closed-loop poles are not too different from the open-loop poles, instability of the unassigned poles is not a problem (as long as they are open-loop stable).

Srinathkumar shows that it is possible to assign an almost arbitrary set of $\min(n, m + p - 1)$ eigenvalues, but we shall not go into details here. Kwon

and Youn [1987] show that it may be possible to assign $m + p$ poles in some examples.

The next concept is quite important, so we shall illustrate it by an exercise.

Exercise 5.2-2: Eigenstructure Assignment Using Dynamic Regulator. We have shown how to select constant feedback gains to assign the closed-loop eigenstructure. However, it is possible to obtain a desired modal structure by using a dynamic compensator. This exercise shows how to design a dynamic compensator for eigenstructure assignment.

Consider the plant

$$\dot{x} = Ax + Bu \quad (1)$$

$$y = Cx \quad (2)$$

with the regulator dynamics defined by

$$\dot{z} = Fz + Gy \quad (3)$$

$$u = Hz + Jy. \quad (4)$$

This corresponds to the situation in Fig. 5.2-1. Matrices F , G , H , and J are to be selected to yield a desired closed-loop eigenstructure.

Show that by defining the augmented plant, input, and output matrices

$$\begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} B & 0 \\ 0 & I \end{bmatrix}, \quad \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \quad (5)$$

and the gain matrix

$$\mathbf{K} = \begin{bmatrix} -J & -H \\ -G & -F \end{bmatrix} \quad (6)$$

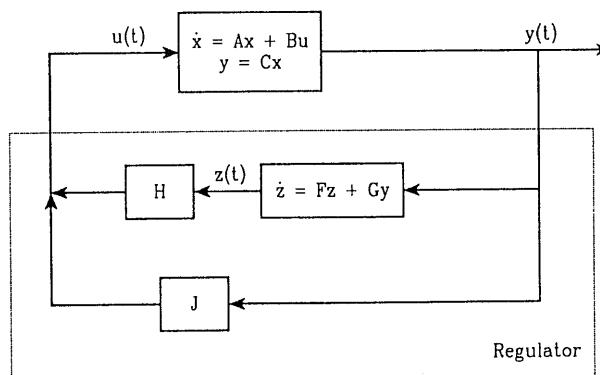


Fig. 5.2-1 Plant with regulator.

the problem of determining F , G , H , and J to yield desired closed-loop poles λ_i and eigenvectors v_i^d may be solved by using the techniques of this section to determine \mathbf{K} .

Note: A problem with this approach is that the regulator matrix F cannot be guaranteed stable. An alternative approach to regulator/observer design is given in [Andry et al., 1984]. ■

Example 5.2-3: Eigenstructure Design of Longitudinal Pitch Pointing Control.

This example is taken from [Sobel and Shapiro, 1985]. A linearized model of the short-period dynamics of an advanced (CCV-type) fighter aircraft is given. These dynamics are augmented by elevator and flaperon actuator dynamics given by the simplified model $20/(s + 20)$ so that the state vector is

$$\mathbf{x} = \begin{bmatrix} \alpha \\ q \\ \gamma \\ \delta_e \\ \delta_f \end{bmatrix}, \quad (1)$$

where the state components are, respectively, angle of attack, pitch rate, flight-path angle, elevator deflection, and flaperon deflection. The control inputs are elevator and flaperon commands so that

$$\mathbf{u} = \begin{bmatrix} \delta_{e_c} \\ \delta_{f_c} \end{bmatrix}. \quad (2)$$

The plant and control matrices are

$$A = \begin{bmatrix} -1.341 & 0.9933 & 0 & -0.1689 & -0.2518 \\ 43.223 & -0.8693 & 0 & -17.251 & -1.5766 \\ 1.341 & 0.0067 & 0 & 0.1689 & 0.2518 \\ 0 & 0 & 0 & -20 & 0 \\ 0 & 0 & 0 & 0 & -20 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20 & 0 \\ 0 & 20 \end{bmatrix}, \quad (3)$$

and the open-loop eigenvalues are

$$\left. \begin{array}{l} \lambda_1 = 5.452 \\ \lambda_2 = -7.662 \\ \lambda_3 = 0.0 \\ \lambda_4 = -20 \\ \lambda_5 = -20 \end{array} \right\} \begin{array}{l} \text{unstable short-period mode} \\ \text{pitch-attitude mode} \\ \text{elevator actuator mode} \\ \text{flaperon actuator mode.} \end{array} \quad (4)$$

The measured output available for control purposes is

$$y = \begin{bmatrix} q \\ n_{zp} \\ \gamma \\ \delta_e \\ \delta_f \end{bmatrix}, \quad (5)$$

where n_{zp} is normal acceleration at the pilot's station. The altitude rate \dot{h} is obtained from the air-data computer and the flight-path angle is then computed using

$$\gamma = \frac{\dot{h}}{TAS} \quad (6)$$

with TAS the true airspeed. The control surface deflections are measured using linear variable differential transformers (LVDT). The relation between $y(t)$ and $x(t)$ is given by

$$y = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 47.76 & -0.268 & 0 & -4.56 & 4.45 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x \equiv Cx. \quad (7)$$

Since there are five outputs and two control inputs, we may place all the closed-loop poles as well as assign the eigenvectors within two-dimensional subspaces. This roughly corresponds to selecting two components of each eigenvector arbitrarily.

The desired closed-loop short-period poles are chosen to meet military specifications for category A, level 1 flight [Mil. Spec. 1797, 1987] (see Section 4.3). Thus the desired short-period damping ratio and frequency are 0.8 and 7 rad/s, respectively.

For stability, we specify that the desired closed-loop pitch-attitude mode should decay exponentially with a time constant of 1, so that the pole should be at $s = -1$. The actuator poles should be near -20 ; however, selecting repeated poles can yield problems with the design algorithm. The desired eigenvalues are thus selected as

$$\begin{aligned} \lambda_1 &= -5.6 + j4.2 \\ \lambda_2 &= -5.6 - j4.2 \} && \text{short-period mode} \\ \lambda_3 &= -1.0 && \text{pitch-attitude mode} \\ \lambda_4 &= -19.0 && \text{elevator actuator mode} \\ \lambda_5 &= -19.5 && \text{flaperon actuator mode.} \end{aligned} \quad (8)$$

TABLE 1. Desired and Achievable Eigenvectors

<i>Desired Eigenvectors</i>		<i>Achievable Eigenvectors</i>	
$\begin{bmatrix} 1 \\ -1 \\ 0 \\ x \end{bmatrix}$ $+j \begin{bmatrix} -1 \\ 1 \\ 0 \\ x \end{bmatrix}$ $\begin{bmatrix} x \\ 1 \\ x \\ x \end{bmatrix}$ $\begin{bmatrix} x \\ x \\ 1 \\ x \end{bmatrix}$ $\begin{bmatrix} x \\ x \\ x \\ x \end{bmatrix}$	α q γ δ_e δ_f	$\begin{bmatrix} -0.93 \\ 1 \\ 0 \\ 8.36 \end{bmatrix}$ $+j \begin{bmatrix} 0 \\ -9.5 \\ 0 \\ -5.13 \end{bmatrix}$ $\begin{bmatrix} -9.5 \\ 0 \\ 1 \\ 0.129 \end{bmatrix}$ $\begin{bmatrix} -5.16 \\ 0.129 \\ -2.8 \\ -3.23 \end{bmatrix}$ $\begin{bmatrix} -0.051 \\ 1.07 \\ -0.006 \\ 1 \end{bmatrix}$	0.01 0.06 -0.014 0 1

α/q
 γ
 δ_e
 δ_f
short period

α/q
 γ
 δ_e
 δ_f
short period

In pitch pointing, the control objective is to allow pitch-attitude control while maintaining constant flight-path angle. To achieve this we select the desired closed-loop eigenvectors to decouple pitch-rate and flight-path angle. Thus an attitude command should be prevented from causing a significant flight path change. The desired closed-loop eigenvectors are shown in Table 1, where x denotes elements of no concern to us. Recall that α and q are associated with the short-period mode.

We now discuss the design procedure and the selection of the D matrix in the design equation (5.2-28). We must determine the vectors v_i and u_i for use in (5.2-33) to solve for the feedback gain matrix K . To accomplish this, first consider the desired structure of the short-period mode. According to Table 5.2-1, the required short-period eigenvectors have two "don't care" entries. Define v_1^d in terms of the required eigenvector as

$$v_1^d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \left[\begin{bmatrix} 1 \\ -1 \\ 0 \\ x \\ x \end{bmatrix} + j \begin{bmatrix} -1 \\ 1 \\ 0 \\ x \\ x \end{bmatrix} \right] = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + j \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} \quad (9)$$

to be the desired vector associated with $\lambda_1 = -5.6 + j4.2$, and select D as the 3×6 coefficient matrix in (9). Then (5.2-28) may be solved for v_1 and u_1 . Then the vectors associated with $\lambda_2 = \lambda_1^*$ are $v_2 = v_1^*$, $u_2 = u_1^*$. The achievable eigenvectors v_1 and v_2 associated with the short-period mode are shown in Table 1.

To determine whether the results to this point are satisfactory, the achievable eigenvectors v_1 and v_2 are compared with the desired eigenvectors. They are satisfactory since there is no coupling to state component 3. Note that although we attempted to select three components of the eigenvectors knowing that there are only two degrees of freedom in this selection, we have nevertheless been fortunate in attaining our design objectives. Had we not been so lucky, it would have been necessary to try different desired eigenvectors, or else slightly different values for the closed-loop poles.

Moving on to the desired structure of λ_5 , examine Table 1 to define

$$v_3^d = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ 0 \\ 1 \\ x \\ x \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (10)$$

to be the desired vector associated with $\lambda_3 = -1.0$ and select D as the 2×6 coefficient matrix in (10). Then (5.2-28) may be solved for v_3 and u_3 . The result is the achievable eigenvector v_3 shown in Table 1; again, it is suitable.

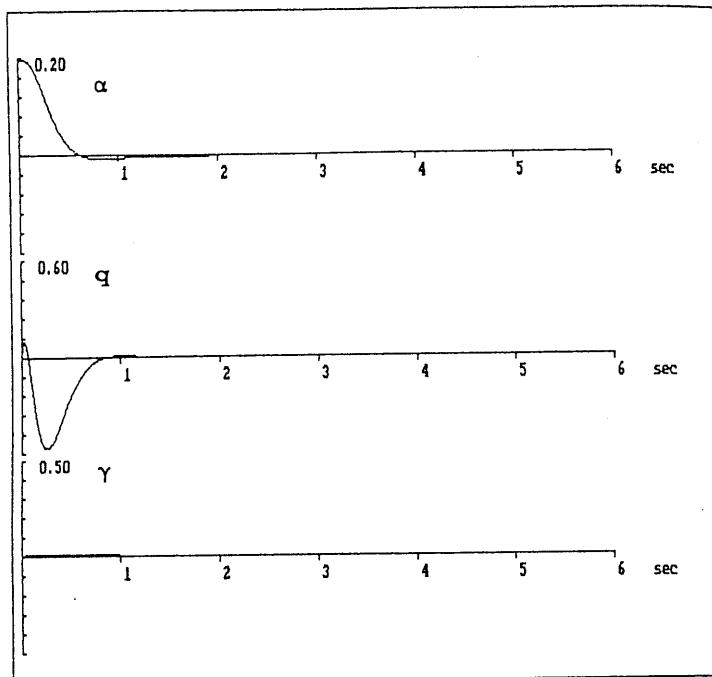


Fig. 5.2-2 Closed-loop response to angle-of-attack initial condition.

To design for the desired structure of λ_4 , examine Table 1 to define

$$v_4^d = [0 \quad 0 \quad 0 \quad 1 \quad 0] \begin{bmatrix} x \\ x \\ x \\ x \\ 1 \\ x \end{bmatrix} = 1 \quad (11)$$

to be the desired vector associated with $\lambda_4 = -19.0$ and select D as the 1×6 coefficient matrix in (11). Then (5.2-28) may be solved for v_4 and u_4 . The results are in the table. Similar procedures apply for λ_5 .

Now that all the requisite vectors v_i and u_i , $i = 1, 2, 3, 4, 5$, have been computed, they are used, along with the C matrix from (7), to solve for the feedback gain using (5.2-33). The result is

$$K = \begin{bmatrix} -0.931 & -0.149 & -3.25 & -0.153 & 0.747 \\ 0.954 & 0.210 & 6.10 & 0.537 & -1.04 \end{bmatrix}. \quad (12)$$

To check the design, a computer simulation was performed. The closed-loop system was excited with an initial condition of 0.2 rad in angle of attack. Note from Fig. 5.2-2 that this excited the short-period mode but had negligible effect on the flight-path angle. ■

5.3 LINEAR QUADRATIC REGULATOR WITH OUTPUT FEEDBACK

Our objective in this section is to show how to use modern control techniques to design stability augmentation systems (SAS) and autopilots. This is accomplished by regulating certain states of the aircraft to zero while obtaining desirable closed-loop response characteristics. It involves the problem of stabilizing the aircraft by placing the closed-loop poles at desirable locations.

Using classical control theory, we were forced to take a one-loop-at-a-time approach to designing multivariable SAS and autopilots. In this section we shall select a performance criterion that reflects our concern with closed-loop stability and good time responses, and then derive matrix equations that may be solved for *all the control gains simultaneously*. These matrix equations are solved using digital computer programs (see Appendix B). This approach thus closes all the loops simultaneously and results in a simplified design strategy for MIMO systems or SISO systems with multiple feedback loops.

Once the performance criterion has been selected, the control gains are explicitly computed by matrix design equations, and closed-loop stability will generally be guaranteed. This means that *the engineering judgment in modern control enters in the selection of the performance criterion*. Different criteria will result in different closed-loop time responses and robustness properties.

We assume the plant is given by the linear time-invariant state-variable model

$$\dot{x} = Ax + Bu \quad (5.3-1)$$

$$y = Cx, \quad (5.3-2)$$

with $x(t) \in \mathbf{R}^n$ the state, $u(t) \in \mathbf{R}^m$ the control input, and $y(t) \in \mathbf{R}^p$ the measured output. The controls will be output feedbacks of the form

$$u = -Ky, \quad (5.3-3)$$

where K is an $m \times p$ matrix of constant feedback coefficients to be determined by the design procedure. Since the regulator problem only involves stabilizing the aircraft and inducing good closed-loop time responses, $u(t)$ will be taken as a pure feedback with no auxiliary input (see Section 5.2).

As we shall see in Section 5.4, output feedback will allow us to design aircraft controllers of any desired structure. This is one reason for preferring it over full state feedback.

In the regulator problem, we are interested in obtaining good time responses as well as in the stability of the closed-loop system. Therefore, we shall select a performance criterion *in the time domain*. Let us now present this criterion.

Quadratic Performance Index

The objective of state regulation for the aircraft is to drive any initial condition error to zero, thus guaranteeing stability. This may be achieved by selecting the control input $u(t)$ to minimize a quadratic *cost* or *performance index* (*PI*) of the type

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt, \quad (5.3-4)$$

where Q and R are symmetric positive semidefinite *weighting matrices*. Positive semidefiniteness of a square matrix M (denoted $M \geq 0$) is equivalent to all its eigenvalues being nonnegative, and also to the requirement that the quadratic form $x^T M x$ be nonnegative for all vectors x . Therefore, the definiteness assumptions on Q and R guarantee that J is nonnegative and lead to a sensible minimization problem. This quadratic PI is a vector version of an integral-squared PI of the sort used in classical control [D'Azzo and Houpis, 1988].

To understand the motivation for the choice of (5.3-4), consider the following. If the square root \sqrt{M} of a positive semidefinite matrix M is defined by

$$M = \sqrt{M}^T \sqrt{M}, \quad (5.3-5)$$

we may write (5.3-4) as

$$J = \frac{1}{2} \int_0^{\infty} (\|\sqrt{Q} x\|^2 + \|\sqrt{R} u\|^2) dt, \quad (5.3-6)$$

with $\|w\|$ the Euclidean norm of a vector w (i.e., $\|w\|^2 = w^T w$). If we are able to select the control input $u(t)$ so that J takes on a minimum finite value, certainly the integrand must become zero for large time. This means that both the linear combination $\sqrt{Q} x(t)$ of the states and the linear combination $\sqrt{R} u(t)$ of the controls must go to zero. In different designs we may select Q and R for different performance requirements, corresponding to specified functions of the state and input. In particular, if Q and R are both chosen nonsingular, the entire state vector $x(t)$ and all the controls $u(t)$ will go to zero with time if J has a finite value.

Since a bounded value for J will guarantee that $\sqrt{Q} x(t)$ and $\sqrt{R} u(t)$ go to zero with time, this formulation for the PI is appropriate for the regulator problem, as any initial condition errors will be driven to zero.

If the state vector $x(t)$ consists of capacitor voltages $v(t)$ and inductor currents $i(t)$, then $\|x\|^2$ will contain terms like $v^2(t)$ and $i^2(t)$. Similarly, if velocity $s(t)$ is a state component, $\|x\|^2$ will contain terms like $s^2(t)$. Therefore, the minimization of the PI (5.3.4) is a generalized *minimum energy* problem. We are concerned with minimizing the energy in the states without using too much control energy.

The relative magnitudes of Q and R may be selected to trade off requirements on the smallness of the state against requirements on the smallness of the input. For instance, a larger control-weighting matrix R will make it necessary for $u(t)$ to be smaller to ensure that $\sqrt{R}u(t)$ is near zero. We say that a larger R *penalizes* the controls more, so that they will be smaller in norm relative to the state vector. On the other hand, to make $x(t)$ go to zero more quickly with time, we may select a larger Q .

As a final remark on the PI, we shall see that the positions of the closed-loop poles depend on the choices for the weighting matrices Q and R . That is, Q and R may be chosen to yield good time responses in the closed-loop system.

Let us now derive matrix design equations that maybe used to solve for the control gain K that minimizes the PI. The result will be the design equations in Table 5.3-1. Software to solve these equations for K is described in Appendix B.

Solution of the LQR Problem

The LQR problem with output feedback is the following. Given the linear system (5.3-1), (5.3-2), find the feedback coefficient matrix K in the control input (5.3-3) that minimizes the value of the quadratic PI (5.3-4). In contrast with most of the classical control techniques given in earlier chapters, this is a *time-domain* design technique.

By substituting the control (5.3-3) into (5.3-1) the closed-loop system equations are found to be

$$\dot{x} = (A - BKC)x \equiv A_c x. \quad (5.3-7)$$

The PI may be expressed in terms of K as

$$J = \frac{1}{2} \int_0^{\infty} x^T (Q + C^T K^T R K C) x dt. \quad (5.3-8)$$

The design problem is now to select the gain K so that J is minimized subject to the dynamical constraint (5.3-7).

This *dynamical* optimization problem may be converted into an equivalent *static* one that is easier to solve as follows. Suppose that we can find a constant, symmetric, positive-semidefinite matrix P so that

$$\frac{d}{dt}(x^T P x) = -x^T (Q + C^T K^T R K C) x. \quad (5.3-9)$$

Then J may be written as

$$J = \frac{1}{2} x^T(0) P x(0) - \frac{1}{2} \lim_{t \rightarrow \infty} x^T(t) P x(t). \quad (5.3-10)$$

Assuming that the closed-loop system is asymptotically stable so that $x(t)$ vanishes with time, this becomes

$$J = \frac{1}{2}x^T(0)Px(0). \quad (5.3-11)$$

If P satisfies (5.3-9), we may use (5.3-7) to see that

$$\begin{aligned} -x^T(Q + C^T K^T R K C)x &= \frac{d}{dt}(x^T Px) = \dot{x}^T Px + x^T P \dot{x} \\ &= x^T(A_c^T P + PA_c)x. \end{aligned} \quad (5.3-12)$$

Since this must hold for all initial conditions, and hence for all state trajectories $x(t)$, we may write

$$g \equiv A_c^T P + PA_c + C^T K^T R K C + Q = 0. \quad (5.3-13)$$

If K and Q are given and P is to be solved for, this is called a *Lyapunov equation*. (A Lyapunov equation is a symmetric linear matrix equation. Note that the equation does not change if its transpose is taken.)

In summary, for any fixed feedback matrix K if there exists a constant, symmetric, positive-semidefinite matrix P that satisfies (5.3-13), and if the closed-loop system is stable, the cost J is given in terms of P by (5.3-11). This is an important result in that the $n \times n$ auxiliary matrix P is independent of the state. Given a feedback matrix K , P may be computed from the Lyapunov equation (5.3-13). Then only the initial condition $x(0)$ is required to compute the closed-loop cost under the influence of the feedback control (5.3-3). That is, we may compute the cost of applying the feedback control $u = -Ky$ before we actually apply it.

It is now necessary to use this result to compute the gain K that minimizes the PI. By using the trace identity

$$\text{tr}(AB) = \text{tr}(BA) \quad (5.3-14)$$

for any compatibly dimensioned matrices A and B (with the trace of a matrix the sum of its diagonal elements), we may write (5.3-11) as

$$J = \frac{1}{2} \text{tr}(PX) \quad (5.3-15)$$

where the $n \times n$ symmetric matrix X is defined by

$$X \equiv x(0)x^T(0) \quad (5.3-16)$$

It is now clear that the problem of selecting K to minimize (5.3-8) subject to the dynamical constraint (5.3-7) on the states is equivalent to the *algebraic* problem of selecting K to minimize (5.3-15) subject to the constraint (5.3-13) on the auxiliary matrix P .

To solve this modified problem, we use the Lagrange multiplier approach [Lewis, 1986] to modify the problem yet again. Thus adjoin the constraint to the PI by defining the Hamiltonian

$$\mathcal{H} = \text{tr}(PX) + \text{tr}(gS) \quad (5.3-17)$$

with S a symmetric $n \times n$ matrix of Lagrange multipliers which still needs to be determined. Then our constrained optimization problem is equivalent to the simpler problem of minimizing (5.3-17) without constraints. To accomplish this we need only set the partial derivatives of \mathcal{H} with respect to all the independent variables P , S , and K equal to zero. Using the facts that for any compatibly dimensioned matrices A , B , and C and any scalar y ,

$$\frac{\partial}{\partial B} \text{tr}(ABC) = A^T C^T \quad (5.3-18)$$

and

$$\frac{\partial y}{\partial B^T} = \left[\frac{\partial y}{\partial B} \right]^T, \quad (5.3-19)$$

the necessary conditions for the solution of the LQR problem with output feedback are given by

$$0 = \frac{\partial \mathcal{H}}{\partial S} = g = A_c^T P + PA_c + C^T K^T R K C + Q \quad (5.3-20)$$

$$0 = \frac{\partial \mathcal{H}}{\partial P} = A_c S + S A_c^T + X \quad (5.3-21)$$

$$0 = \frac{1}{2} \frac{\partial \mathcal{H}}{\partial K} = R K C S C^T - B^T P S C^T. \quad (5.3-22)$$

The first two of these are Lyapunov equations and the third is an equation for the gain K . If R is positive definite (i.e., all eigenvalues greater than zero, which implies nonsingularity; denoted $R > 0$) and $C S C^T$ is nonsingular, then (5.3-22) may be solved for K to obtain

$$K = R^{-1} B^T P S C^T (C S C^T)^{-1}. \quad (5.3-23)$$

To obtain the output feedback gain K minimizing the PI (5.3-4), we need to solve the three coupled equations (5.3-20), (5.3-21), and (5.3-23). This situation is quite strange, for to find K we must determine along the way the values of two auxiliary and apparently unnecessary $n \times n$ matrices, P and S . These auxiliary quantities may, however, not be as unnecessary as it appears, for note that the optimal cost may be determined directly from P and the initial state by using (5.3-11).

The Initial Condition Problem. Unfortunately, the dependence of X in (5.3-16) on the initial state $x(0)$ is undesirable, since it makes the optimal gain dependent on the initial state through equation (5.3-21). In many applications $x(0)$ may not be known. This dependence is typical of output-feedback design. We shall see at the end of this chapter that in the case of state feedback it does not occur. Meanwhile, it is usual [Levine and Athans, 1970] to sidestep this problem by minimizing not the PI (5.3-4) but its *expected value*, that is, $E\{J\}$. Then (5.3-11) and (5.3-16) are replaced by

$$E\{J\} = \frac{1}{2}E\{x^T(0)Px(0)\} = \frac{1}{2}\text{tr}(PX), \quad (5.3-24)$$

where the symmetric $n \times n$ matrix

$$X \equiv E\{x(0)x^T(0)\} \quad (5.3-25)$$

is the initial autocorrelation of the state. It is usual to assume that nothing is known of $x(0)$ except that it is uniformly distributed on a surface described

TABLE 5.3-1. LQR with Output Feedback

System Model

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

Control

$$u = -Ky$$

Performance Index

$$J = \frac{1}{2}E\left[\int_0^\infty (x^TQx + u^TRu) dt\right]$$

with

$$Q \geq 0, \quad R > 0$$

Optimal Gain Design Equations

$$0 = A_c^T P + PA_c + C^T K^T R K C + Q \quad (5.3-26)$$

$$0 = A_c S + S A_c^T + X \quad (5.3-27)$$

$$K = R^{-1} B^T P S C^T (C S C^T)^{-1} \quad (5.3-28)$$

where

$$A_c = A - B K C, \quad X = E\{x(0)x^T(0)\}$$

Optimal Cost

$$J = \frac{1}{2}\text{tr}(PX) \quad (5.3-29)$$

by X . That is, we assume that the actual initial state is unknown, but that it is nonzero with a certain expected Euclidean norm. For instance, if the initial states are assumed to be uniformly distributed on the unit sphere, then $X = I$, the identity. This is a sensible assumption for the regulator problem, where we are trying to drive arbitrary nonzero initial states to zero.

The design equations for the LQR with output feedback are collected in Table 5.3-1 for convenient reference. We shall now discuss their solution for K .

Determining the Optimal Feedback Gain

The importance of this modern LQ approach to controls design is that the matrix equations in Table 5.3-1 are used to solve for all the $m \times p$ elements of K at once. This corresponds to *closing all the feedback loops simultaneously*. Moreover, as long as certain reasonable conditions (to be discussed) on the plant and PI weighting matrices hold, *the closed-loop system is generally guaranteed to be stable*. In view of the trial-and-error successive-loop-closure approach used in stabilizing multivariable systems using classical approaches, this is quite important.

The equations for P , S , and K are coupled nonlinear matrix equations in three unknowns. It is important to discuss some aspects of their solution for the optimal feedback gain matrix K .

Numerical Solution Techniques. There are three basic numerical techniques for determining the optimal output-feedback gain K . First, we may use a numerical optimization routine such as the simplex algorithm in [Nelder and Mead, 1964; Press et al., 1986]. This algorithm would use only (5.3-26) and (5.3-29). For a given value of K , it would solve the Lyapunov equation for P and then use P in the second equation to determine $E\{J\}$. Based on this, it would vary the elements of K to minimize $E\{J\}$. The Lyapunov equation may be solved using, for instance, subroutine ATXPXA in [Bartels and Stewart, 1972]. See also the NASA controls design package ORACLS [Armstrong, 1980].

A second approach for computing K is to use a gradient-based routine such as Davidon–Fletcher–Powell [Press et al., 1986]. This routine would use all of the design equations in Table 5.3-1. For a given value of K , it would solve the two Lyapunov equations to find the auxiliary matrices P and S . Then it would use the third design equation in the form (5.3-22). Note that if P satisfies the first Lyapunov equation, then $g = 0$ so that [see (5.3-17)] $E\{J\} = \frac{1}{2}E\{H\}$ and $\partial E\{J\}/\partial K = \frac{1}{2}\partial E\{\mathcal{H}\}/\partial K$. Thus the third design equation gives the gradient of $E\{J\}$ with respect to K , which would be used by the routine to update the value of K .

TABLE 5.3-2. Optimal Output Feedback Solution Algorithm

-
1. Initialize:
Set $k = 0$.
Determine a gain K_0 so that $A - BK_0C$ is asymptotically stable.
 2. k th iteration:
Set $A_k = A - BK_kC$.
Solve for P_k and S_k in
$$0 = A_k^T P_k + P_k A_k + C^T K_k^T R K_k C + Q$$
$$0 = A_k S_k + S_k A_k^T + X$$

Set $J_k = \frac{1}{2} \text{tr}(P_k X)$.
Evaluate the gain update direction
$$\Delta K = R^{-1} B^T P S C^T (C S C^T)^{-1} - K_k$$

Update the gain by
$$K_{k+1} = K_k + \alpha \Delta K$$

where α is chosen so that
$$A - BK_{k+1}C$$
 is asymptotically stable
$$J_{k+1} \equiv \frac{1}{2} \text{tr}(P_{k+1} X) \leq J_k$$

If J_{k+1} and J_k are close enough to each other, go to 3.
Otherwise, set $k = k + 1$ and go to 2.
 3. Terminate:
Set $K = K_{k+1}$, $J = J_{k+1}$.
Stop.
-

Finally, an iterative solution algorithm was presented in [Moerder and Calise, 1985]. It is given in Table 5.3-2. It was shown in [Moerder and Calise, 1985] that the algorithm converges to a local minimum for J if the following conditions hold.

Conditions for Convergence of the LQ Solution Algorithm:

1. There exists a gain K such that A_c is stable. If this is true, we call the system (5.3-1)/(5.3-2) *output stabilizable*.
2. The output matrix C has full row rank p .
3. Control weighting matrix R is positive definite. This means that all the control inputs should be weighted in the PI.
4. Q is positive semidefinite and (\sqrt{Q}, A) is *detectable*. That is, the observability matrix polynomial

$$O(s) \equiv \begin{bmatrix} sI - A \\ -\sqrt{Q} \end{bmatrix} \quad (5.3-30)$$

has full rank n for all values of the complex variable s not contained in the left-half plane [Kailath, 1980].

If these conditions hold, the algorithm finds an output-feedback gain that stabilizes the plant and minimizes the PI. The detectability condition means that any unstable system modes must be observable in the PI. Then if the PI is bounded, which it is if the optimization algorithm is successful, signals associated with the unstable modes must go to zero as t becomes large; that is, they are stabilized in the closed-loop system.

Initial Stabilizing Gain. Since all three algorithms for solving the matrix equations in Table 5.3-1 for K are iterative in nature, a basic issue for all of them is the selection of an initial stabilizing output-feedback gain K_0 . That is, to start the algorithms, it is necessary to provide a K_0 such that $(A - BK_0C)$ is stable. See, for instance, Table 5.3-2.

One technique for finding such a gain is given in [Broussard and Halyo, 1983]. Another possibility is to use the eigenstructure assignment techniques of the preceding section to determine an initial gain for the LQ solution algorithm. We could even select a stabilizing gain using the classical techniques of Chapter 4 and then use modern design techniques to tune the control gains for optimal performance.

A quite convenient technique for finding an initial stabilizing gain K_0 is discussed in Section 5.5. This involves finding a full $m \times n$ state-variable feedback matrix and then zeroing the entries that are not needed in the $m \times p$ output-feedback matrix for the given measured outputs. Note that there are many techniques for finding a full state feedback that stabilizes a system given A and B (see Section 5.7 and [Lewis, 1986]).

Iterative Design. Software that solves for the optimal output-feedback gain K is described in Appendix B. Given good software, design using the LQ approach is straightforward. A design procedure would involve selecting the *design parameters* Q and R , determining the optimal gain K , and simulating the closed-loop response and frequency-domain characteristics. If the results are not suitable, different matrices Q and R are chosen and the design is repeated. Good software makes a design iteration take only a few minutes.

This approach introduces the notion of *tuning the design parameters Q and R for good performance*. In the next two sections we shall present sensible techniques for obtaining suitable PI weighting matrices Q and R that do not depend on individually selecting all of their entries.

Example 5.3-1 will illustrate these notions.

Selection of the PI Weighting Matrices

Once the PI weighting matrices Q and R have been selected, the determination of the optimal feedback gain K is a formal procedure relying on the solution of nonlinear coupled matrix equations. Therefore, the engineering judgment in modern LQ design appears in the selection of Q and R . There are some guidelines for this which we shall now discuss.

Observability in the Choice of Q . For stabilizing solutions to the output-feedback problem, it is necessary for (\sqrt{Q}, A) to be detectable. The detectability condition basically means that Q should be chosen so that all unstable states are weighted in the PI. Then, if J is bounded so that $\sqrt{Q}x(t)$ vanishes for large t , the open-loop unstable states will be forced to zero through the action of the control. This means exactly that the unstable poles must have been stabilized by the feedback control gain.

A stronger condition than detectability is *observability*, which amounts to the full rank of $O(s)$ for all values of s . Observability is easier to check than detectability since it is equivalent to the full rank n of the *observability matrix*

$$O = \begin{bmatrix} \sqrt{Q} \\ \sqrt{Q}A \\ \vdots \\ \sqrt{Q}A^{n-1} \end{bmatrix}, \quad (5.3-31)$$

which is a constant matrix and so easier to deal with than $O(s)$. In fact, O has full rank n if and only if the observability gramian $O^T O$ is nonsingular. Since the gramian is an $n \times n$ matrix, its determinant is easily examined using available software [e.g., singular-value decomposition/condition number (IMSL)]. The observability of (\sqrt{Q}, A) means basically that *all* states are weighted in the PI.

From a numerical point of view, if (\sqrt{Q}, A) is observable, a positive definite solution P to (5.3-26) results; otherwise, P may be singular. Since P helps determine K through (5.3-28), it is found that if P is singular, it may result in some zero-gain elements in K . That is, if (\sqrt{Q}, A) is not observable, the LQ algorithm can refuse to close some of the feedback loops.

This observability condition amounts to a restriction on the selection of Q , and is a drawback of modern control (see Example 5.3-1). In Section 5.5 we shall show how to avoid this condition by using a modified PI.

The Structure of Q . The choice of Q can be confronted more easily by considering the performance objectives of the LQR. Suppose that a *performance output*

$$z = Hx \quad (5.3-32)$$

is required to be small in the closed-loop system. For instance, in an aircraft lateral regulator it is desired for the sideslip angle, yaw rate, roll angle, and roll rate to be small (see Example 5.3-1). Therefore, we might select $z = [\beta \ r \ \phi \ p]^T$. Once $z(t)$ has been chosen, the performance output matrix H may be formally written down.

The signal $z(t)$ may be made small by LQR design by selecting the PI

$$J = \frac{1}{2} \int_0^\infty (z^T z + u^T R u) dt, \quad (5.3-33)$$

which amounts to using the PI in Table 5.3-1 with $Q = H^T H$, so that Q may be computed from H . That is, by weighting *performance outputs* in the PI, Q is directly given.

Maximum Desired Values of $z(t)$ and $u(t)$. A convenient guideline for selecting Q and R is given in [Bryson and Ho, 1975]. Suppose that the performance output (5.3-32) has been defined so that H is given. Consider the PI

$$J = \frac{1}{2} \int_0^\infty (z^T \bar{Q} z + u^T R u) dt. \quad (5.3-34)$$

Then, in Table 5.3-1 we have $Q = H^T \bar{Q} H$. To select \bar{Q} and R , one might proceed as follows, using the *maximum allowable deviations* in $z(t)$ and $u(t)$.

Define the maximum allowable deviation in component $z_i(t)$ of $z(t)$ as z_{iM} and the maximum allowable deviation in component $u_i(t)$ of the control input $u(t)$ as u_{iM} . Then \bar{Q} and R may be selected as $\bar{Q} = \text{diag}\{q_i\}$, $R = \text{diag}\{r_i\}$, with

$$q_i = \frac{1}{z_{iM}^2}, \quad r_i = \frac{1}{r_{iM}^2}. \quad (5.3-35)$$

The rationale for this choice is easy to understand. For instance, as the allowed limits z_{iM} on $z_i(t)$ decrease, the weighting in the PI placed on $z_i(t)$ increases, which requires smaller excursions in $z_i(t)$ in the closed-loop system.

Implicit Model Following. The implicit model-following design technique in Section 5.6 shows how to select Q and R so that the closed-loop system behaves like a prescribed ideal model. The ideal model may be selected according to flying-qualities requirements (see Section 4.3). It should be selected so that its poles and zeros correspond to the desired closed-loop time-response characteristics.

Asymptotic Properties of the LQR. Consider the PI

$$J = \frac{1}{2} \int_0^\infty (x^T Q x + \rho u^T R u) dt, \quad (5.3-36)$$

where ρ is a scalar design parameter. There are some quite nice results that describe the asymptotic performance of the LQR as ρ becomes small and as

ρ becomes large [Kwakernaak and Sivan, 1972; Harvey and Stein, 1978; Grimble and Johnson, 1988].

These results detail the asymptotic closed-loop eigenstructure of the LQR and are of some assistance in selecting Q and R . Unfortunately, they are only well developed for the case of full state-variable feedback, where $C = I$ and all the states are allowed for feedback. Thus they are appropriate in connection with the discussion in Section 5.7.

Example 5.3-1: LQR Design for F-16 Lateral Regulator. In Example 4.4-3 we designed a roll damper/yaw damper for a low-speed flight condition of the F-16. Successive loop closures were used to perform the design using the root-locus approach. In this example we should like to demonstrate the power of the LQ design equations in Table 5.3-1 by designing a lateral regulator.

In our approach we shall select the design parameters Q and R in the table and then use the design equations there to close all the feedback loops simultaneously by computing K . The objective is to design a closed-loop controller to provide for the function of a lateral SAS as well as the closure of the roll-attitude loop. This objective involves the design of two feedback channels with multiple loops, but it is straightforward to deal with using modern control techniques. The simplicity of MIMO design using the LQR will be evident.

a. *Aircraft State Equations.* We used the F-16 linearized lateral dynamics at the nominal flight condition in Table 3.4-3 ($V_T = 502$ ft/s, 300 psf dynamic pressure, cg at $0.35\bar{c}$) retaining the lateral states sideslip β , bank angle ϕ , roll rate p , and yaw rate r . Additional states δ_a and δ_r are introduced by the aileron and rudder actuators

$$\delta_a = \frac{20.2}{s + 20.2} u_a, \quad \delta_r = \frac{20.2}{s + 20.2} u_r. \quad (1)$$

A washout filter

$$r_w = \frac{s}{s + 1} r \quad (2)$$

is used, with r the yaw rate and r_w the washed-out yaw rate. The washout filter state is denoted x_w . Thus the entire state vector is

$$x = [\beta \quad \phi \quad p \quad r \quad \delta_a \quad \delta_r \quad x_w]^T. \quad (3)$$

The full-state-variable model of the aircraft plus actuators, washout filter, and control dynamics is of the form

$$\dot{x} = Ax + Bu, \quad (4)$$

with

$$A = \begin{bmatrix} -0.3220 & 0.0640 & 0.0364 & -0.9917 & 0.0003 & 0.0008 & 0 \\ 0 & 0 & 1 & 0.0037 & 0 & 0 & 0 \\ -30.6492 & 0 & -3.6784 & 0.6646 & -0.7333 & 0.1315 & 0 \\ 8.5396 & 0 & -0.0254 & -0.4764 & -0.0319 & -0.0620 & 0 \\ 0 & 0 & 0 & 0 & -20.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20.2 & 0 \\ 0 & 0 & 0 & 57.2958 & 0 & 0 & -1 \end{bmatrix} \quad (5)$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20.2 & 0 \\ 0 & 20.2 \\ 0 & 0 \end{bmatrix}.$$

The control inputs are the rudder and aileron servo inputs so that

$$u = \begin{bmatrix} u_a \\ u_r \end{bmatrix} \quad (6)$$

and the output is

$$y = \begin{bmatrix} r_w \\ p \\ \beta \\ \phi \end{bmatrix}. \quad (7)$$

Thus $y = Cx$ with

$$C = \begin{bmatrix} 0 & 0 & 0 & 57.2958 & 0 & 0 & -1 \\ 0 & 0 & 57.2958 & 0 & 0 & 0 & 0 \\ 57.2958 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (8)$$

The factor of 57.2958 converts radians to degrees. The feedback control will be output feedback of the form $u = -Ky$, so that K is a 2×4 matrix. That is, we shall select eight feedback gains.

For this system the open-loop dutch roll mode has poles at $-0.4425 \pm j3.063$ and so has insufficient damping. The spiral mode has a pole at -0.01631 .

b. LQR Output Feedback Design. For the computation of the feedback gain K , it is necessary to select PI weighting matrices Q and R in Table 5.3-1. Then the software described in Appendix B is used to compute the optimal gain K using the design equations in the table. Our philosophy for selecting Q and R follows.

First, let us discuss the choice of Q . It is desired to obtain good stability of the dutch roll mode, so that β^2 and r^2 should be weighted in the PI by factors of q_{dr} . To obtain stability of the roll mode, which in closed-loop will consist primarily of p and ϕ , we may weight p^2 and ϕ^2 in the PI by factors of q_r . We do not care about δ_a and δ_r , so it is not necessary to weight them in the PI; the control weighting matrix R will prevent unreasonably large control inputs. Thus so far we have

$$Q = \text{diag}\{q_{dr}, q_r, q_r, q_{dr}, 0, 0, 0\}. \quad (9)$$

We do not care directly about x_w ; however, it is necessary to weight it in the PI. This is because omitting it would cause problems with the observability condition. A square root of Q in (9) is

$$\sqrt{Q} = [\sqrt{q_{dr}} \quad \sqrt{q_r} \quad \sqrt{q_r} \quad \sqrt{q_{dr}} \quad 0 \quad 0 \quad 0]. \quad (10)$$

Consequently, the observability matrix (5.3-31) has a right-hand column of zero; hence the system is unobservable. This may be noted in simpler fashion by examining the A matrix in (5), where the seventh state x_w is seen to have no influence on the states that are weighted in (9). To correct this potential problem, we chose

$$Q = \text{diag}\{q_{dr}, q_r, q_r, q_{dr}, 0, 0, 1\}. \quad (11)$$

As far as the R matrix goes, it is generally satisfactory to select it as

$$R = \rho I, \quad (12)$$

with I the identity matrix and ρ a scalar design parameter.

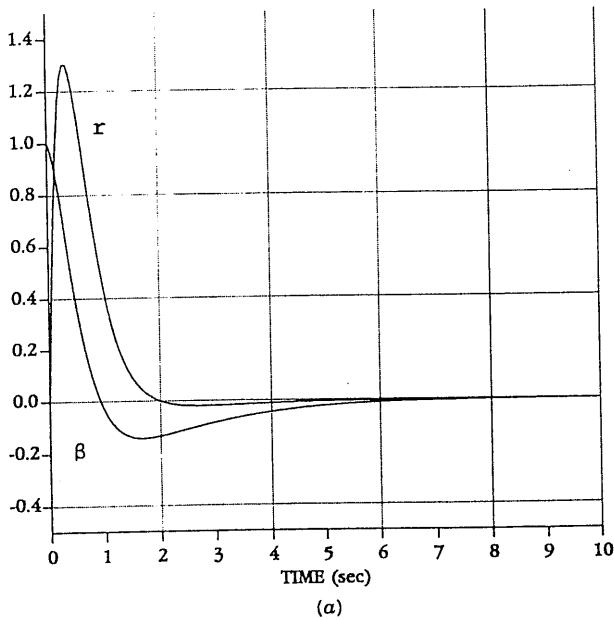
Now the design equations in Table 5.3-1 were solved using the software described in Appendix B for several choices of ρ, q_{dr}, q_r . After a few trials, we obtained a good result using $\rho = 0.1$, $q_{dr} = 50$, $q_r = 100$. For this selection the optimal feedback gain was

$$K = \begin{bmatrix} -0.56 & -0.44 & 0.11 & -0.35 \\ -1.19 & -0.21 & -0.44 & 0.26 \end{bmatrix}. \quad (13)$$

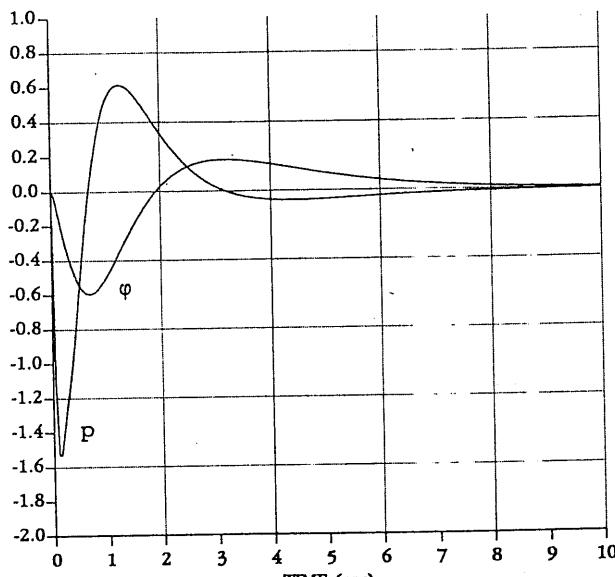
The resulting closed-loop poles were at

$$\begin{aligned} s = -3.13 &\pm j0.83 && \text{dutch roll mode } (r, \beta) \\ -0.82 &\pm j0.11 && \text{roll mode } (p, \phi) \\ -11.47 &\pm j17.18, -15.02. && \end{aligned} \quad (14)$$

To verify the design a simulation was performed. The initial state was selected as $x(0) = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$; that is, we chose $\beta(0) = 1$. Figure 5.3-1 shows the results. Part (a) shows the dutch roll mode and part (b)



(a)



(b)

Fig. 5.3-1 Closed-loop lateral response: (a) dutch roll states β and r ; (b) roll mode states ϕ and p .

the roll mode. Note that the responses correspond to the poles in (14), where the dutch roll is the faster mode. Compare to the results of Example 4.4-3.

This design has two deficiencies. First, it uses eight feedback gains in (13). This is undesirable for two reasons: (1) it requires the gain scheduling of all eight gains, and (2) the control system has no structure. That is, all outputs are fed back to both inputs; zeroing some of the gains would give the controller more structure in terms of feeding back certain outputs to only one or the other of the inputs.

The second deficiency is that it was necessary to juggle the entries of Q to obtain a good solution. Actually, due to our weighting of β^2 and r^2 by q_{dr} , and ϕ^2 and p^2 by q_r , the design was fairly straightforward and took about half an hour in all. It was, however, necessary to weight the washout filter state x_w , which is not obvious without considering the observability question.

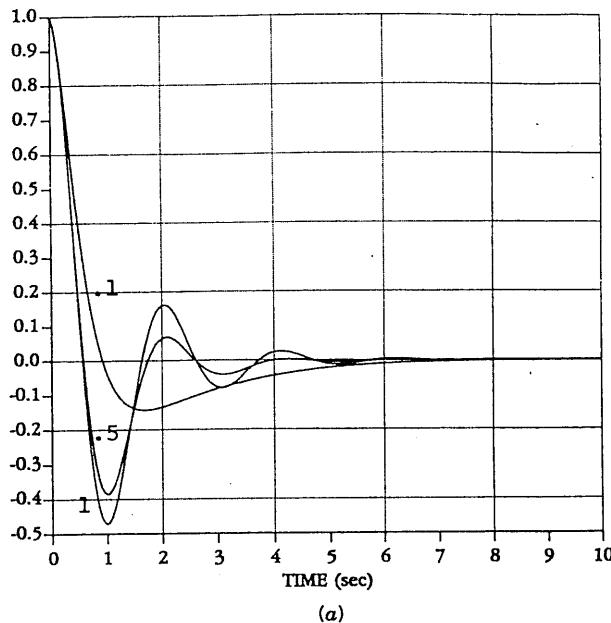
In Section 5.5 we shall show how to overcome both of these deficiencies: the former using "constrained output feedback" and the latter using time weighting like t^k in the PI.

c. *Effect of Weighting Parameters.* It is interesting to examine more closely the effects of the design parameters, namely, the entries of the PI weighting matrices Q and R . Using the same Q as above, we show the sideslip response in Fig. 5.3-2a for control weightings of $\rho = 0.1, 0.5$, and 1 . Increased control weighting in the PI generally suppresses the control signals in the closed-loop system; that is, less control effort is allowed. As less control effort is allowed, the control is less effective in controlling the modes. Indeed, according to the figure, as ρ increases the undershoot in β increases. Moreover, with increasing ρ the control is also less effective in suppressing the undesirable oscillations in the dutch roll mode which were noted in the open-loop system.

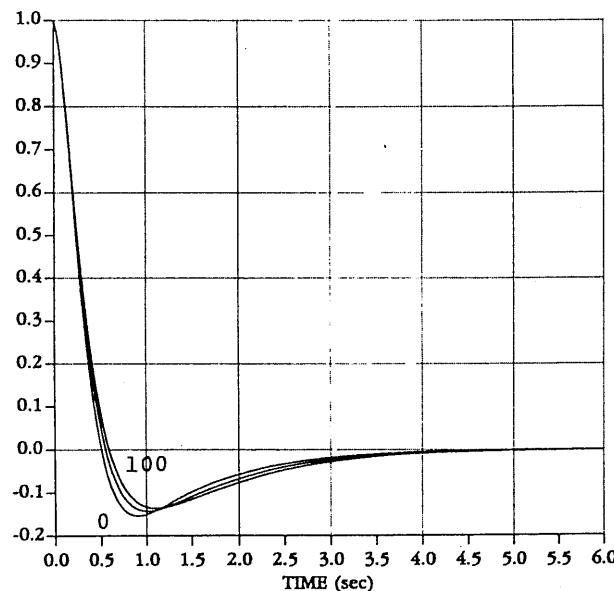
As far as the effect of the dutch roll weighting q_{dr} goes, examine Fig. 5.3-2b, where $\rho = 0.1$ and $q_r = 100$ as in part a, but the sideslip response is shown for $q_{dr} = 0, 50$, and 100 . As q_{dr} increases, the undershoot decreases, reflecting the fact that increased weighting on β^2 in the PI will result in smaller excursions in β in closed-loop.

One last point is worth noting. The open-loop system is stable; therefore, it is clear that it is detectable, since all the unstable modes are observable for any choice of Q (there are no unstable modes). Thus the design would work if we omitted the weighting on x_w^2 in the Q matrix (although, it turns out, the closed-loop poles are not as good). In general, however, the detectability condition is difficult to check in large systems that are open-loop unstable; thus the observability condition is used instead. Failing to weight an undetectable state can lead to some zero elements of K , meaning that some feedback loops are not closed. Thus, to guarantee that this does not occur, Q should be selected so that (\sqrt{Q}, A) is observable.

To avoid all this discussion on observability, we may simply use a modified nonstandard PI with weighting like t^k . Such a PI is introduced in Section 5.5 and leads to a simplified design procedure.



(a)



(b)

Fig. 5.3-2 Effect of PI weighting parameters: (a) sideslip as a function of ρ ($\rho = 0.1, 0.5, 1$); (b) sideslip as a function of q_{dr} ($q_{dr} = 0, 50, 100$).

d. Gain Scheduling. For implementation on an aircraft, the control gains in (13) should be gain scheduled. To accomplish this, the nonlinear aircraft equations are linearized at several equilibrium flight conditions over the desired flight envelope to obtain state-variable models like (4) with different A and B matrices. Then the LQR design is repeated for those different systems.

A major advantage of LQR design will quickly be apparent, for once the control structure has been selected, it takes only a minute or two to run the software to find the optimal gains for a new A and B using the design equations in Table 5.3-1. Note that the optimal gains for one point in the gain schedule can be used as initial stabilizing gains in the LQ solution algorithm for the next point.

It is important, however, to be aware of an additional consideration. The optimal gains at each gain scheduling point should guarantee *robust stability and performance*; that is, they should guarantee stability and good performance at points *near* the design equilibrium point. Such robust stability can be verified after the LQ design by using multivariable frequency-domain techniques. These techniques are developed in Section 6.2, where the remarks on robustness to plant parameter variations are particularly relevant to gain scheduling. ■

5.4 TRACKING A COMMAND

In aircraft control we are often interested not in regulating the state near zero, which we discussed in the preceding section, but in *following a nonzero reference command signal*. For example, we may be interested in designing a control system for optimal step-response shaping. This reference-input tracking or *servodesign* problem is important in the design of command augmentation systems (CAS), where the reference command may be, for instance, desired pitch rate or normal acceleration. In this section and the next we cover tracker design.

It should be mentioned that the *optimal* linear quadratic (LQ) tracker of modern control is not a causal system [Lewis, 1986]. It depends on solving an “adjoint” system of differential equations backward in time, and so is impossible to implement. A suboptimal “steady-state” tracker using full state-variable feedback is available, but it offers no convenient structure for the control system in terms of desired dynamics such as PI control, washout filters, and so on. Thus there have been problems with using it in aircraft control.

Modified versions of the LQ tracker have been presented in [Davison and Ferguson, 1981] and [Gangsaas et al., 1986]. There, controllers of desired structure can be designed since the approaches are output-feedback based. The optimal gains are determined numerically to minimize a PI with, possibly, some constraints.

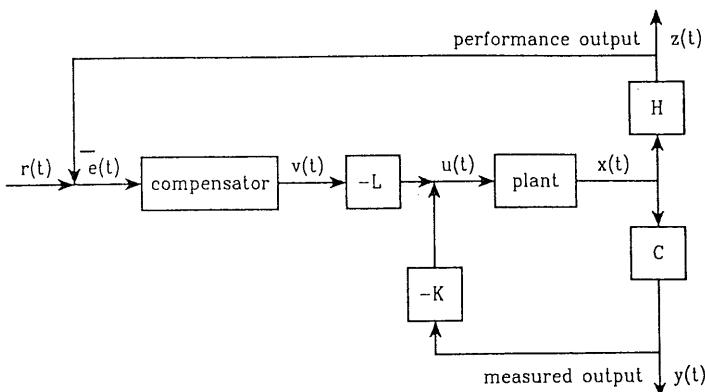


Fig. 5.4-1 Plant with compensator of desired structure.

It is possible to design a tracker by first designing a regulator using, for instance, Table 5.3-1. Then some feedforward terms are added to guarantee perfect tracking [Kwakernaak and Sivan, 1972]. The problem with this technique is that the resulting tracker has no convenient structure and often requires derivatives of the reference command input. Moreover, servosystems designed using this approach depend on knowing the dc gain exactly. If the dc gain is not known exactly, the performance deteriorates. That is, the design is *not robust* to uncertainties in the model.

Here we discuss an approach to the design of tracking control systems which is more useful in aircraft control applications [Stevens et al., 1991]. This approach will allow us to design a servo control system that has any structure desired. This structure will include a unity-gain outer loop that feeds the performance output back and subtracts it from the reference command, thus defining a tracking error $e(t)$ which should be kept small (see Fig. 5.4-1). It can also include compensator dynamics such as a washout filter or an integral controller. The control gains are chosen to minimize a quadratic performance index (PI). We are able to give explicit design equations for the control gains (see Table 5.4-1), which may be solved using the software described in Appendix B.

A problem with the tracker developed in this section is the need to select the design parameters Q and R in the PI in Table 5.4-1. There are some intuitive techniques available for choosing these parameters (see Section 5.3); however, in Section 5.5 we shall show how modified PIs may be used to make the selection of Q and R almost transparent, yielding tracker design techniques that are very convenient for use in aircraft control systems design. We shall show, in fact, that *the key to achieving required performance using modern design strategies is in selecting an appropriate PI*.

Tracker with Desired Structure

In aircraft controls design there is a wealth of experience and knowledge that dictates in many situations what sort of compensator dynamics yield good performance from the point of view of both the controls engineer and the pilot. For example, a washout circuit may be required, or it may be necessary to augment some feedforward channels with integrators to obtain a steady-state error of exactly zero.

The control system structures used in classical aircraft design also give good *robustness properties*. That is, they perform well even if there are disturbances or uncertainties in the system. Thus the multivariable approach developed here usually affords this robustness. Formal techniques for verifying closed-loop robustness for multivariable control systems are given in Chapter 6.

Our approach to tracker design allows controller dynamics of any desired structure and then determines the control gains that minimize a quadratic PI over that structure. Before discussing the tracker design, let us recall from Section 3.8 how compensator dynamics may be incorporated into the aircraft state equations.

A dynamic compensator of prescribed structure may be incorporated into the system description as follows.

Consider the situation in Fig. 5.4-1 where the plant is described by

$$\dot{x} = Ax + Bu \quad (5.4-1)$$

$$y = Cx \quad (5.4-2)$$

with state $x(t)$, control input $u(t)$, and $y(t)$ the *measured output* available for feedback purposes. In addition,

$$z = Hx \quad (5.4-3)$$

is a *performance output*, which must track the given *reference input* $r(t)$. The performance output $z(t)$ is not generally equal to $y(t)$.

It is important to realize that for perfect tracking it is necessary to have as many control inputs in vector $u(t)$ as there are command signals to track in $r(t)$ [Kwakernaak and Sivan, 1972].

The dynamic compensator has the form

$$\begin{aligned}\dot{w} &= Fw + Ge \\ v &= Dw + Je\end{aligned} \quad (5.4-4)$$

with state $w(t)$, output $v(t)$, and input equal to the *tracking error*

$$e(t) = r(t) - z(t). \quad (5.4-5)$$

F , G , D , and J are known matrices chosen to include the desired structure in the compensator.

The allowed form for the plant control input is

$$u = -Ky - Lv, \quad (5.4-6)$$

where the constant gain matrices K and L are to be chosen in the controls design step to result in satisfactory tracking of $r(t)$. This formulation allows for both feedback and feedforward compensator dynamics.

As we have seen in Chapter 3, these dynamics and output equations may be written in augmented form as

$$\frac{d}{dt} \begin{bmatrix} x \\ w \end{bmatrix} = \begin{bmatrix} A & 0 \\ -GH & F \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ G \end{bmatrix} r \quad (5.4-7)$$

$$\begin{bmatrix} y \\ v \end{bmatrix} = \begin{bmatrix} C & 0 \\ -JH & D \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ J \end{bmatrix} r \quad (5.4-8)$$

$$z = [H \ 0] \begin{bmatrix} x \\ w \end{bmatrix}, \quad (5.4-9)$$

and the control input may be expressed as

$$u = -[K \ L] \begin{bmatrix} y \\ v \end{bmatrix}. \quad (5.4-10)$$

Note that in terms of the augmented plant/compensator state description, the admissible controls are represented as a *constant output feedback* [K L]. In the augmented description, all matrices are known except the gains K and L , which need to be selected to yield acceptable closed-loop performance.

A comment on the compensator matrices F , G , D , and J is in order. Often, these matrices are completely specified by the structure of the compensator. Such is the case, for instance, if the compensator contains integrators. However, if it is desired to include a washout or a lead-lag, it may not be clear exactly how to select the time constants. In such cases, engineering judgment will usually give some insight. However, it may sometimes be necessary to go through the design to be proposed, and then if required, return to readjust F , G , D , and J and reperform the design.

LQ Formulation of the Tracker Problem

By redefining the state, the output, and the matrix variables to streamline the notation, we see that the augmented equations (5.4-7)–(5.4-9) that contain the dynamics of both the aircraft and the compensator are of the form

$$\dot{x} = Ax + Bu + Gr \quad (5.4-11)$$

$$y = Cx + Fr \quad (5.4-12)$$

$$z = Hx. \quad (5.4-13)$$

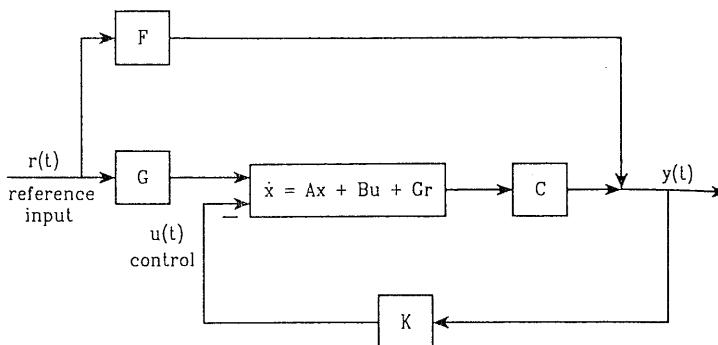


Fig. 5.4-2 Plant/feedback structure.

In this description, let us take the state $x(t) \in \mathbf{R}^n$, control input $u(t) \in \mathbf{R}^m$, reference input $r(t) \in \mathbf{R}^q$, performance output $z(t) \in \mathbf{R}^q$, and measured output $y(t) \in \mathbf{R}^p$. The admissible controls (5.4-10) are proportional output feedbacks of the form

$$u = -Ky = -KCx - KFr \quad (5.4-14)$$

with constant gain K to be determined. This situation corresponds to the block diagram in Fig. 5.4-2. Since K is an $m \times p$ matrix, we intend to close all the feedback loops simultaneously by computing K .

Using these equations the closed-loop system is found to be

$$\begin{aligned} \dot{x} &= (A - BKC)x + (G - BKF)r \\ &\equiv A_c x + B_c r. \end{aligned} \quad (5.4-15)$$

In the remainder of this subsection, we shall use the formulation (5.4-11)–(5.4-14), assuming that the compensator, if required, has already been included in the system dynamics and demonstrating how to select the constant output-feedback gain matrix K using LQ techniques.

Our formulation differs sharply from the traditional formulations of the optimal tracker problem [Kwakernaak and Sivan, 1972; Lewis, 1986]. Note that (5.4-14) includes both feedback and feedforward terms, so that both the closed-loop poles and compensator zeros may be affected by varying the gain K (see Example 5.4-1). Thus we should expect better success in shaping the step response than by placing only the poles.

Since the performance specifications of aircraft are often given in terms of time-domain criteria [Mil. Spec. 1797, 1987] (see Section 4.3) and these criteria are closely related to the step response, we shall assume henceforth that the reference input $r(t)$ is a step command with magnitude r_0 . Designing for such a command will yield suitable time-response characteristics. Although our design is based on step-response shaping, it should be clearly

realized that the resulting control system, if properly designed, will give good time responses for *any arbitrary reference command signal* $r(t)$.

Let us now formulate an optimal control problem for selecting the control gain K to guarantee tracking of $r(t)$. Then we shall derive the design equations in Table 5.4-1, which are used to determine the optimal K . These equations are solved using software like that described in Appendix B.

The Deviation System. Denote steady-state values by overbars and deviations from the steady-state values by tildes. Then the state, output, and control deviations are given by

$$\tilde{x}(t) = x(t) - \bar{x} \quad (5.4-16)$$

$$\tilde{y}(t) = y(t) - \bar{y} = K\tilde{x} \quad (5.4-17)$$

$$\tilde{z}(t) = z(t) - \bar{z} = H\tilde{x} \quad (5.4-18)$$

$$\tilde{u}(t) = u(t) - \bar{u} = -KCx - KFr_0 - (-KC\bar{x} - KFr_0) = -KC\tilde{x}(t)$$

or

$$\tilde{u} = -K\tilde{y}. \quad (5.4-19)$$

The tracking error $e(t) = r(t) - z(t)$ is given by

$$e(t) = \tilde{e}(t) + \bar{e} \quad (5.4-20)$$

with the error deviation given by

$$\tilde{e}(t) = e(t) - \bar{e} = (r_0 - Hx) - (r_0 - H\bar{x}) = -H\tilde{x}$$

or

$$\tilde{e} = -\tilde{z}. \quad (5.4-21)$$

Since in any acceptable design the closed-loop plant will be asymptotically stable, A_c is nonsingular. According to (5.4-15), at steady state

$$0 = A_c\bar{x} + B_cr_0, \quad (5.4-22)$$

so that the steady-state state response \bar{x} is

$$\bar{x} = -A_c^{-1}B_cr_0 \quad (5.4-23)$$

and the steady-state error is

$$\bar{e} = r_0 - H\bar{x} = (I + HA_c^{-1}B_c)r_0. \quad (5.4-24)$$

To understand this expression, note that the closed-loop transfer function from r_0 to z [see (5.4-15) and (5.4-13)] is

$$H(s) = H(sI - A_c)^{-1}B_c. \quad (5.4-25)$$

The steady-state behavior may be investigated by considering the dc value of $H(s)$ (i.e., $s = 0$); this is just $-HA_c^{-1}B_c$, the term appearing in (5.4-24).

Using (5.4-16), (5.4-19), and (5.4-23) in (5.4-15) the closed-loop dynamics of the state deviation are seen to be

$$\dot{\tilde{x}} = A_c \tilde{x} \quad (5.4-26)$$

$$\tilde{y} = C\tilde{x} \quad (5.4-27)$$

$$\tilde{z} = H\tilde{x} = -\tilde{e} \quad (5.4-28)$$

and the control input to the deviation system (5.4-26) is (5.4-19). Thus the step-response shaping problem has been converted to a *regulator problem* for the deviation system

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u}. \quad (5.4-29)$$

Again, we emphasize the difference between our approach and traditional ones (e.g., [Kwakernaak and Sivan, 1972]). Once the gain K in (5.4-19) has been found, the control for the plant is given by (5.4-14), which inherently has both feedback and feedforward terms. Thus no extra feedforward term need be added to make \bar{e} zero.

Performance Index. To make the tracking error $e(t)$ in (5.4-20) small, we propose to attack two equivalent problems: the problem of regulating the error deviation $\tilde{e}(t) = -\tilde{z}(t)$ to zero, and the problem of making small the steady-state error \bar{e} .

Note that we do not assume a Type I system which would force \bar{e} to be equal to zero. This can be important in aircraft controls, where it may not be desirable to force the system to be of Type I by augmenting all control channels with integrators. This augmentation complicates the servo structure. Moreover, it is well known from classical control theory that suitable step responses may often be obtained without resorting to inserting integrators in all the feedforward channels.

To make small both the error deviation $\tilde{e}(t) = -H\tilde{x}(t)$ and the steady-state error \bar{e} , we propose selecting K to minimize the performance index (PI)

$$J = \frac{1}{2} \int_0^{\infty} (\tilde{e}^T \tilde{e} + \tilde{u}^T R \tilde{u}) dt + \frac{1}{2} \bar{e}^T V \bar{e} \quad (5.4-30)$$

with $R > 0$, $V \geq 0$, design parameters. The integrand is the standard

quadratic PI with, however, a weighting V included on the steady-state error. Note that the PI weights the control *deviations* and not the controls themselves. If the system is of Type I, containing integrators in all the feedforward paths, then V may be set to zero since the steady-state error is automatically zero.

Making small the error deviation $\tilde{e}(t)$ improves the transient response, while making small the steady-state error $\bar{e}(t)$ improves the steady-state response. If the system is of Type 0, these effects involve a trade-off, so that then there is a design trade-off involved in selecting the size of V .

We can generally select $R = rI$ and $V = vI$, with r and v scalars. This simplifies the design since now only a few parameters must be tuned during the interactive design process.

According to (5.4-21), $\tilde{e}^T \tilde{e} = \tilde{x}^T H^T H \tilde{x}$. Referring to Table 5.3-1, therefore, it follows that the matrix Q there is equal to $H^T H$, where H is known. That is, weighting the error deviation in the PI has already shown us how to select the design parameter Q , affording a considerable simplification.

The problem we now have to solve is how to select the control gains K to minimize the PI J for the deviation system (5.4-29). Then the tracker control for the original system is given by (5.4-14).

We should point out that the proposed approach is suboptimal in the sense that minimizing the PI does not necessarily minimize a quadratic function of the total error $e(t) = \bar{e} + \tilde{e}(t)$. It does, however, guarantee that both $\tilde{e}(t)$ and \bar{e} are small in the closed-loop system, which is a design goal.

Solution of the LQ Tracker Problem

It is now necessary to solve for the optimal feedback gain K that minimizes the PI. The design equations needed are now derived. They appear in Table 5.4-1.

By using (5.4-26) and a technique like the one used in Section 5.3 (see problems), the optimal cost is found to satisfy

$$J = \frac{1}{2} \tilde{x}^T(0) P \tilde{x}(0) + \frac{1}{2} \tilde{e}^T V \tilde{e} \quad (5.4-31)$$

with $P \geq 0$ the solution to

$$0 = g \equiv A_c^T P + PA_c + Q + C^T K^T R K C \quad (5.4-32)$$

with $Q = H^T H$ and \bar{e} given by (5.4-24).

In our discussion of the linear quadratic regulator we assumed that the initial conditions were uniformly distributed on a surface with known characteristics. While this is satisfactory for the regulator problem, it is an unsatisfactory assumption for the tracker problem. In the latter situation the system starts at rest and must achieve a given final state that is dependent on the reference input, namely (5.4-23). To find the correct value of $\tilde{x}(0)$, we note

that since the plant starts at rest [i.e., $x(0) = 0$], according to (5.4-16),

$$\bar{x}(0) = -\bar{x}, \quad (5.4-33)$$

so that the optimal cost (5.4-31) becomes

$$J = \frac{1}{2}\bar{x}^T P \bar{x} + \frac{1}{2}\bar{e}^T V \bar{e} = \frac{1}{2} \text{tr}(P X) + \frac{1}{2}\bar{e}^T V \bar{e}, \quad (5.4-34)$$

with P given by (5.4-32), \bar{e} given by (5.4-24), and

$$X \equiv \bar{x} \bar{x}^T = A_c^{-1} B_c r_0 r_0^T B_c^T A_c^{-T}, \quad (5.4-35)$$

with $A_c^{-T} = (A_c^{-1})^T$.

The optimal solution to the unit-step tracking problem, with (5.4-11) initially at rest, may now be determined by minimizing J in (5.4-34) over the gains K , subject to the constraint (5.4-32) and equations (5.4-24) and (5.4-35).

This algebraic optimization problem can be solved by any well-known numerical method (see [Press et al., 1986; Söderström, 1978]). A good approach for a fairly small number ($mp \leq 10$) of gain elements in K is the SIMPLEX minimization routine [Nelder and Mead, 1964]. To evaluate the PI for each fixed value of K in the iterative solution procedure, one may solve (5.4-32) for P using subroutine ATXPXA in [Bartels and Stewart, 1972] and then employ (5.4-34). Software for determining the optimal control gains K is described in Appendix B.

Design Equations for Gradient-Based Solution. As an alternative solution procedure, one may use gradient-based techniques (e.g., the Davidon–Fletcher–Powell algorithm [Press et al., 1986]), which are generally faster than non-gradient-based approaches.

To find the gradient of the PI with respect to the gains, define the Hamiltonian

$$\mathcal{H} = \text{tr}(P X) + \text{tr}(g S) + \frac{1}{2}\bar{e}^T V \bar{e} \quad (5.4-36)$$

with S a Lagrange multiplier. Now, using the basic matrix calculus identities,

$$\frac{\partial Y^{-1}}{\partial x} = -Y^{-1} \frac{\partial Y}{\partial x} Y^{-1} \quad (5.4-37)$$

$$\frac{\partial U V}{\partial x} = \frac{\partial U}{\partial x} V + U \frac{\partial V}{\partial x} \quad (5.4-38)$$

$$\frac{\partial y}{\partial x} = \text{tr} \left[\frac{\partial y}{\partial z} \cdot \frac{\partial z^T}{\partial x} \right] \quad (5.4-39)$$

we may proceed as in the preceding section, with, however, a little more patience due to the extra terms (see the problems!), to obtain the necessary conditions for a solution given in Table 5.4-1.

To find K by a gradient minimization algorithm, it is necessary to provide the algorithm with the values of J and $\partial J/\partial K$ for a given K . The value of J is given by the expression in Table 5.4-1 for the optimal cost. To find $\partial J/\partial K$ given K , solve (5.4-40)–(5.4-41) for P and S . Then since these equations hold, $\partial J/\partial K = \partial \mathcal{H}/\partial K$, which may be found using (5.4-42). These equations

TABLE 5.4-1. LQ Tracker with Output Feedback

System Model

$$\begin{aligned}\dot{x} &= Ax + Bu + Gr \\ y &= Cx + Fr \\ z &= Hx\end{aligned}$$

Control

$$u = -Ky$$

Performance Index

$$J = \frac{1}{2} \int_0^{\infty} (\tilde{x}^T Q \tilde{x} + \tilde{u}^T R \tilde{u}) dt + \frac{1}{2} \bar{e}^T V \bar{e}, \quad \text{with } Q = H^T H$$

Optimal Output Feedback Gain

$$0 = \frac{\partial \mathcal{H}}{\partial S} = A_c^T P + PA_c + Q + C^T K^T R K C \quad (5.4-40)$$

$$0 = \frac{\partial \mathcal{H}}{\partial P} = A_c S + S A_c^T + X \quad (5.4-41)$$

$$0 = \frac{1}{2} \frac{\partial \mathcal{H}}{\partial K} = R K C S C^T - B^T P S C^T + B^T A_c^{-T} (P + H^T V H) \bar{x} \bar{y}^T - B^T A_c^{-T} H^T V r_0 \bar{y}^T \quad (5.4-42)$$

with r a unit step of magnitude r_0 and

$$\bar{x} = -A_c^{-1} B_c r_0 \quad (5.4-43)$$

$$\bar{y} = C \bar{x} + F r_0 \quad (5.4-44)$$

$$X = \bar{x} \bar{x}^T = A_c^{-1} B_c r_0 r_0^T B_c^T A_c^{-T} \quad (5.4-45)$$

where

$$A_c = A - B K C, \quad B_c = G - B K F$$

Optimal Cost

$$J = \frac{1}{2} \text{tr}(P X) + \frac{1}{2} \bar{e}^T V \bar{e}$$

should be compared to those in Table 5.3-1. Note that the dependence of X on the gain K [see (5.4-45)] and the presence of \bar{e} in the PI have resulted in extra terms being added in (5.4-42).

Determining the Optimal Feedback Gain

The issues in finding the optimal output-feedback gain K in the tracker problem of Table 5.4-1 are the same as those discussed in connection with the regulator problem of Table 5.3-1. They are: choice of Q to satisfy detectability, choice of solution technique, finding an initial stabilizing gain, and iterative design by tuning Q and R .

We emphasize that there are only a few design parameters in our approach, namely r and v (since we can generally select $R = rI$, $V = vI$). Thus it is not difficult or time consuming to come up with good designs. Much of the simplicity of our approach derives from the fact that Q in the PI is equal to $H^T H$, which is known.

Let us now illustrate the servo design procedure by an example.

Example 5.4-1: Normal Acceleration CAS. In Example 4.5-3 we designed a normal acceleration CAS using classical control theory. In that example, successive loop closures were used with root-locus design to obtain the feedback gains. Here we shall show that using the LQ design equations in Table 5.4-1 we can close all the loops simultaneously. Thus the design procedure is more straightforward. We shall also demonstrate that using LQ design, *the algorithm automatically selects the zero of the compensator for optimal performance*.

a. *Control System Structure.* The normal acceleration control system is shown in Fig. 5.4-3, where r is a reference step input in g's and $u(t)$ is the elevator actuator voltage. An integrator has been added in the feedforward path to achieve zero steady-state error. The performance output that should track the reference command r is $z = n_z$, so that the tracking error is $e = r - n_z$. The state and measured output are

$$x = \begin{bmatrix} \alpha \\ q \\ \delta_e \\ \alpha_F \\ \epsilon \end{bmatrix}, \quad y = \begin{bmatrix} \alpha_F \\ q \\ e \\ \epsilon \end{bmatrix}, \quad (1)$$

with $\epsilon(t)$ the integrator output and α_F the filtered measurements of angle of attack.

Exactly as in Example 4.5-3, we linearized the F-16 dynamics about the nominal flight condition in Table 3.4-3 (502 ft/s, level flight, dynamic pressure of 300 psf, $x_{cg} = 0.35\bar{c}$) and augmented the dynamics to include the

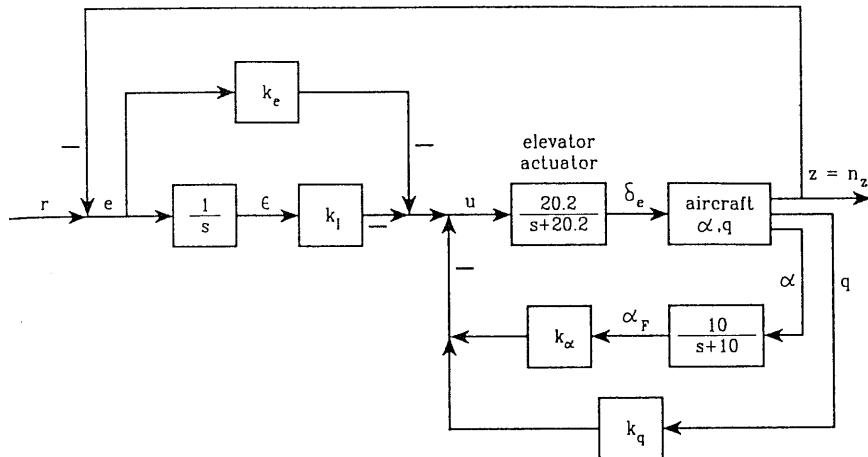


Fig. 5.4-3 G-command system.

elevator actuator, angle-of-attack filter, and compensator dynamics. The result is

$$\dot{x} = Ax + Bu + Gr \quad (2)$$

$$y = Cx + Fr \quad (3)$$

$$z = Hx, \quad (4)$$

with

$$A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 & 0 & 0 \\ 0.82225 & -1.07741 & -0.17555 & 0 & 0 \\ 0 & 0 & -20.2 & 0 & 0 \\ 10 & 0 & 0 & -10 & 0 \\ -16.26 & -0.9788 & 0.04852 & 0 & 0 \end{bmatrix}, \quad (5a)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 20.2 \\ 0 \\ 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 57.2958 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 & 0 \\ -16.26 & -0.9788 & 0.04852 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad (5b)$$

$$H = [-16.26 \quad 0.9788 \quad -0.04852 \quad 0 \quad 0]. \quad (5c)$$

The factor of 57.2958 is added to convert angles from radians to degrees.

The control input is

$$u = -Ky = -[k_\alpha \ k_q \ k_e \ k_I]y = -k_\alpha\alpha_F - k_qq - k_ee - k_I\epsilon. \quad (6)$$

It is desired to select the four control gains to guarantee a good response to a step command r . Note that k_α and k_q are feedback gains, while k_e and k_I are feedforward gains.

Note that the proportional-plus-integral compensator is given by

$$k_e + \frac{k_I}{s} = k_e \frac{s + k_I/k_e}{s}, \quad (7)$$

which has a zero at $s = -k_I/k_e$. Since the LQ design algorithm will select all four control gains, it will *automatically select the optimal location for the compensator zero*.

b. Performance Index and Determination of the Control Gains. Due to the integrator, the system is of Type I. Therefore, the steady-state error \bar{e} is automatically equal to zero. A natural PI thus seems to be

$$J = \frac{1}{2} \int_0^\infty (\tilde{e}^2 + \rho \tilde{u}^2) dt \quad (8)$$

with ρ a scalar weighting parameter. Since $\tilde{e} = H\tilde{x}$, this corresponds to the PI in Table 5.4-1 with

$$Q = H^T H = \begin{bmatrix} 264 & 16 & 1 & 0 & 0 \\ 16 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (9)$$

This is, unfortunately, not a suitable Q matrix since (H, A) is not observable in open loop. Indeed, according to Fig. 5.4-3, observing the first two states α and q can never give information about ϵ in the open-loop configuration (where the control gains are zero). Thus the integrator state is unobservable in the PI. Since the integrator pole is at $s = 0$, (H, A) is undetectable (unstable unobservable pole), so that any design based on (9) would, in fact, yield a value for the integral gain of $k_I = 0$.

We shall show in Section 5.5 a very convenient way to correct problems like this. There we shall introduce a time weighting of t^k into the PI. In the

meantime, to correct the observability problem here, let us select

$$Q = H^T H = \begin{bmatrix} 264 & 16 & 1 & 0 & 0 \\ 16 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10)$$

where we include a weighting on $\epsilon(t)$ to make it observable in the PI.

Now, we selected $\rho = 1$ and solved the design equations in Table 5.4-1 for the optimal control gain K using the software described in Appendix B. For this Q and ρ the feedback matrix was

$$K = [0.006 \quad -0.152 \quad 1.17 \quad 0.996] \quad (11)$$

and the closed-loop poles were

$$\begin{aligned} s = & -1.15 \pm j0.69 \\ & -1.60, -9.98, -19.54. \end{aligned} \quad (12)$$

These yield a system that is not fast enough; the complex pair is also unsuitable in terms of flying-qualities requirements.

After repeating the design using several different Q and ρ , we decided on

$$Q = H^T H = \begin{bmatrix} 264 & 16 & 1 & 0 & 0 \\ 16 & 60 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 \end{bmatrix}, \quad (13)$$

$\rho = 0.01$. The decreased control weighting ρ has the effect of allowing larger control effort and so speeding up the response. The increased weighting on the integrator output $\epsilon(t)$ has the effect of forcing n_z to its final value of r more quickly, hence also speeding up the response. The increased weighting on the second state component q has the effect of regulating excursions in $\tilde{q}(t)$ closer to zero, and hence of providing increased damping.

With this Q and ρ the control matrix was

$$K = [-1.629 \quad -1.316 \quad 18.56 \quad 77.6] \quad (14)$$

and the closed-loop poles were at

$$\begin{aligned} s = & -2.98 \pm j3.17, \\ & -19.31 \pm j4.64 \\ & -5.91. \end{aligned} \quad (15)$$

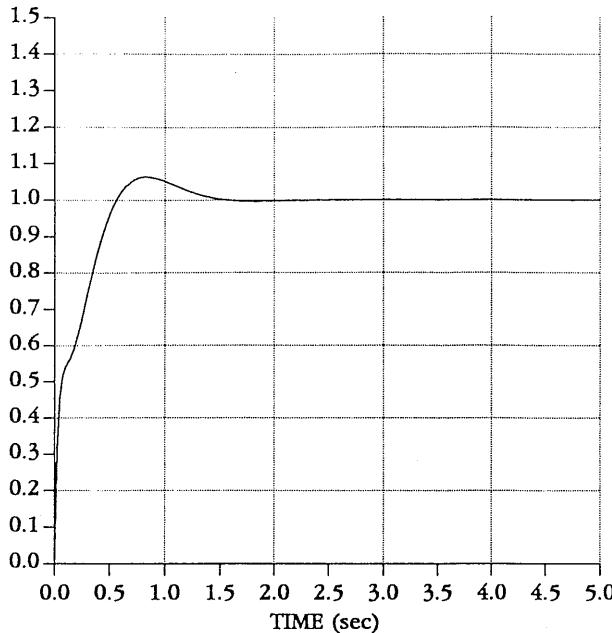


Fig. 5.4-4 Normal acceleration step response.

The closed-loop step response is shown in Fig. 5.4-4; it is fairly fast with an overshoot of 6%. Note the hump in the initial response due to the non-minimum-phase zero (see Section 4.5). Further tuning of the elements of Q and R could provide less overshoot, a faster response, and a smaller gain for the angle-of-attack feedback. (It is worth noting that we shall obtain a far better response with more reasonable gains in Example 5.5-2, where we use a PI with time-dependent weighting like t^k .)

According to (7), the compensator zero has been placed by the LQ algorithm at

$$s = -\frac{k_I}{k_e} = -4.18. \quad (16)$$

Using the software described in Appendix B, the entire design, including determining K for different choices of Q and ρ until a suitable design was reached, took about 30 minutes.

c. *Discussion.* We can now emphasize an important aspect of modern LQ design. As long as $Q \geq 0$, $R > 0$, and (\sqrt{Q}, A) is observable, the closed-loop system designed using Table 5.4-1 is generally stable. Thus the LQ theory has allowed us to tie the control system design to some *design parameters* which

may be tuned to obtain acceptable behavior—namely, the elements of weighting matrices Q and R . Using the software described in Appendix B, for a given Q and R the optimal gain K is easily found. If it is not suitable in terms of time responses and closed-loop poles, the elements of Q and R can be changed and the design repeated. The importance of this is that for admissible Q and R , *closed-loop stability is guaranteed*.

A disadvantage of the design equations in Table 5.4-1 is the need to try different Q and R until suitable performance is obtained, as well as the need for (H, A) to be observable. In Section 5.5 we shall introduce a different PI with time weighting of t^k which eliminates these deficiencies.

Another point needs to be made. Using the control (6)/(3) in (2) yields the closed-loop plant

$$\dot{x} = (A - BKC)x + (G - BKF)r \quad (17)$$

whence the closed-loop transfer function from $r(t)$ to $z(t)$ is

$$H(s) = H(sI - (A - BKC))^{-1}(G - BKF). \quad (18)$$

Note that the transfer function numerator depends on the optimal gain K . That is, this scheme uses optimal positioning of *both the poles and zeros* to attain step-response shaping.

d. Selection of Initial Stabilizing Gain. In order to initialize the algorithm that determines the optimal K by solving the design equations in Table 5.4-1, it is necessary to find an initial gain that stabilizes the system. In this example we simply selected gains with signs corresponding to the static loop sensitivity of the individual transfer functions, since this corresponds to negative feedback (see Section 3.7 and Chapter 4). The static loop sensitivities from u to α and from u to q are negative, so negative gains were chosen for these loops. The initial gain used was

$$K = [-0.1 \quad -0.1 \quad 1 \quad 1]. \quad (19)$$

5.5 MODIFYING THE PERFORMANCE INDEX

Modern control theory affords us the ability to close all the feedback loops simultaneously by solving matrix equations for the gain matrix K . With a sensible problem formulation, it also *guarantees the stability of the closed-loop system*. These two fundamental properties make modern design very useful for aircraft control systems. One should recall the difficulty in guaranteeing

closed-loop stability in multiloop control systems using one-loop-at-a-time design (Chapter 4).

An additional important advantage is as follows. The standard aircraft control system structures used in classical design have been developed to yield good *robustness properties*. That is, they yield good performance even if there are disturbances in the systems, or modeling inaccuracies such as plant parameter variations or high-frequency unmodeled dynamics (e.g., flexible aircraft modes). Since the approach described here allows these standard structures to be incorporated into the control system, it generally yields robust control systems. We shall discuss procedures for formally verifying robustness in Chapter 6.

In the LQ regulator design method of Section 5.3 and the LQ tracker design method of Section 5.4, it was necessary to select the performance index (PI) weighting matrices Q and R as design parameters. Moreover, it was necessary to satisfy an observability property in selecting Q . There are some good approaches that give guidance in selecting Q , such as Bryson's approach (see Section 5.3). Note also that in Table 5.4-1, $Q = H^T H$, where H is known. However, due to the observability requirement the design parameters Q and R do not necessarily correspond to actual performance objectives.

In this section we show how to modify the PI to considerably simplify the selection of the weighting matrices Q and R in Table 5.4-1. The observability of (\sqrt{Q}, A) will be unnecessary. The PIs shown in this section correspond to actual performance objectives and involve only a few design parameters, even for systems with many states and many control gains to determine. These facts, coupled with the capability already demonstrated of employing a compensator with any desired structure, will result in a powerful and convenient approach to the design of multivariable aircraft control systems.

A wide range of performance objectives may be attained by using modifications of the PI. We shall consider several modifications, all of which are useful depending on the performance objectives. The important concept to grasp is that *the key to obtaining suitable closed-loop behavior using LQ design lies in selecting a suitable PI for the problem at hand*. At the end of the section we present several examples in aircraft controls design to demonstrate this issue as well as the directness of the approach.

We shall again be concerned with the system-plus-compensator

$$\dot{x} = Ax + Bu + Gr. \quad (5.5-1)$$

We are trying to determine controls that are static output feedbacks of the form

$$u = -Ky \quad (5.5-2)$$

with

$$y = Cx + Fr \quad (5.5-3)$$

the measured output and

$$z = Hx \quad (5.5-4)$$

the performance output, which is to track the reference command r . If we are interested in regulation and not tracking, then G and F do not appear in the equations and z is not defined.

Constrained Feedback Matrix

In many applications it is desired for certain elements of the feedback gain matrix K to be zero to avoid coupling between certain output/input pairs. Zeroing certain gains allows us to specify the detailed structure of the control system. For instance, it may be desired that the error in channel 1 of the controller not be coupled to the control input in channel 2. Zeroing some gains also simplifies the gain-scheduling problem by reducing the number of nonzero gains requiring tabulation. This is called *constrained* output-feedback design.

Gain Element Weighting. Certain elements k_{ij} of K can be made small simply by weighting them in the performance index, that is, by selecting a PI like

$$J = \frac{1}{2} \int_0^{\infty} (\tilde{x}^T Q \tilde{x} + \tilde{u}^T R \tilde{u}) dt + \sum_i \sum_j g_{ij} k_{ij}^2. \quad (5.5-5)$$

Gain-element weight g_{ij} is chosen large to make the (i, j) th element k_{ij} of the feedback matrix K small in the final design. Then, in implementing the controller, the small elements of K may simply be set to zero.

The design problem is now to minimize

$$J = \frac{1}{2} \text{tr}(P X) + \sum_i \sum_j g_{ij} k_{ij}^2, \quad (5.5-6)$$

with P satisfying the matrix equation in Table 5.3-1 or Table 5.4-1, as appropriate. This may be accomplished by using the equations in Table 5.3-1 (if we are interested in regulation) or Table 5.4-1 (if we are interested in tracking) to numerically minimize the PI, but with the extra term involving the gain weighting that appears in (5.5-6) [Moerder and Calise, 1985].

Computing an Initial Stabilizing Gain. The iterative algorithms that solve the design equations in Tables 5.3-1 and 5.4-1 require initial stabilizing gains. Unfortunately, stabilizing output-feedback gains can be complicated to find in large multivariable systems. A few ways to find K_0 so that $(A - BK_0C)$ is stable were mentioned in Section 5.3 and Example 5.4-1d. Gain element

weighting can be used to solve the problem of determining an initial stabilizing output-feedback gain, as we now see.

There are many techniques for finding a stabilizing *state-variable* feedback given the plant system matrix A and control matrix B (see [Kailath, 1980], as well as Section 5.7). That is, it is straightforward to find a K_0 so that $(A - BK_0)$ is stable. Routines that perform this are available in standard software packages such as ORACLS [Armstrong, 1980]. Unfortunately, for flight controls purposes, state-feedback design is unsuitable for reasons such as those we have discussed. However, suppose that an $m \times n$ stabilizing state-feedback gain has been found. Then, to determine an $m \times p$ output-feedback gain, it is only necessary to weight in the PI the elements of the state feedback matrix that do not correspond to measured outputs. The algorithm will then provide a suitable output feedback gain matrix by driving those elements to zero.

Gain Element Fixing. There is an attractive alternative to gain element weighting for fixing gain matrix elements. If a numerical technique such as simplex [Press et al., 1986] is used to determine the optimal control by varying K and directly evaluating J , we may simply fix certain elements of K and not allow the simplex to vary them. This allows the fixed elements to be retained at any desired (possibly nonzero) value and takes far fewer computations than gain element weighting, especially if many elements of K are fixed.

If, on the other hand, a gradient-based routine such as Davidon-Fletcher-Powell [Press et al., 1986] is used in conjunction with the design equations in Tables 5.3-1 or 5.4-1, it is easy to modify the gradient $\partial J / \partial K$ to leave certain elements of K fixed. Indeed, to fix element (i, j) of K , one need only set element (i, j) of $\partial J / \partial K$ equal to zero.

These approaches require fewer operations than the gain weighting approach based on (5.5-5) and are incorporated in the software described in Appendix B, which is called program LQ. Illustrations of controls design using constrained output feedback are provided in the examples.

Derivative Weighting

As we shall soon show in an example, it is often convenient to weight in the PI not the states themselves but their derivatives. This is because rates of change of the states can in some design specifications be more important than the values of the states. For instance, elevator rate of change has a closer connection with required control energy than does elevator deflection. To accommodate such situations, we may consider the PI

$$J = \frac{1}{2} \int_0^{\infty} \dot{x}^T Q \dot{x} dt. \quad (5.5-7)$$

One way to formulate this optimization problem is to convert this PI to one

that weights the states and inputs but has a state/input cross-weighting term [simply substitute (5.4-29) into J]. This optimization problem is solved in [Lewis, 1986].

An alternative (see the problems) is to minimize

$$J = \frac{1}{2} \operatorname{tr}[P\dot{\tilde{x}}(0)\dot{\tilde{x}}^T(0)] \quad (5.5-8)$$

with P the solution to

$$A_c^T P + P A_c + Q = 0. \quad (5.5-9)$$

Again, any optimization technique may be used. More details on this formulation may be found in [Quintana et al., 1976].

In the step-response shaping problem, the value of the initial state derivative vector to use in (5.5-8) is easy to determine, since $x(0) = 0$ and \tilde{x} is a constant so that according to (5.4-16) and (5.4-15)

$$\dot{\tilde{x}}(0) = B_c r_0. \quad (5.5-10)$$

Time-Dependent Weighting

One final form of the PI remains to be discussed. A step response that is apparently good (i.e., fast, with acceptable overshoot and settling time) may contain a slow pole(s) with small residue, so that the response creeps for a long time as it nears its final value. The quadratic performance criterion penalizes small errors relatively lightly and so does not tend to suppress this kind of behavior.

Thus, in the spirit of the classical (ITAE, ISTSE, etc.) performance indices [D'Azzo and Houpis, 1988] we define a PI that contains a time-weighted component:

$$J = \frac{1}{2} \int_0^\infty (t^k \tilde{x}^T P \tilde{x} + \tilde{x}^T Q \tilde{x}) dt. \quad (5.5-11)$$

If we are interested in including a control-weighting term $\tilde{u}^T R \tilde{u}$ in (5.5-11) and in using the output feedback (5.5-2), we may add the term $C^T K^T R K C$ (since $\tilde{u}^T R \tilde{u} = \tilde{x}^T C^T K^T R K C \tilde{x}$) to the appropriate state-weighting matrix P or Q , depending on whether we wish to multiply the control weighting term by t^k . For instance, if the control-input term is not to be weighted by t^k , the PI (5.5-11) takes on the form

$$J = \frac{1}{2} \int_0^\infty [t^k \tilde{x}^T P \tilde{x} + \tilde{x}^T (Q + C^T K^T R K C) \tilde{x}] dt. \quad (5.5-12)$$

If it is desired to have the control weighting multiplied by t^k , the term $C^T K^T R K C$ should be added to P instead of Q .

Whether or not the control effort should be time-weighted is a matter for experiment with the particular design. The time-varying weighting in the PI places a heavy penalty on errors that occur late in the response and is thus very effective in suppressing the effect of a slow pole as well as in eliminating lightly damped settling behavior.

Due to the factor t^k , the optimal gain K that minimizes J is time varying. However, to obtain useful designs we shall determine the suboptimal solution that assumes a time-invariant control gain K . Note that time-varying gains would be very difficult to gain schedule.

We may successively integrate by parts (see the problems) to show that the value of (5.5-12) for a given value of K is given by successively solving the nested Lyapunov equations

$$\begin{aligned} 0 &= g_0 \equiv A_c^T P_0 + P_0 A_c + P \\ 0 &= g_1 \equiv A_c^T P_1 + P_1 A_c + P_0 \\ &\vdots \\ 0 &= g_{k-1} \equiv A_c^T P_{k-1} + P_{k-1} A_c + P_{k-2} \\ 0 &= g_k \equiv A_c^T P_k + P_k A_c + k! P_{k-1} + Q + C^T K^T R K C. \end{aligned} \quad (5.5-13)$$

Then

$$J = \frac{1}{2} \tilde{x}^T(0) P_k \tilde{x}(0) = \frac{1}{2} \bar{x}^T P_k \bar{x} = \frac{1}{2} \text{tr}(P_k X). \quad (5.5-14)$$

A minimization routine such as Simplex [Nelder and Mead, 1964; Press et al., 1986] can be used to find the optimal gains using (5.5-13) and (5.5-14) to evaluate the PI for a specified value of the gain K .

Alternatively, to use a faster gradient-based routine, we may determine the gradient of J with respect to K . To do so, define the Hamiltonian

$$\mathcal{H} = \frac{1}{2} \text{tr}(P_k X) + \text{tr}(g_0 S_0) + \cdots + \text{tr}(g_k S_k), \quad (5.5-15)$$

where $S_i \geq 0$ are matrices of undetermined Lagrange multipliers. Then, by differentiating \mathcal{H} with respect to all variables, necessary conditions for a minimum may be found (see the Problems). These design equations for the LQ tracker with time weighting are summarized in Table 5.5-1.

To use a gradient-based optimization routine such as Davidon-Fletcher-Powell [Press et al., 1986], we may proceed as follows. For a given K , solve the nested Lyapunov equations for P_i and S_i . Since the g_i are then all zero, (5.5-15) shows that $J = \mathcal{H}$. Then (5.5-23) gives the gradient of J with respect to K , which is used by the gradient-based routine to find the updated value of K .

If it is desired to use LQ *regulator* design (as opposed to tracker design, that is, Table 5.3-1) with time-dependent weighting, one need only set $X = I$

[assuming that $E\{x(0)x^T(0)\} = I$] and $\bar{x} = 0$ in the tracker design equations of Table 5.5-1.

Software to determine the optimal value of K given the design parameters k , Q , and R (for both the regulator and tracker) is described in Appendix B. It is called program LQ.

A combination of derivative and time-dependent weighting occurs in the PI:

$$J = \frac{1}{2} \int_0^\infty \left(t^k \dot{\tilde{x}}^T P \dot{\tilde{x}} + \dot{\tilde{x}}^T Q \dot{\tilde{x}} \right) dt. \quad (5.5-16)$$

The optimal gains in this situation may be determined by minimizing

$$J = \frac{1}{2} \dot{\tilde{x}}^T(0) P_k \dot{\tilde{x}}(0) = \frac{1}{2} r_0^T B_c^T P_k B_c r_0 \quad (5.5-17)$$

subject to (5.5-13) with $R = 0$.

A Fundamental Design Property

We now mention a fact of key importance in connection with time-dependent weighting. We shall be very concerned to use PIs that are sensible from a design point of view. That is, we shall not be content to select P and Q in Table 5.5-1 as $n \times n$ matrices and juggle their entries until a suitable design occurs. This sort of approach is one of the fundamental flaws of modern LQ design.

A sensible PI is one of the form

$$J = \frac{1}{2} \int_0^\infty \left(t^k \tilde{e}^T \tilde{e} + r \tilde{u}^T \tilde{u} \right) dt. \quad (5.5-18)$$

where according to Section 5.4, the error deviation is given by

$$\tilde{e} = -H\tilde{x}, \quad (5.5-19)$$

with $z = Hx$ the performance output. This PI corresponds to our desire to make the error small without too much control energy. Since $\tilde{e}^T \tilde{e} = \tilde{x}^T H^T H \tilde{x}$, it amounts to using the PI in Table 5.5-1 with $Q = 0$, $R = rI$, and $P = H^T H$.

However, if (H, A) is not observable and if $k = 0$, there may be problems with any LQ design [Lewis, 1986]. Specifically, in this case the Lyapunov equation

$$A_c^T P + P A_c + H^T H + C^T K^T R K C = 0 \quad (5.5-20)$$

may not have a positive-definite solution P . This could result in some of the feedback gains being set to zero in the LQ optimal solution.

TABLE 5.5-1. LQ Tracker with Time-Weighted PI

System Model

$$\begin{aligned}\dot{x} &= Ax + Bu + Gr \\ y &= Cx + Fr\end{aligned}$$

Control

$$u = -Ky$$

Performance Index

$$J = \frac{1}{2} \int_0^{\infty} [t^k \bar{x}^T P \bar{x} + \bar{x}^T (Q + C^T K^T R K C) \bar{x}] dt$$

Optimal Output Feedback Control

$$\begin{aligned}0 &= g_0 \equiv A_c^T P_0 + P_0 A_c + P \\ 0 &= g_1 \equiv A_c^T P_1 + P_1 A_c + P_0 \\ &\vdots \\ 0 &= g_{k-1} \equiv A_c^T P_{k-1} + P_{k-1} A_c + P_{k-2} \\ 0 &= g_k \equiv A_c^T P_k + P_k A_c + k! P_{k-1} + Q + C^T K^T R K C\end{aligned}\tag{5.5-21}$$

$$\begin{aligned}0 &= A_c S_k + S_k A_c^T + X \\ 0 &= A_c S_{k-1} + S_{k-1} A_c^T + k! S_k \\ 0 &= A_c S_{k-2} + S_{k-2} A_c^T + S_{k-1} \\ &\vdots \\ 0 &= A_c S_0 + S_0 A_c^T + S_1\end{aligned}\tag{5.5-22}$$

$$0 = \frac{\partial \mathcal{H}}{\partial K} = R K C S_k C^T - B^T (P_0 S_0 + \cdots + P_k S_k) C^T + B^T A_c^{-T} P_k \bar{x} \bar{y}^T\tag{5.5-23}$$

with r a unit step of magnitude r_0 and

$$\bar{x} = -A_c^{-1} B_c r_0\tag{5.5-24}$$

$$\bar{y} = C \bar{x} + F r_0\tag{5.5-25}$$

$$X = \bar{x} \bar{x}^T = A_c^{-1} B_c r_0 r_0^T B_c^T A_c^{-T}\tag{5.5-26}$$

where

$$A_c = A - B K C, \quad B_c = G - B K F$$

Optimal Cost

$$J = \frac{1}{2} \text{tr}(P_k X)$$

To correct this, we could add a term like $\tilde{x}^T Q \tilde{x}$ in the PI, with (\sqrt{Q}, A) observable. This, however, is exactly what we are trying to avoid, since it will give us all of the elements of Q as design parameters that should be varied until a suitable K results. To avoid this counterintuitive approach, we need only select $k > 0$ in the PI in Table 5.5-1. To see why, consider the case $Q = 0$ and examine Table 5.5-1. Note that even if (\sqrt{P}, A) is not observable, $[(k!P_{k-1})^{1/2}, A]$ may be observable for some $k > 0$. If so, the last Lyapunov equation in (5.5-21) will have a positive definite solution P_k , which will correct the observability problem. That is, by using time weighting, the LQ observability problem is corrected. We shall illustrate this point in Example 5.5-2.

Example 5.5-1: Constrained Feedback Control for F-16 Lateral Dynamics. In Example 5.3-1 we showed how to design a lateral stability augmentation system for an F-16. The resulting gain matrix K had eight nonzero entries. It would be desirable to avoid gain scheduling such a large number of gains, as well as to avoid feedback from roll rate and bank angle to rudder, and from washed-out yaw-rate and sideslip to aileron. That is, the gain matrix should have the form

$$K = \begin{bmatrix} 0 & x & 0 & x \\ x & 0 & x & 0 \end{bmatrix}. \quad (1)$$

This *constrained output feedback* is quite easy to achieve using the techniques just discussed. Indeed, select a PI of the form (5.5-5) with $g_{11} = 1000$, $g_{13} = 1000$, $g_{22} = 1000$, $g_{24} = 1000$ in order to weight the unwanted entries of $K = [k_{ij}]$. Then the algorithm of Table 5.3-1, with the modified equation (5.5-6) used to evaluate the PI in a numerical minimization scheme, yields the feedback gain matrix

$$K = \begin{bmatrix} -1E-3 & -0.55 & 1E-3 & -0.49 \\ -1.14 & -1E-3 & 0.05 & 1E-3 \end{bmatrix} \approx \begin{bmatrix} 0 & -0.55 & 0 & -0.49 \\ -1.14 & 0 & 0.05 & 0 \end{bmatrix}. \quad (2)$$

The same Q and R were used as in Example 5.3-1. The resulting closed-loop poles are

$$\begin{aligned} s = & -1.16 \pm j0.99 && \text{dutch roll mode } (r, \beta) \\ & -0.79 && \text{spiral mode} \\ & -7.42 && \text{roll subsidence mode} \\ & -11.54 \pm j19.51, -12.27. && \end{aligned} \quad (3)$$

Note that the spiral and roll subsidence modes now consist of two real poles so that the complex roll mode is absent. The closed-loop response is shown in Fig. 5.5-1. It should be compared to the response obtained in Example 5.3-1 as well as Example 4.4-3.

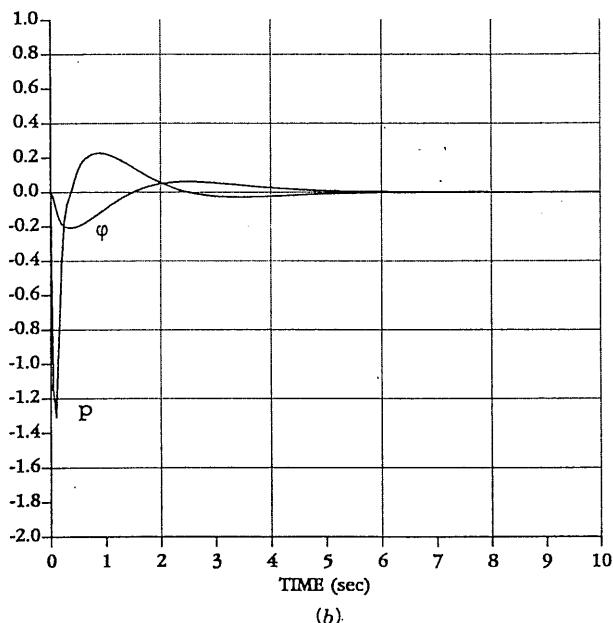
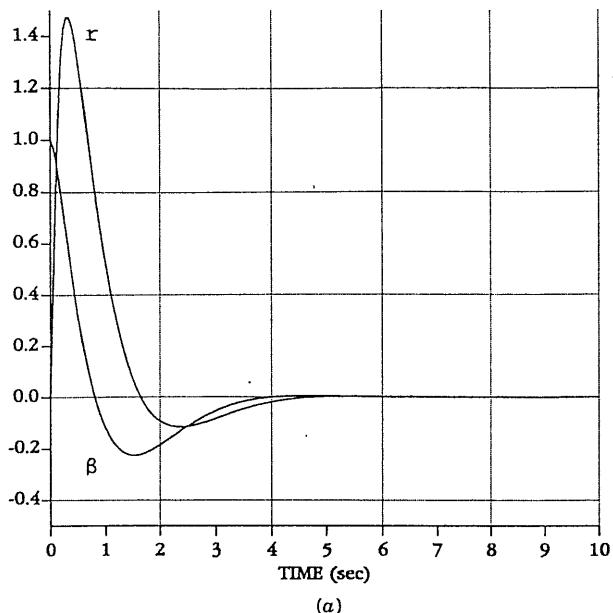


Fig. 5.5-1 Closed-loop lateral response. (a) dutch roll states β and r , (b) spiral and roll subsidence states ϕ and p .

An alternative design technique is simply to use the option in program LQ of instructing the program to leave certain elements of K fixed at zero during the minimization procedure. ■

Example 5.5-2: Time-Dependent Weighting Design of Normal Acceleration CAS. In Example 5.4-1 we designed a normal acceleration CAS. A deficiency with that approach was the need to check for the observability of (\sqrt{Q}, A) ; there, unobservability led us to weight the integrator output in Q . In this example we show how to avoid the observability issue by using time-dependent weighting in the PI.

The aircraft and controller dynamics are the same as in Example 5.4-1. Here, however, we shall select the time-weighted PI

$$J = \frac{1}{2} \int_0^{\infty} (t^2 \tilde{e}^2 + \rho \tilde{u}^2) dt, \quad (1)$$

which is entirely sensible from a performance point of view and contains only one design parameter to be tuned. This corresponds to the PI in Table 5.5-1 with $P = H^T H$, $Q = 0$, $R = \rho$.

Selecting $\rho = 0.05$ and using program LQ we obtained the control gains

$$K = [-0.847 \quad -0.452 \quad 1.647 \quad 8.602], \quad (2)$$

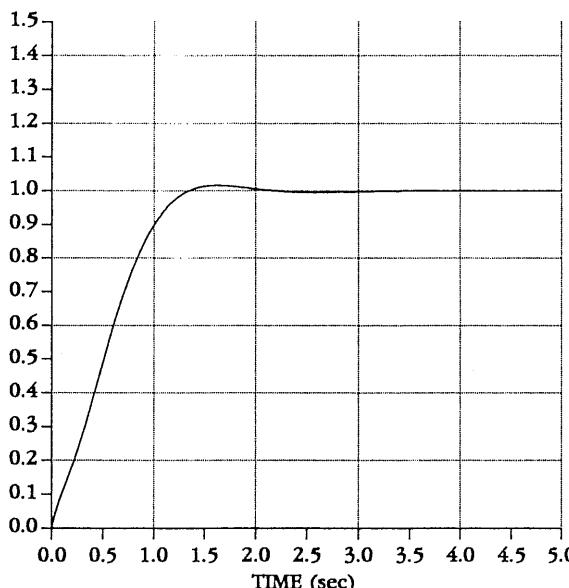


Fig. 5.5-2 Normal acceleration step response.

the closed-loop poles

$$\begin{aligned}s = & -1.90 \pm j2.58 \\& -2.35 \\& -13.88 \pm j3.12,\end{aligned}\quad (3)$$

and the step response shown in Fig. 5.5-2. It is much better than the result of Example 5.4-1, and was obtained without juggling the elements of the Q matrix or worrying about observability issues. By using time weighting in the PI, we have formulated a design problem that has only one design parameter that needs to be varied, namely the control weighting ρ . This entire design took 5 minutes. Contrast to Example 4.5-3. ■

Example 5.5-3: Pitch-Rate Control System Using LQ Design. In this example we reconsider pitch rate control system design using LQ techniques. The approach to be used here should be compared to the classical approach used in Example 4.5-1. It will be demonstrated how two of the PIs just developed can simplify the control system design, since they have *only one design parameter that must be tuned to obtain good performance*. This LQ technique is therefore in sharp contrast to the classical approach, where we had to vary all three elements of the gain matrix in successive loop-closure design. It is also in contrast to the traditional modern LQ approaches, where all the elements of the PI weighting matrices must generally be tuned to obtain good performance and where the observability properties of the PI must be considered in selecting the state weighting matrix.

Since we are using a modern LQ-based approach, a sensible formulation of the problem should result in closed-loop stability for all selections of the design parameter. This is an extremely important property of modern control design techniques and in complete contrast to classical techniques where stability in multiloop systems can be difficult to achieve.

a. *Aircraft and Control System Dynamics.* The pitch control system is shown in Fig. 5.5-3, where the control input is elevator actuator voltage $u(t)$ and r is a reference step input corresponding to the desired pitch command. Thus the performance output $z(t)$ is the pitch rate q . The measured outputs $y(t)$ are pitch q and angle of attack α ; however, since α measurements are quite noisy, a low-pass filter with a cutoff frequency of 10 rad/s is used to provide filtered measurements α_F of the angle of attack. To ensure zero steady-state error an integrator was added in the feedforward channel; this corresponds to the compensator dynamics. The integrator output is ϵ .

We used the short-period approximation to the F-16 dynamics linearized about the nominal flight condition in Table 3.4-3 (502 ft/s, 0 ft altitude, level flight, with the cg at $0.35\bar{c}$). Thus the basic aircraft states of interest are α and q . An additional state is introduced by the elevator actuator. The elevator deflection is δ_e .

The states and outputs of the plant plus compensator are

$$x = \begin{bmatrix} \alpha \\ q \\ \delta_e \\ \alpha_F \\ \epsilon \end{bmatrix}, \quad y = \begin{bmatrix} \alpha_F \\ q \\ \epsilon \end{bmatrix} \quad (1)$$

and the system dynamics are described by

$$\dot{x} = Ax + Bu + Gr \quad (2)$$

$$y = Cx + Fr \quad (3)$$

$$z = Hx \quad (4)$$

with

$$A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 & 0 & 0 \\ 0.82225 & -1.07741 & -0.17555 & 0 & 0 \\ 0 & 0 & -20.2 & 0 & 0 \\ 10.0 & 0 & 0 & -10 & 0 \\ 0 & -57.2958 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 20.2 \\ 0 \\ 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 57.2958 & 0 \\ 0 & 57.2958 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$$H = [0 \ 57.2958 \ 0 \ 0 \ 0].$$

The factor of 57.2958 is added to convert angles from radians to degrees.

The control input is

$$u = -Ky = -[k_\alpha \ k_q \ k_I]y = -k_\alpha\alpha_F - k_qq - k_I\epsilon. \quad (5)$$

It is required to select the feedback gains to yield good closed-loop response to a step input at r , which corresponds to a single-input/multi-output design problem.

Now consider two LQ designs based on two different performance indices. The modified PIs introduced in this section will mean that we do not need to worry about observability issues and that *only one design parameter will appear*. This is significant in view of the fact that there are five states and three control gains to find.

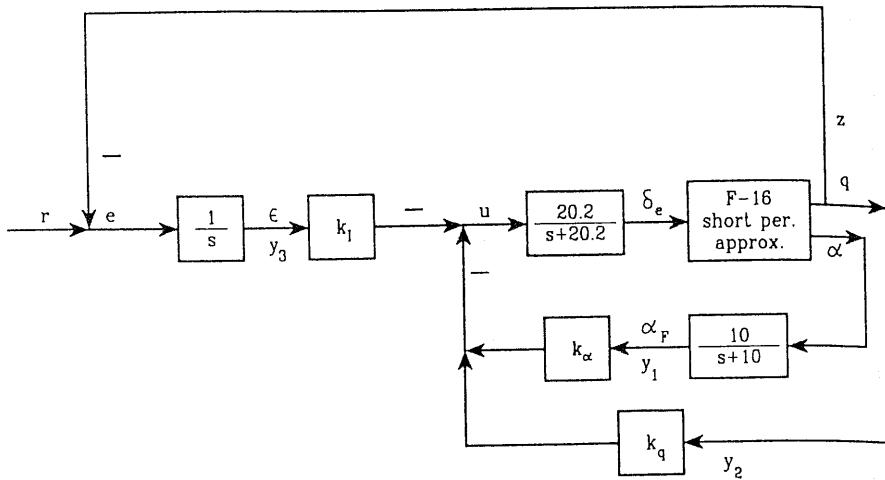


Fig. 5.5-3 Pitch-rate control system.

Since the integrator makes the system Type I, the steady-state error \bar{e} is equal to zero and

$$e(t) = \bar{e}(t). \quad (6)$$

Thus the PI term involving \bar{e} in Section 5.4 is not required.

b. Time-Dependent Weighting Design. Consider the PI

$$J = \frac{1}{2} \int_0^{\infty} (t^2 \bar{e}^2 + \rho \ddot{u}^2) dt. \quad (7)$$

This is a natural PI that corresponds to the actual performance requirements of keeping the tracking error small without using too much control energy, and also has the important advantage of requiring the adjustment of only one design parameter ρ . It amounts to using $P = H^T H$, $Q = 0$, $R = \rho$ in Table 5.5-1.

Program LQ was used to solve the design equations in Table 5.5-1 for several values of ρ . A good step response was found with $\rho = 1$, which yielded optimal gains of

$$K = [-0.046 \quad -1.072 \quad 3.381] \quad (8)$$

closed-loop poles of $s = -8.67 \pm j9.72$, -9.85 , -4.07 , and -1.04 , and the step response in Fig. 5.5-4. Compare to the results of Example 4.5-1.

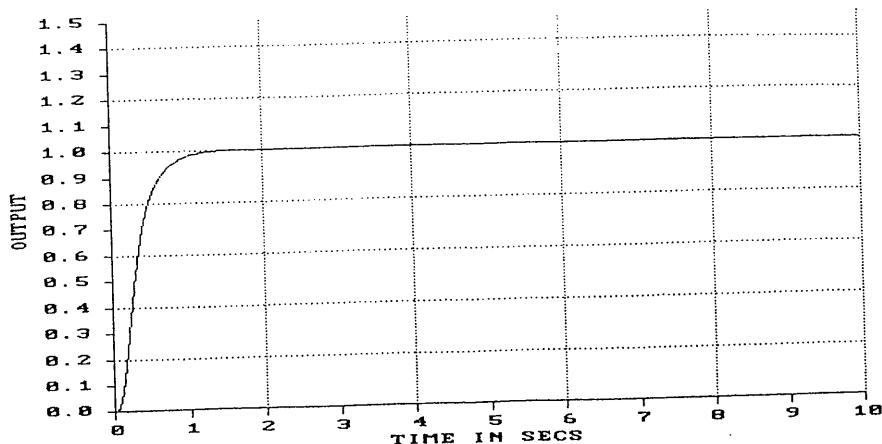


Fig. 5.5-4 Pitch-rate step response using time-dependent weighting design.

c. *Derivative Weighting Design.* Since elevator actuator *rate* has a stronger intuitive connection to "control activity" than does elevator displacement, let us illustrate derivative weighting by repeating the design. Select the PI

$$J = \frac{1}{2} \int_0^\infty (pt^2 e^2 + \dot{\delta}_e^2) dt. \quad (9)$$

Since $e(t) = \dot{\epsilon}(t)$, this may be written

$$J = \frac{1}{2} \int_0^\infty (pt^2 \dot{\epsilon}^2 + \dot{\delta}_e^2) dt, \quad (10)$$

with $\epsilon(t)$ and $\delta_e(t)$ the deviations in the integrator output and elevator deflection. This is exactly the derivative weighting PI (5.5-16) with $P = \text{diag}\{0, 0, 0, 0, p\}$ and $Q = \text{diag}\{0, 0, 1, 0, 0\}$.

It should be emphasized that we have again been careful to formulate the problem in such a way that only one design parameter, namely p , needs to be adjusted in the iterative design phase.

The software described in Appendix B was used to minimize (5.5-17) subject to (5.5-13) for several values of p . The weight $p = 10$ led to a good step response, as shown in Fig. 5.5-5. The feedback gain matrix was

$$K = [-0.0807 \quad -0.475 \quad 1.361] \quad (11)$$

and the closed-loop poles were at $s = -3.26 \pm j2.83, -1.02, -10.67$, and -14.09 . These poles are virtually identical to those obtained in Example 4.5-1. Compare the design process in this example with the design process in that example. ■

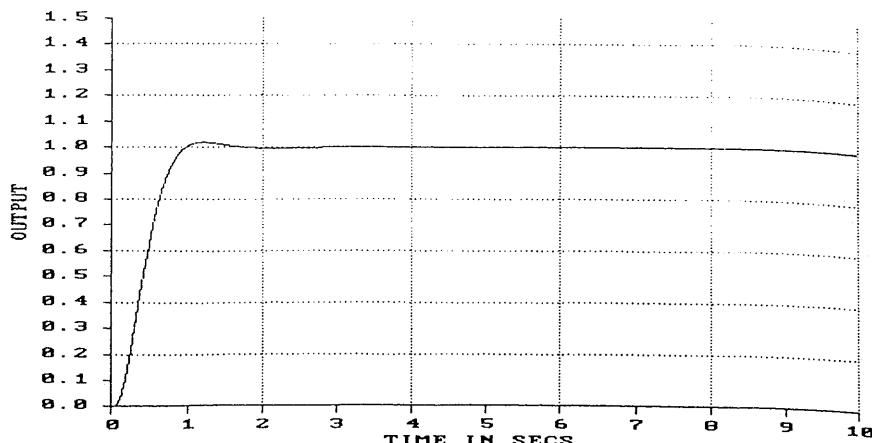


Fig. 5.5-5 Pitch-rate step response using derivative weighting design.

Example 5.5-4: Multivariable Wing Leveler. In this example we shall illustrate a multi-input/multi-output (MIMO) design using the LQ approach developed in this chapter. This example should be compared with Example 4.4-3, where we designed a two-input/two-output roll damper/yaw damper using classical control by successive loop closures. It may also be compared with Examples 4.5-4 and 4.6-7.

a. *Control System Structure.* The control system shown in Fig. 5.5-6 is meant to hold the aircraft's wings level while providing yaw damping by holding washed-out yaw rate r_w at zero. It is a two-channel system. In the upper channel there is an outer-loop unity-gain feedback of bank angle ϕ with an inner-loop feedback of roll rate p . This channel has a PI compensator to make the system Type I to achieve zero steady-state bank angle error. The control input for the upper channel is aileron deflection δ_a . The lower channel has a feedback of washed-out yaw rate r_w ; in this channel the control input is rudder deflection δ_r .

The reference command is $r_c = [r_\phi \quad r_r]^T$. The tracking control system should hold ϕ at the commanded value of r_ϕ , and r_w at the commanded value of r_r , which is equal to zero. To hold the wings level, r_ϕ is set equal to zero, although it could be any commanded bank angle. The tracking error is $e = [e_\phi \quad e_r]^T$ with

$$\begin{aligned} e_\phi &= r_\phi - \phi \\ e_r &= r_r - r_w. \end{aligned} \tag{1}$$

b. *State Equations for Aircraft and Control Dynamics.* As in Example 5.3-1, we used the F-16 linearized lateral dynamics at the nominal flight condition

in Table 3.4-3 ($V_T = 502$ ft/s, 300 psf dynamic pressure, cg at $0.35\bar{c}$) retaining the lateral states sideslip β , bank angle ϕ , roll rate p , and yaw rate r . Additional states δ_a and δ_r are introduced by the aileron and rudder actuators. The washout filter state is called x_w . We denote by ϵ the output of the controller integrator in the upper channel. Thus the entire state vector is

$$x = [\beta \quad \phi \quad p \quad r \quad \delta_a \quad \delta_r \quad x_w \quad \epsilon]^T. \quad (2)$$

The full state-variable model of the aircraft plus actuators, washout filter, and control dynamics is of the form

$$\dot{x} = Ax + Bu + Gr_c \quad (3)$$

with

$$A = \begin{bmatrix} -0.3220 & 0.0640 & 0.0364 & -0.9917 & 0.0003 & 0.0008 & 0 & 0 \\ 0 & 0 & 1 & 0.0037 & 0 & 0 & 0 & 0 \\ -30.6492 & 0 & -3.6784 & 0.6646 & -0.7333 & 0.1315 & 0 & 0 \\ 8.5395 & 0 & -0.0254 & -0.4764 & -0.0319 & -0.0620 & 0 & 0 \\ 0 & 0 & 0 & 0 & -20.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20.2 & 0 & 0 \\ 0 & 0 & 0 & 57.2958 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20.2 & 0 \\ 0 & 20.2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}. \quad (4)$$

The performance output that should follow the reference input $[r_\phi \quad r_r]^T$ is

$$z = \begin{bmatrix} \phi \\ r_w \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 57.2958 & 0 & 0 & -1 & 0 \end{bmatrix} x = Hx, \quad (5)$$

where the factor 57.2958 converts radians to degrees. According to the figure, if we define the measured output as

$$y = \begin{bmatrix} \epsilon \\ e_r \\ p \\ e_\phi \end{bmatrix} = Cx + Fr_c \quad (6)$$

with

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -57.2958 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$F = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad (7)$$

the control input $u = [u_a \ u_r]^T$ may be expressed as

$$u = -Ky \quad (8)$$

with

$$K = \begin{bmatrix} k_1 & 0 & k_3 & k_4 \\ 0 & k_2 & 0 & 0 \end{bmatrix}. \quad (9)$$

The control gains k_i must now be determined for satisfactory closed-loop response. Therefore, this is an output-feedback design problem exactly of the form addressed in this chapter. Note that some of the entries of K must be constrained to zero to yield the desired control structure shown in Fig. 5.5-6.

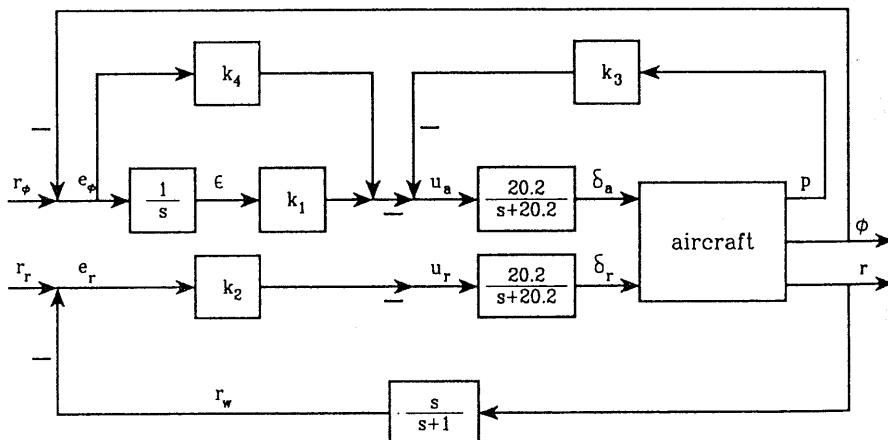


Fig. 5.5-6 Wing-leveler lateral control system.

c. *LQ Output Feedback Design.* To guarantee tracking by $z(t)$ of the reference command $r_c(t)$, we may select the PI

$$J = \frac{1}{2} \int (t^2 \tilde{x}^T P \tilde{x} + \tilde{u}^T \tilde{u}) dt + \frac{1}{2} v \bar{e}^T \bar{e} \quad (10)$$

with $\tilde{x}(t)$ and $\tilde{u}(t)$ the state and control deviations defined in Section 5.4 and \bar{e} the steady-state error. Although the integrator in the upper control channel guarantees that \bar{e}_ϕ will be zero, the steady-state error weighting v is required to ensure that \bar{e}_r is small. Note that v is a scalar.

The design equations for K using this PI are given in Table 5.5-1, with, however, the extra terms from Table 5.4-1 added to (5.5-23) due to the steady-state error weighting v . Thus K is easily determined using program LQ.

Several attempts were made to obtain suitable closed-loop behavior using different values for v and P . Finally, it was found that good behavior was obtained with $v = 10$ and P selected to weight the states β , ϕ , p , r , and ϵ , as well as the cross-term in ϕr . That is,

$$p_{11} = p_{22} = p_{33} = p_{44} = p_{88} = 100, \quad p_{24} = p_{42} = 10. \quad (11)$$

The motivation for the p_{24} cross-weighting is that after a few design attempts with different P , it was found that there were always several barely stable and badly damped complex pole pairs in the closed-loop system. The p_{24} , p_{42} cross-weighting penalizes the dutch roll mode, which was one of the ones yielding problems. The motivation for selecting p_{88} weighting is that good results are generally obtained if the integrator output is weighted.

Using the final selection of v and P , the control gains were found to be

$$\begin{aligned} k_1 &= 15.04, & k_2 &= 0.1822, \\ k_3 &= -5.348, & k_4 &= 22.52, \end{aligned} \quad (12)$$

yielding closed-loop poles of

$$\begin{aligned} &-0.72 \pm j3.03 \\ &-1.12 \pm j0.07 \\ &-2.43, -5.05 \\ &-15.3, -19.4. \end{aligned} \quad (13)$$

d. *Simulation.* The closed-loop response to a reference command of $r_\phi = 1$, $r_r = 0$ is shown in Fig. 5.5-7. The transient response and steady-state errors

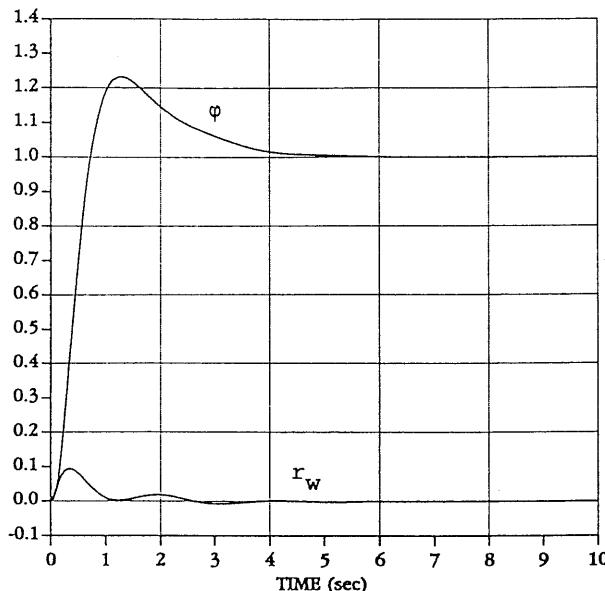


Fig. 5.5-7 Closed-loop response to a command of $r_\phi = 1$, $r_r = 0$. Bank angle ϕ (rad) and washed-out yaw rate (rad/s).

are both quite satisfactory. This is despite the presence of an underdamped pole pair at $-0.72 \pm j3.03$. One should recall the discussion in Example 4.4-3, where the strong coupling between the aircraft roll and yaw channels was emphasized. Despite this, Fig. 5.5-7 shows that we have been quite successful in decoupling the yaw rate from the bank angle. ■

Example 5.5-5: Glide-Slope Coupler. A glide-slope coupler is part of an automatic landing system—it guides an aircraft down a predetermined flight path to the end of a runway. At the end of the descent another control system, the automatic flare control (Example 5.6-1) is switched in to cause the aircraft to flare to a landing.

In this example we design a glide-slope coupler for the longitudinal dynamics of a medium-sized transport aircraft. Our approach should be compared to the frequency-domain approach in Example 4.6-4. See also [Blakelock, 1965].

a. Aircraft Dynamics. The important inputs are both elevator and throttle for this problem, since both are needed to fly down a glide path in a coordinated manner. Exactly as in Example 4.6-4, the longitudinal dynamics of the aircraft were linearized about a velocity of $V_T = 250$ ft/s with the cg at $0.25\bar{c}$ and including throttle and elevator actuators. The state and control

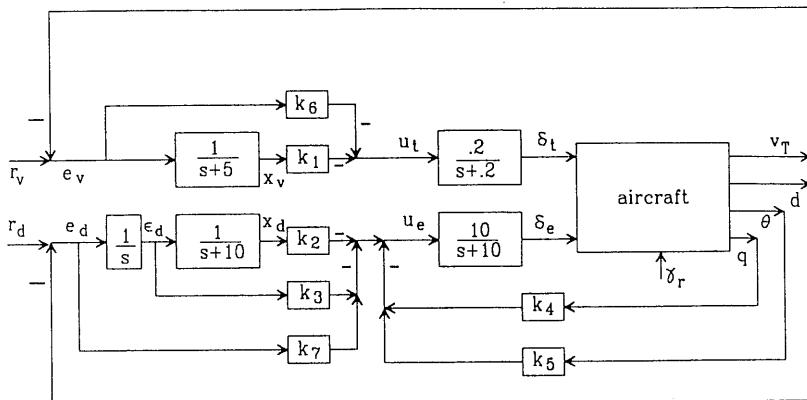


Fig. 5.5-8 Glide-slope coupler.

input are

$$x = [v_T \quad \alpha \quad \theta \quad q \quad \delta_t \quad \delta_e]^T, \quad u = [u_t \quad u_e]^T, \quad (1)$$

with v_T the deviation from trim velocity. The dynamics are described by

$$\dot{x} = Ax + Bu, \quad (2)$$

where A and B may be found by referring to Example 4.6-4. [In finding the A and B in (2) from the matrices in Example 4.6-4, note our selection of states.]

At this point it is worthwhile to examine Fig. 5.5-8, which we are starting to construct.

b. Glide Slope Geometry. The glide-slope geometry is discussed in Example 4.6-4. The commanded or reference flight path angle $-\gamma_r$ is generally 2.5° . The perpendicular distance from the glide path is $d(t)$.

Our control objectives in the glide-slope coupler are to regulate to zero the off-glide-path distance d and the deviation v_T from trim velocity. Then the aircraft will remain on the glide path with the nominal velocity of $V_T = 250$ ft/s. To accomplish this, the two control inputs are throttle and elevator. The outputs available for feedback are pitch rate q , pitch angle θ , v_T , and d , which is available from measurements taken from the ground.

The component of velocity perpendicular to the glide path is given by

$$d = V_T \sin(\gamma - \gamma_r) \approx V_T(\gamma - \gamma_r) \quad (3)$$

when $(\gamma - \gamma_r)$ is small. We shall assume that the velocity deviation v_T is small, and take V_T in (3) as the trim velocity 250 ft/s. To follow the glide

path, we require $\dot{d} = 0$, so that the flight path angle γ should be equal to γ_r . Then the aircraft will descend at an angle of $\gamma_r = -2.5^\circ$.

In terms of variables in the state vector in (1), we may use $\gamma = \theta - \alpha$ to write

$$\dot{d} = V_T \theta - V_T \alpha - \frac{V_T}{57.2958} \gamma_r = V_T \theta - V_T \alpha - 4.3633 \gamma_r, \quad (4)$$

with θ and α in radians and γ_r in degrees. Therefore, we may include the off-glide-path distance d as a state in (1) by redefining

$$x = [v_T \quad \alpha \quad \theta \quad q \quad d \quad \delta_t \quad \delta_e]^T. \quad (5)$$

c. *Control System Structure.* Our objective is to regulate v_T and d to zero. Thus we may define the performance output as

$$z = \begin{bmatrix} v_T \\ d \end{bmatrix} = Hx. \quad (6)$$

Now examine Fig. 5.5-8, which we have drawn to show that this may be considered as a tracking problem with reference commands r_v and r_d of zero. The tracking error is $e = [e_v \quad e_d]^T$ with

$$\begin{aligned} e_v &= r_v - v_T \\ e_d &= r_d - d. \end{aligned} \quad (7)$$

To obtain zero steady-state error in $v_T(t)$ and $d(t)$, we could add integrators in each of the forward error paths. However, according to the open-loop dynamics in Example 4.6-4 there are already several poles near the origin. Adding more poles near the origin makes the problem of stabilization more difficult.

Since we are more concerned about keeping d exactly zero, let us only add an integrator in the forward path corresponding to the tracking error in d . We can then obtain a small enough error in v_T without a forward-path integrator by using weighting of the steady-state error, as we shall soon see.

An additional consideration for including a forward-path integrator in the d channel is the following. Note from (4) and Fig. 5.5-8 that the commanded glide-path angle γ_r acts as a constant disturbance of magnitude -2.5° into the system. The disturbance affects \dot{d} . To reject this constant disturbance, we need a Type I system with respect to d , which requires the integrator in the d feedforward path.

We can gain considerable insight by having root-locus design techniques in mind during a design by modern control. Thus to pull the closed-loop poles into the left-half plane, we may add compensator zeros in the left-half plane. To implement the compensators without pure differentiation, we should add poles relatively far in the left-half plane, where they will not appreciably affect the root locus. Thus let us propose a lead compensator in each forward channel (see Fig. 5.5-8).

The compensators we propose are of the form

$$\frac{w_v}{e_v} = \frac{k_1}{s+5} + k_6 = k_6 \frac{s + (5 + k_1/k_6)}{s+5} \quad (8)$$

$$u_t = -w_v$$

and

$$\frac{w_d}{e_d} = \frac{k_2}{s(s+10)} + \frac{k_3}{s} + k_7 \quad (9)$$

$$= k_7 \frac{s^2 + (10 + k_3/k_7)s + (k_2 + 10k_3)/k_7}{s(s+10)}$$

$$u_e = -w_d.$$

The important point to note is that by varying the control gains, we may adjust *both the compensator gain and its zeros*. Thus the *LQ optimization routine can adjust the zeros of the compensators*, presumably inducing lead compensation where it is required. We have selected the throttle compensator pole at $s = -5$ and the distance compensator pole at $s = -10$; however, any poles far to the left compared to the aircraft poles would suffice.

As we have seen in Example 4.6-4, selecting multiple control gains by classical techniques requires a successive-loop-closure approach. We hope to show that finding suitable gains using modern control theory is far easier, given a sensible problem formulation.

To formulate the controller so that the gains may be determined by our output-feedback LQ approach, note that state variable representations of (8) and (9) are given by

$$\dot{x}_v = -5x_v + e_v = -5x_v - v_T + r_v \quad (10)$$

$$u_t = -k_1x_v - k_6e_v = -k_1x_v - k_6(-v_T + r_v) \quad (11)$$

and

$$\dot{\epsilon}_d = e_d = -d + r_d \quad (12)$$

$$\dot{x}_d = -10x_d + \epsilon_d \quad (13)$$

$$u_e = -k_2x_d - k_3\epsilon_d - k_7e_d = -k_2x_d - k_3\epsilon_d - k_7(-d + r_d). \quad (14)$$

The dynamical equations (4), (10), (12), and (13) may be incorporated into the system description by defining the augmented state

$$x = [v_T \quad \alpha \quad \theta \quad q \quad d \quad \delta_t \quad \delta_e \quad x_v \quad x_d \quad \epsilon_d]^T. \quad (15)$$

Then the augmented system is described by

$$\dot{x} = Ax + Bu + Gr \quad (16)$$

with

$$\begin{aligned}
 A &= \begin{bmatrix} -0.04 & 19.0096 & -32.1689 & 0 & 0 & 10.1 & 0 & 0 & 0 \\ -0.001 & -0.64627 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.7739 & 0 & -0.529765 & 0 & 0.02463 & -0.011 & 0 & 0 \\ 0 & -250 & 250 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \\
 B &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -4.3633 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}
 \end{aligned}
 \tag{17}$$

To incorporate the constant disturbance γ_r required in (4), we have defined an augmented input

$$u' = [u^T \ \ \gamma_r]^T = [u_t \ \ u_e \ \ \gamma_r]^T. \quad (18)$$

Inputs such as γ_r , which are not actual controls, nor reference signals $r(t)$ in the usual tracking system sense, are called *exogenous inputs*. Although they play the role of disturbances in the system, they are crucial in guaranteeing the desired system behavior. Indeed, were we to ignore γ_r , the glide-slope coupler would always make the aircraft fly a horizontal path!

It should be clearly understood that for the design of the control system, only the control input $u(t)$ is used. The full input $u'(t)$ will be required only in the simulation stage, where γ_r will be set equal to -2.5° to obtain the desired landing approach behavior.

In (16)/(17) the reference input is defined as

$$r = [r_v \ \ r_d]^T, \quad (19)$$

which is zero for the glide-slope coupler.

The equations (11) and (14) may be incorporated by defining a new measured output as

$$y = [x_v \ \ x_d \ \ \epsilon_d \ \ q \ \ \theta \ \ e_v \ \ e_d]^T. \quad (20)$$

Then

$$y = Cx + Fr \quad (21)$$

with

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 57.2958 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 57.2958 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$F = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (22)$$

Now, according to Fig. 5.5-8, the control vector $u(t)$ is given by the output feedback

$$u = \begin{bmatrix} u_t \\ u_e \end{bmatrix} = - \begin{bmatrix} k_1 & 0 & 0 & 0 & 0 & k_6 & 0 \\ 0 & k_2 & k_3 & k_4 & k_5 & 0 & k_7 \end{bmatrix} y = -Ky, \quad (23)$$

which has some elements constrained to zero.

According to (6), we may write

$$z = \begin{bmatrix} v_T \\ d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x = Hx. \quad (24)$$

At this point we have succeeded in casting the glide slope coupler design problem into the formulation required in Tables 5.4-1 and 5.5-1.

It is important to understand the construction of the matrices in (17), (22), and (24), for this problem formulation stage is one of the most important phases in the LQ design technique.

d. PI and Controls Design. The other important phase in LQ design is the selection of an appropriate PI. Since the loop gain around the velocity loop is not of Type I, we will require weighting of the steady-state error to force v_T to go to zero at steady state. Thus let us propose the PI

$$J = \frac{1}{2} \int_0^{\infty} (q t^2 \tilde{e}^T \tilde{e} + \tilde{u}^T \tilde{u}) dt + \frac{1}{2} v \tilde{e}^T \tilde{e}. \quad (25)$$

The motivation for the weighting t^2 follows. Weighting \tilde{e} in the PI makes practical sense since we want it to vanish. However, $\tilde{e}^T \tilde{e} = \tilde{x}^T H^T H \tilde{x}$, and (H, A) is not observable. In fact, the compensator states are not observable through $z = Hx$. An LQ design without the weighting t^2 would, therefore, fail. To correct the situation, we could weight the entire state in the PI by using a term like $\tilde{x}^T Q \tilde{x}$. However, this would give us too many design parameters (i.e., the elements of Q) and lead to a counterintuitive situation.

We prefer to work with sensible PIs, and in this situation we want to retain the weighting of $\tilde{e}(t)$, which is the variable of direct concern to us. Therefore, we use t^2 weighting to correct the observability problem. See the discussion preceding Example 5.5-1.

With t^2 weighting, a large value of the scalar q will result in a closed-loop system that is too fast. After several design iterations, it was found that suitable values for the PI design parameters were $q = 0.001$, $v = 100$. We employed program LQ to solve for the optimal gain K using the design equations of Table 5.5-1, including the steady-state error weighting from Table 5.4-1. We selected the option of fixing seven of the gain elements to zero as required by (23).

With $q = 0.001$, $v = 100$, the optimal control gains were

$$K = \begin{bmatrix} 2.598 & 0 & 0 & 0 & 0 & -0.9927 & 0 \\ 0 & 583.7 & -58.33 & -2.054 & -1.375 & 0 & 6.1 \end{bmatrix} \quad (26)$$

and the closed-loop poles were at

$$\begin{aligned} & -0.27 \pm j1.01 \\ & -0.36 \pm j0.49 \\ & -0.37 \pm j0.09 \\ & -1.18, -4.78, -8.38, -10.08. \end{aligned} \quad (27)$$

Thus the slowest time constant is $1/0.27 \approx 4$ s.

e. Simulation and Discussion. A simulation of the glide-slope coupler appears in Fig. 5.5-9. The aircraft was initialized in level flight at 1500 ft. The glide-slope coupler was switched on as the aircraft crossed through the glide path.

For simulation purposes, we used the exogenous input $\gamma_r = -2.5^\circ$ (the desired glide-path angle) and reference commands of $r_v = 0$, $r_d = 0$. Altitude h was added as a state using the equation for vertical velocity

$$\dot{h} = V_T \sin \gamma \approx V_T(\theta - \alpha), \quad (28)$$

with V_T assumed to be the trim velocity of 250 ft/s.

According to the altitude plot in Fig. 5.5-9a, after a small transient lasting about 20 s, the aircraft settles down onto the glide path and follows it down. Touchdown occurred at 137.5 s. Figure 5.5-9b shows the off-glide-path error d .

Figure 5.5-9c shows angle of attack and pitch angle. Note that after the transient, the flight-path angle is given by $\gamma = \theta - \alpha = -2.5^\circ$. Since in the descending configuration the aircraft is no longer at the original trim condition, a small angle of attack α of -0.18° remains at steady state. The final pitch angle θ is -2.68° .

According to Fig. 5.5-9d, the velocity deviation v_T settles out at 0.29 ft/s. This is a consequence of the fact that there is no integrator in the forward e_v path in Fig. 5.5-8. Thus the steady-state velocity on the glide path is $V_T = 250.29$ ft/s; this is very suitable from a performance point of view. The smallness of the steady-state deviation despite the fact that the v_T loop is of Type 0 is a consequence of the steady-state error weighting v in the PI (25).

Finally, the elevator and throttle control efforts δe and δt are shown in Fig. 5.5-9e. Note the coordinated control achieved in this two-input system using the LQ approach. Since the descent down the glide path does not represent the original trim condition, the steady-state values of the control

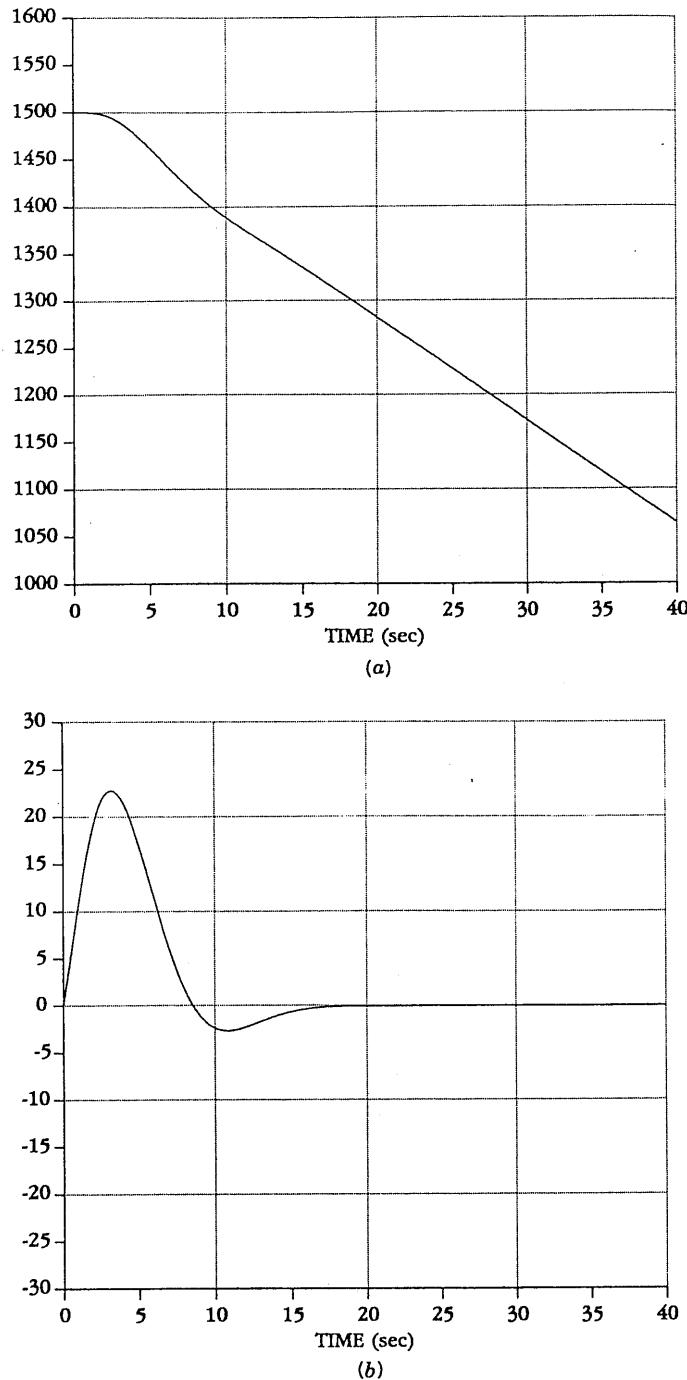
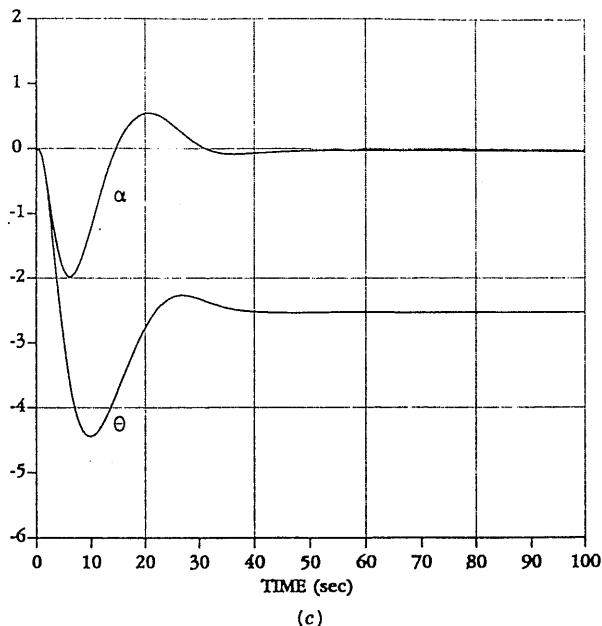
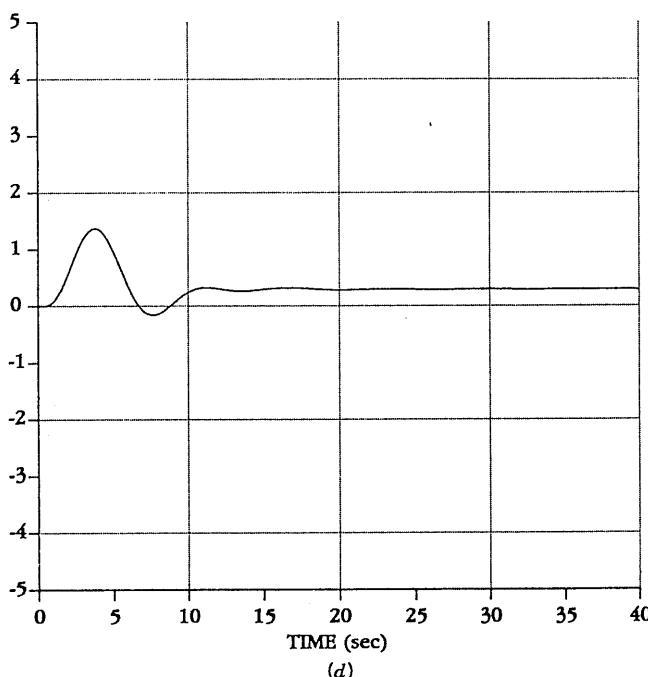


Fig. 5.5-9 Glide-slope coupler responses: (a) altitude h (ft); (b) off-glide path distance d (ft).



(c)



(d)

Fig. 5.5-9 (Continued) (c) Angle of attack α and pitch angle θ (deg); (d) velocity deviation v_T (ft/s).

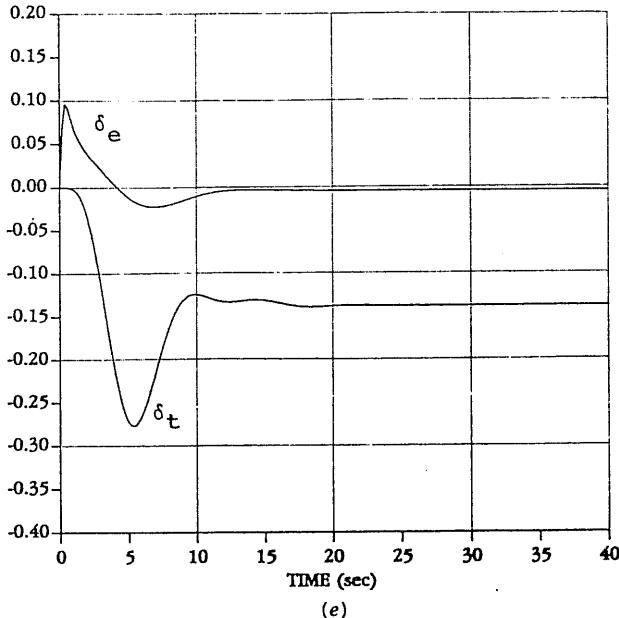


Fig. 5.5-9 (Continued) (e) Control efforts δ_e (rad) and δ_t (per unit).

efforts are not zero. Intuitively, less throttle is required to maintain 250 ft/s if the aircraft is descending.

Figure 5.5-9 shows that as the aircraft passes through the glide path, the elevator is pushed forward and the throttle is cut. As a result, the angle of attack and pitch angle decrease. After a slight positive position error d and an initial increase in velocity v_T , further control effort stabilizes the aircraft on the glide path.

It is interesting to note the fundamental mechanism behind the glide slope coupler. Namely, we regulate d in (3)-(4) to zero so that $\gamma = \gamma_r = -2.5^\circ$. Then, according to (28), $\dot{h} = V_T \sin \gamma_r$, the appropriate descent rate to stay on the glide path.

With the optimal gains in (26), according to (8) the velocity channel compensator is

$$\frac{w_v}{e_v} = -0.9927 \frac{s + 2.38}{s + 5}, \quad (29)$$

which is a lead compensator as anticipated. The zeros in the d channel compensator could similarly be found. It is important to note that our formulation has resulted in the *compensator zeros being selected in an optimal fashion*. This is an improvement over root-locus design, where the zeros are

determined using the engineering judgment that actually only applies for single-input/single-output systems.

It should be mentioned that determining an initial stabilizing gain K_0 for program LQ is not easy. In this example, we used the root-locus techniques in Chapter 4 to find the initial gain. Other approaches were discussed earlier in the subsection entitled "Constrained Feedback Matrix." ■

5.6 MODEL-FOLLOWING DESIGN

In Section 4.3 we discussed flying qualities and gave the military flying-qualities specifications for the various aircraft modes. These desirable flying qualities could be viewed as constituting an *ideal model with good performance* which we would like to reproduce in the actual aircraft. For instance, to obtain good longitudinal performance we could select suitable short period and phugoid poles from the flying-qualities specifications tabulated in Section 4.3. Then we could determine a state-variable realization of an ideal model with this behavior (see [Stern and Henke, 1971].) Finally, we could design a control system to make the actual aircraft behave like this ideal model.

This approach to control system design is the powerful model-following design technique. In this section we show how to design controllers that make the aircraft behave like a desired model. We shall discuss two fundamentally different sorts of model-following control, "explicit" and "implicit," which result in controllers of different structure [Armstrong, 1980; Kreindler and Rothschild, 1976; O'Brien and Broussard, 1978].

Explicit Model-Following Control

Regulation with Model-Following Behavior. First, we shall consider the regulator problem, where the objective is to drive the plant state to zero. Then we shall treat the more difficult tracker or servo problem, where the plant is to follow a reference command with behavior like the prescribed model. Let the plant be described in state-variable form by

$$\dot{x} = Ax + Bu \quad (5.6-1)$$

$$y = Cx \quad (5.6-2)$$

$$z = Hx \quad (5.6-3)$$

with state $x(t) \in \mathbf{R}^n$ and control input $u(t) \in \mathbf{R}^m$. The measured output $y(t)$ is available for feedback purposes.

A model is prescribed with dynamics

$$\dot{\underline{x}} = \underline{A}\underline{x} \quad (5.6-4)$$

$$\underline{z} = \underline{H}\underline{x} \quad (5.6-5)$$

where the model matrix \underline{A} reflects a system with desirable handling qualities such as speed of response, overshoot, and so on. The model states suitable for feedback purposes are given by

$$\underline{y} = \underline{Cx}. \quad (5.6-6)$$

Model quantities will be denoted by underbars or the subscript m .

Notice that the model has no reference input, since we are considering the regulator problem here. That is, the plant should have the same unforced response as the model, which translates into suitable locations of the poles.

It is desired to select the plant control $u(t)$ so that the plant performance output $\underline{z}(t)$ matches the model output $\underline{z}(t)$, for then the plant will exhibit the desirable time response of the model. That is, we should like to minimize the *model mismatch error*

$$e = \underline{z} - z = \underline{Hx} - Hx. \quad (5.6-7)$$

To achieve this control objective, let us select the performance index

$$J = \frac{1}{2} \int_0^{\infty} (e^T Q e + u^T R u) dt, \quad (5.6-8)$$

with $Q > 0$ (to ensure that all components of the error vanish) and $R > 0$.

We can cast this model-matching problem into the form of the regulator problem whose solution appears in Table 5.3-1 as follows.

Define the augmented state $x' = [x^T \ \underline{x}^T]^T$ and the augmented system

$$\dot{x}' = \begin{bmatrix} A & 0 \\ 0 & \underline{A} \end{bmatrix} x' + \begin{bmatrix} B \\ 0 \end{bmatrix} u \equiv A' x' + B' u \quad (5.6-9)$$

$$y' = \begin{bmatrix} y \\ \underline{y} \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & \underline{C} \end{bmatrix} x' \equiv C' x', \quad (5.6-10)$$

so that

$$e = [-H \ \underline{H}] x' \equiv H' x'. \quad (5.6-11)$$

Then the PI (5.6-8) may be written

$$J = \frac{1}{2} \int_0^{\infty} ((x')^T Q' x' + u^T R u) dt, \quad (5.6-12)$$

with

$$Q' = \begin{bmatrix} H^T Q H & -H^T Q \underline{H} \\ -\underline{H}^T Q H & \underline{H}^T Q \underline{H} \end{bmatrix}. \quad (5.6-13)$$

At this point it is clear that the design equations of Table 5.3-1 can be used to select $u(t)$ if the primed quantities A' , B' , C' , Q' , are used there. The conditions for convergence of the algorithm in Table 5.3-2 require that (A', B', C') be output stabilizable and $(\sqrt{Q'}, A')$ be detectable. Since the model matrix $\underline{\mathbf{A}}$ is certainly stable, the block diagonal form of A' and C' shows that output stabilizability of the plant (A, B, C) is required. The second condition requires detectability of the plant (H, A) .

The form of the resulting output-feedback control law is quite interesting. Indeed, the optimal feedback is of the form

$$u = -K' y' \equiv -[K_p \quad K_m] y' = -K_p y - K_m \underline{y}. \quad (5.6-14)$$

Thus not only the plant output but also the *model* output is required. That is, the model acts as a *compensator* to drive the plant states to zero in such a fashion that the performance output $z(t)$ follows the model output $\underline{z}(t)$.

Tracking with Model-Following Behavior. Unfortunately, while the model-following regulator problem has a direct solution that is easy to obtain, the model-following *tracker* problem is not so easy. In this situation, we should like the plant (5.6-1)–(5.6-3) to behave like the model

$$\dot{\underline{x}} = \underline{\mathbf{A}}\underline{x} + \underline{\mathbf{B}}r \quad (5.6-15)$$

$$\underline{z} = \underline{\mathbf{H}}\underline{x}, \quad (5.6-16)$$

which is *driven by the reference input* $r(t)$. The approach above yields

$$\dot{x}' = \begin{bmatrix} A & 0 \\ 0 & \underline{\mathbf{A}} \end{bmatrix} x' + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ B \end{bmatrix} r \equiv A'x' + B'u + G'r \quad (5.6-17)$$

and thus the derivation in Section 5.3 results in a PI that contains a term in $r(t)$, for which the determination of the optimal feedback gains is not easy [Lewis, 1986].

A convenient technique for designing a practical tracker is the *command-generator tracker (CGT)* technique, where the tracking problem is converted into a regulator problem [Franklin et al., 1986]. In this approach, a generator system is assumed for the reference input. We shall apply it here.

Thus suppose that for some initial conditions the reference command $r(t)$ satisfies the differential equation

$$r^{(d)} + a_1 r^{(d-1)} + \cdots + a_d r = 0 \quad (5.6-18)$$

for a given degree d and set of coefficients a_i . Most command signals of interest satisfy such an equation. For instance, the unit step of magnitude r_0

satisfies

$$\dot{r} = 0 \quad (5.6-19)$$

with $r(0) = r_0$, while the ramp (velocity command) with slope v_0 satisfies

$$\ddot{r} = 0 \quad (5.6-20)$$

with $r(0) = 0$, $\dot{r}(0) = v_0$. We call (5.6-18) the *command generator system*.

Define the command generator characteristic polynomial as

$$\Delta(s) = s^d + a_1 s^{d-1} + \cdots + a_d. \quad (5.6-21)$$

Then denoting d/dt in the time domain by D , we may write

$$\Delta(D)r = 0. \quad (5.6-22)$$

Multiplying the augmented dynamics (5.6-17) by $\Delta(D)$ results in the modified system

$$\dot{\xi} = A'\xi + B'\mu, \quad (5.6-23)$$

where the modified state and control input are

$$\xi = \Delta(D)x' = (x')^{(d)} + a_1(x')^{(d-1)} + \cdots + a_dx' \quad (5.6-24)$$

$$\mu = \Delta(D)u = u^{(d)} + a_1u^{(d-1)} + \cdots + a_du. \quad (5.6-25)$$

The reason for these manipulations is that because of (5.6-22), the reference command $r(t)$ does not appear in (5.6-23). Let us partition ξ as

$$\xi = \begin{bmatrix} \xi_p \\ \xi_m \end{bmatrix}. \quad (5.6-26)$$

Applying $\Delta(D)$ to the model mismatch error (5.6-7) results in

$$\Delta(D)e = [-H \ H]\xi = H'\xi. \quad (5.6-27)$$

This may be expressed in terms of state variables using the observability canonical form [Kailath, 1980], which for scalar $e(t)$ and $d = 3$, for instance, is

$$\dot{\epsilon} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ -a_3 & -a_2 & -a_1 \end{bmatrix}\epsilon + \begin{bmatrix} 0 \\ H' \end{bmatrix}\xi \equiv F\epsilon + \begin{bmatrix} 0 \\ H' \end{bmatrix}\xi \quad (5.6-28)$$

$$e = [1 \ 0 \ 0]\epsilon, \quad (5.6-29)$$

where $\epsilon(t) = [e \quad \dot{e} \quad \cdots \quad e^{(d-1)}]^T$ is the vector of the error and its first $d - 1$ derivatives.

Collecting all the dynamics (5.6-23)–(5.6-28) into one system yields

$$\frac{d}{dt} \begin{bmatrix} \epsilon \\ \xi \end{bmatrix} = \left[\begin{array}{c|c} F & 0 \\ \hline H' & A' \\ 0 & \end{array} \right] \begin{bmatrix} \epsilon \\ \xi \end{bmatrix} + \begin{bmatrix} 0 \\ B' \end{bmatrix} \mu. \quad (5.6-30)$$

Let us now note what we have achieved. Using the command generator polynomial $\Delta(s)$, we have prefiltered the augmented state, control input, and error to obtain a system (5.6-30) *that is not driven by the reference input $r(t)$* . Using this system we may now perform an LQ regulator design, since if its state goes to zero, the tracking error $e(t)$ vanishes. That is, by performing a regulator design (using Table 5.3-1) for (5.6-30), we may design a *tracker* control system that causes the original plant to follow the reference command with performance like that of the ideal model.

For the regulator design, we shall take the outputs available for feedback in (5.6-30) as

$$\nu = \begin{bmatrix} I & 0 & 0 \\ 0 & C & 0 \\ 0 & 0 & C \end{bmatrix} \begin{bmatrix} \epsilon \\ \xi_p \\ \xi_m \end{bmatrix} \quad (5.6-31)$$

To achieve small error without using too much control energy, we may select the PI (5.6-8) [with $u(t)$ replaced by $\mu(i)$]. According to (5.6-29), the error is given in terms of the state of (5.6-30) by

$$e = h \begin{bmatrix} \epsilon \\ \xi \end{bmatrix}, \quad (5.6-32)$$

with $h = [1 \quad 0 \quad \cdots \quad 0]$ the first row of the identity matrix. Therefore, in the PI we should weight the state of (5.6-30) using

$$Q' = h^T Q h. \quad (5.6-33)$$

Since the observability canonical form is always observable, the augmented system (5.6-30) is detectable if the plant (H, A) and the model $(\underline{H}, \underline{A})$ are both detectable.

Now, by applying the equations of Table 5.3-1 to the system (5.6-30) with outputs (5.6-31) and PI weights Q' and R , we may compute the control gains

in the control law

$$\mu = -[K_\epsilon \ K_p \ K_m] \begin{bmatrix} \epsilon \\ C\xi_p \\ C\xi_m \end{bmatrix} \quad (5.6-34)$$

or

$$\Delta(s)u = -K_\epsilon\epsilon - K_pC\Delta(s)x - K_mC\Delta(s)\underline{x}. \quad (5.6-35)$$

To determine the optimal tracking control input $u(t)$ for the original system, write this as

$$\Delta(s)(u + K_p\underline{y} + K_m\underline{\underline{y}}) = -K_\epsilon\epsilon \equiv -[K_d \ \cdots \ K_2 \ K_1] \begin{bmatrix} e \\ \dot{e} \\ \vdots \\ e^{(d-1)} \end{bmatrix}. \quad (5.6-36)$$

Thus we obtain the transfer function

$$\frac{u + K_p\underline{y} + K_m\underline{\underline{y}}}{e} = -\frac{K_1s^{d-1} + \cdots + K_{d-1}s + K_d}{s^d + a_1s^{d-1} + \cdots + a_d}, \quad (5.6-37)$$

which may be implemented in reachability canonical form [Kailath 1980] to obtain the control structure shown in Fig. 5.6-1.

CGT Structure. The structure of this *model-following command generator tracker* (CGT) is very interesting. It consists of an output feedback K_p , a feedforward compensator that is nothing but the reference model with a gain of K_m , and an additional feedforward filter in the error channel that guarantees perfect tracking. Note that if $d = 1$ so that $r(t)$ is a unit step, the error filter is a PI controller. If $d = 2$ so that $r(t)$ is a ramp, the error filter consists of two integrators, resulting in a type 2 system that gives zero steady-state error. What this means is the CGT design *automatically adds the compensator of appropriate structure to guarantee that the system has the correct type for perfect tracking*.

It is extremely interesting to note that the augmented state description (5.6-30) is nothing but the state description of Fig. 5.6-1. It should be emphasized that this technique is extremely direct to apply. Indeed, given the prescribed model and the command generator polynomial $\Delta(s)$, the system (5.6-30)/(5.6-31) may be written down immediately, and the equations in Table 5.3-1 used to select the feedback gains.

A word on the command generator assumption (5.6-22) is in order. In point of fact, for aircraft applications $r(t)$ is usually the pilot's command

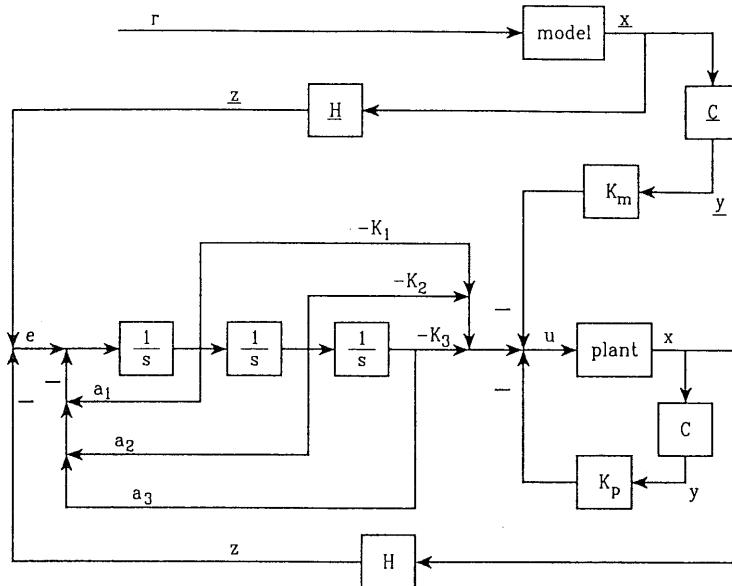


Fig. 5.6-1 Explicit model-following command generator tracker for $d = 3$.

input. For control systems design it is not necessary to determine the actual coefficients a_i that describe the pilot command, although this is one approach [Kreindler and Rothschild, 1976]. Instead, the performance objectives should be taken into account to select $\Delta(s)$. For instance, if it desired for the aircraft to follow a position command, we may select the command generator $\dot{r} = 0$. On the other hand, if the aircraft should follow a rate (velocity) command, we may select $\ddot{r} = 0$. Then when the actual command input $r(t)$ is applied (which may be neither a unit step nor a unit ramp), the aircraft will exhibit the appropriate closed-loop behavior.

Implicit Model-Following Control

We shall now discuss a formulation that results in a radically different sort of control scheme. In explicit model following, which is also called *model in the system control*, the model explicitly appeared in the controller as a feedforward compensator. On the other hand, implicit model following, also called *model in the performance index*, is a completely different approach in which the model does not appear in the control structure. Indeed, implicit model following can be viewed simply as a technique for selecting the weighting matrices in the PI in a meaningful way (see [Armstrong, 1980] and Kreindler and Rothschild, [1976]).

Suppose that the performance output $z(t)$ of the plant prescribed by (5.6-1)–(5.6-3) is required to follow the model given by

$$\dot{z} = \underline{A}z. \quad (5.6-38)$$

The model matrix \underline{A} has poles corresponding to desirable handling qualities of the plant, such as may be found in [Mil. Spec. 1797, 1987; Stern and Henke, 1971].

When the control objective is met, the performance output will satisfy the differential equation (5.6-38). Thus we may define an error by

$$e = \dot{z} - \underline{A}z. \quad (5.6-39)$$

This is a different sort of error than we have seen before.

To make $e(t)$ small without using too much control energy, we may choose $u(t)$ to minimize the PI (5.6-8). Since $\dot{z} = HAx + HBu$, this becomes

$$J = \frac{1}{2} \int_0^{\infty} \left[(HAx + HBu - \underline{A}Hx)^T Q (HAx + HBu - \underline{A}Hx) + u^T R u \right] dt. \quad (5.6-40)$$

or

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q' x + 2x^T W u + u^T R' u) dt, \quad (5.6-41)$$

where

$$Q' = (HA - \underline{A}H)^T Q (HA - \underline{A}H) \quad (5.6-42)$$

$$W = (HA - \underline{A}H)^T QHB, \quad R' = (B^T H^T QHB + R).$$

The additional term in W is a *cross-weighting* between $u(t)$ and $x(t)$.

In Table 5.3-1 we have given the LQ regulator design equations to determine the optimal output feedback gains for the case $W = 0$. By using techniques like those in that derivation (see the Problems), we may derive the modified design equations for the case of $W \neq 0$. They are

$$0 = A_c^T P + PA_c + Q + C^T K^T R K C - W K C - C^T K^T W^T \quad (5.6-43)$$

$$0 = A_c S + S A_c^T + X \quad (5.6-44)$$

$$0 = R K C S C^T - (P B + W)^T S C^T, \quad (5.6-45)$$

where

$$A_c = A - B K C. \quad (5.6-46)$$

The optimal cost is still given by

$$J = \frac{1}{2} \operatorname{tr}(P X). \quad (5.6-47)$$

To find the optimal output-feedback gains in

$$u = -K y \quad (5.6-48)$$

for implicit model following, it is only necessary to solve these design equations using Q' , W , and R' . For this, a technique like that in Table 5.3-2 may be used. Alternatively, algorithms such as the simplex or Davidon-Fletcher-Powell may be employed.

Note that implicit model following in the regulator case is nothing but a convenient technique for selecting the PI weighting matrices Q' , R' (and W) to guarantee desirable behavior, since the right-hand sides of (5.6-42) are known. Indeed, it is reasonable to select $R = \rho I$ and $Q = I$.

It is possible to design a tracking control system using implicit model following by using the CGT approach. However, this system has an undesirable structure from the point of view of aircraft controls since it generally requires derivatives of the performance output $z(t)$.

Example 5.6-1: Automatic Flare Control by Model-Following Design. Model-following design may be used to design a control system that makes the aircraft behave like an ideal model [Kriendl and Rothschild, 1976]. Such a model may be constructed using the military flying-qualities requirements discussed in Section 4.3 so that it has good performance. However, this is not the only use for model-following design in aircraft controls.

In this example we complete the design of the automatic landing system that was begun in Example 5.5-5. There, we constructed a glide-slope coupler whose function is to conduct an aircraft down a glide path toward the runway. Here we shall show that explicit model-following design may be used to design the automatic flare control system whose function is to cause the aircraft to flare gently to a touchdown. A classical design for a flare control system was performed in Example 4.6-5.

a. *Determining the Reference Model.* The control system is basically an altitude hold system with a time-varying reference or commanded altitude $\underline{h}(t)$. A gentle flare is described by an exponential, so that the commanded altitude should obey the differential equation

$$\dot{\underline{h}} = -\frac{1}{\tau} \underline{h} + r, \quad \text{initial condition } \underline{h}(0) = \underline{h}_0, \quad (1)$$

where τ and \underline{h}_0 are chosen for the desired flare characteristics. Equation (1)

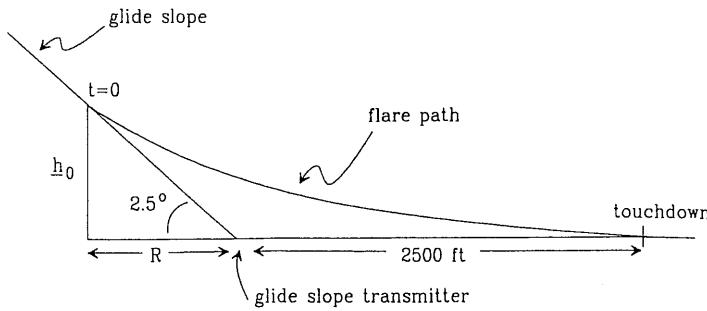


Fig. 5.6-2 Flare-path geometry.

is the reference model [see (5.6-15)]

$$\begin{aligned}\dot{\underline{x}} &= \underline{A}\underline{x} + \underline{B}r \\ \underline{z} &= \underline{H}\underline{x},\end{aligned}\quad (2)$$

with $\underline{A} = -1/\tau$, $\underline{B} = 1$, $\underline{H} = 1$. Thus $\underline{z} = \underline{x} = \underline{h}$. The model reference input is $r(t)$, which is equal to the constant value of zero in this example. Then

$$\underline{h}(t) = \underline{h}_0 e^{-t/\tau}. \quad (3)$$

The geometry of the flare path shown in Fig. 5.6-2 may be used to determine the flare time constant τ and initial altitude \underline{h}_0 (see Example 4.6-5 and [Blakelock, 1965]). In Example 5.5-5 we designed a glide-slope coupler for a total velocity of $V_T = 250$ ft/s. Thus on the glide path the rate of descent is

$$\dot{\underline{h}} = V_T \sin(-2.5^\circ) \approx -V_T \frac{2.5}{57.2958} = -10.91 \text{ ft/s}. \quad (4)$$

The flare control system is turned on at time $t = 0$ shown in the figure. Therefore, for (1) we obtain $\dot{\underline{h}}(0) = -10.91$, and

$$\underline{h}_0 = -\tau \dot{\underline{h}}(0) = 10.91\tau \text{ ft}. \quad (5)$$

The distance R is thus given by

$$R = \frac{\underline{h}_0}{\tan(2.5^\circ)} \approx \underline{h}_0 \times \frac{57.2958}{2.5} = V_T \tau \text{ ft}. \quad (6)$$

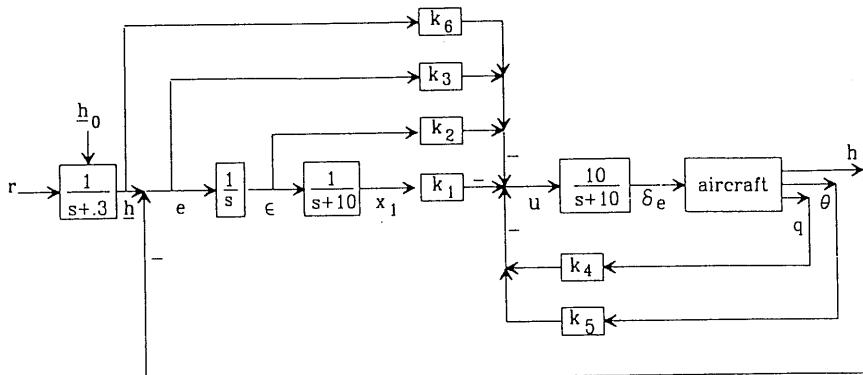


Fig. 5.6-3 Automatic flare control system.

If it is desired to touch down 2500 ft beyond the glide-slope transmitter, and if we assume that $\underline{h}(t)$ given in (1) vanishes in 4τ seconds, then

$$4\tau V_T = R + 2500 = V_T \tau + 2500 \quad (7)$$

or

$$3\tau V_T = 2500, \quad (8)$$

so that

$$\tau = 3.333 \text{ s.} \quad (9)$$

This yields the reference model

$$\dot{\underline{h}} = -0.3\underline{h} + r, \quad \underline{h}_0 = 36.37 \text{ ft}, \quad (10)$$

with reference input $r(t)$ taking the constant value of zero.

b. Basic Aircraft and Controller. The flare control system is shown in Fig. 5.6-3. For small flight-path angles the aircraft altitude is given by

$$\dot{h} = V_T \sin \gamma = V_T \sin(\theta - \alpha) \approx V_T \theta - V_T \alpha, \quad (11)$$

which is the same as the equation in Example 5.5-5 for \dot{d} (with d the off-glide path distance) without the term in γ_r . What this means is that an altitude-hold system is given by the lower d -hold channel in the glide-slope coupler in Fig. 5.5-8, with d replaced everywhere by h and γ_r removed. Indeed, the control gains in that system were used as an initial stabilizing guess in the LQ design for this example.

In this example we want to illustrate the model-following design procedure for the h control channel only. A complete design would include a second velocity-hold channel exactly as in the glide-slope coupler.

We used low-velocity longitudinal Jacobians for a medium-sized transport linearized about $V_T = 250$ ft/s, $cg = 0.25\bar{c}$, as in Example 5.5-5. For the flare control system h -channel, we may use the short-period approximation, with, however, θ retained due to the need to compute the altitude using (11).

The state of the aircraft plus the lead compensator is

$$x = [\alpha \quad \theta \quad q \quad h \quad \delta_e \quad x_1]^T \quad (12)$$

with x_1 the compensator state (see Fig. 5.6-3). The performance output is

$$z = h = [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0]x = Hx \quad (13)$$

and the control input $u(t)$ is the elevator servo command. According to the figure, the measured outputs corresponding to the aircraft and the lead compensator are

$$y = [x_1 \quad q \quad \theta]^T. \quad (14)$$

c. *Explicit Model-Following Control.* We should like the reference output $z(t)$ to follow the model altitude $\underline{h}(t)$ given by (2)/(10). Since the model's reference input $r(t)$ has the constant value of zero, $r(t)$ satisfies the differential equation

$$\dot{r} = 0, \quad (15)$$

so that the command generator polynomial (5.6-21) is given by

$$\Delta(s) = s. \quad (16)$$

The model mismatch altitude error (5.6-7) is given by

$$e = \underline{h} - h. \quad (17)$$

Therefore, the observability canonical form realization (5.6-28) is

$$\dot{\xi} = [-H \quad \underline{H}] \xi = [0 \quad 0 \quad 0 \quad -1 \quad 0 \quad 0 \quad 1] \xi, \quad (18)$$

with $\xi(t)$ the modified state $\Delta(s)[x^T \quad x]^T$.

According to (18), $F = 0$ in the augmented system (5.6-30). Thus we are required to incorporate an integrator in the control system [see (5.6-37) and Fig. 5.6-1]. This we have already done in Fig. 5.6-3.

The overall dynamics of the modified system (5.6-30) are given by

$$\begin{aligned}\dot{X} &= AX + Bu \\ y &= CX,\end{aligned}\tag{19}$$

with X the augmented state that contains the basic aircraft and compensator dynamics, the model dynamics (10), and the integrator required by (18). For convenience, we shall order the states differently than in (5.6-30), taking

$$X = [\alpha \quad \theta \quad q \quad h \quad \delta_e \quad x_1 \quad \epsilon \quad \underline{h}]^T\tag{20}$$

According to Fig. 5.6-3, the outputs are

$$Y = [x_1 \quad \epsilon \quad e \quad q \quad \theta \quad \underline{h}]^T.\tag{21}$$

The model state \underline{h} is included as an output due to the development leading to (5.6-37).

With this structure, the plant matrices are given by

$$A = \begin{bmatrix} -0.64627 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -0.7739 & 0 & -0.52977 & 0 & -0.011 & 0 & 0 & 0 \\ -250 & 250 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.3 \end{bmatrix}\tag{22}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 10 \\ 0 \\ 0 \\ 0 \end{bmatrix}\tag{23}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 57.2958 & 0 & 0 & 0 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}\tag{24}$$

Then, according to Fig. 5.6-3, the control input $u(t)$ is given by

$$u = -Ky = (k_1x_1 + k_2\epsilon + k_3e + k_4q + k_5\theta + k_6h). \quad (25)$$

The control structure shown in Fig. 5.6-3 and described here is nothing but the structure required for model following according to Fig. 5.6-1.

d. PI and LQ Control Gain Design. Although the explicit model following design technique discussed in this section involves using the LQ *regulator* design equations from Table 5.3-1 on the augmented system (5.6-30), we have found that the results are generally better using LQ *tracker design with time-weighted PI*. Thus we used the design equations in Table 5.5-1 with the auxiliary matrices

$$\begin{aligned} G &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]^T \\ F &= [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\ H &= [0 \ 0 \ 0 \ -1 \ 0 \ 0 \ 0 \ 1]^T, \end{aligned} \quad (26)$$

which were determined from Fig. 5.6-3. (Note the redefinition of the matrix H .)

The PI was selected as

$$J = \frac{1}{2} \int_0^\infty (qt^2\tilde{e}^2 + \tilde{u}^2) dt. \quad (27)$$

It is important to note that a sensible formulation of the problem has resulted in the appearance of *only one design parameter*, q , in the PI. Thus we shall not be faced with tuning many design parameters in an effort to obtain suitable responses. In view of the fact that there are eight states and six control gains to determine, this is quite significant. No steady-state error weighting is used in the PI since the plant is of Type I.

After several design iterations using the software of Appendix B to solve for K using the design equations in Table 5.5-1, we decided on $q = 0.001$ and obtained the control gains

$$K = [593.4 \ -59.30 \ 6.154 \ -0.56 \ -1.00 \ -0.01852]. \quad (28)$$

The closed-loop poles were at

$$\begin{aligned} &-0.15 \pm j0.23 \\ &-0.15 \pm j1.02 \\ &-0.30, -0.92 \\ &-9.43, -10.22. \end{aligned} \quad (29)$$

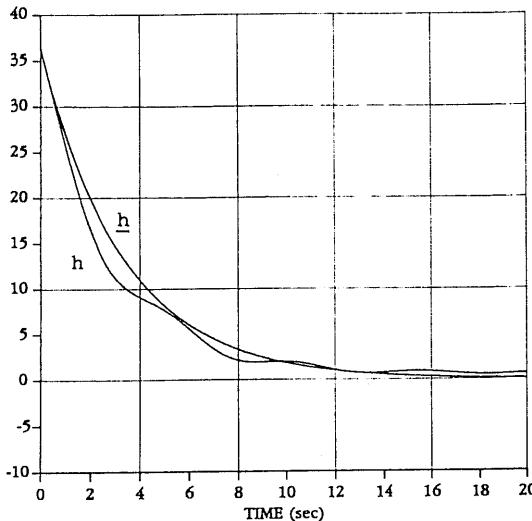


Fig. 5.6-4 Controlled flare, altitude in feet.

Note that the model pole of $s = -0.3$ has not moved since it is uncontrollable.

e. *Simulation and Discussion.* The controlled flare is shown in Fig. 5.6-4—it matches the desired flare $h(t)$ very well. To obtain this graph it is necessary to use initial conditions $x(0)$ corresponding to the equilibrium state on the glide slope from Example 5.5-5. The flight-path angle γ is shown in Fig. 5.6-5a. Shown in Fig. 5.6-5b is the elevator command δ_e ; in examining this figure recall that upward elevator deflection (i.e., back stick) is defined as negative.

The poles in (29) are quite slow and there is one badly damped pair. However, the time responses are acceptable. This is because the flare control system is engaged with the aircraft on the glide path, so that there are no sudden reference command changes to excite the underdamped mode. Moreover, the flare is gentle, so that the time scale of the desired motion is on the order of the time scale of the closed-loop poles.

Although the control gain from the model state h to elevator servo command u is small, it plays a very important function. As may be seen in [Blakelock, 1965], the tendency of the flare control system without model state feedforward is to lag behind the desired response. This results in a flare that is always below the desired path and requires a modification in the design flare time constant τ . The feedforward of h corrects this problem in a simple manner.

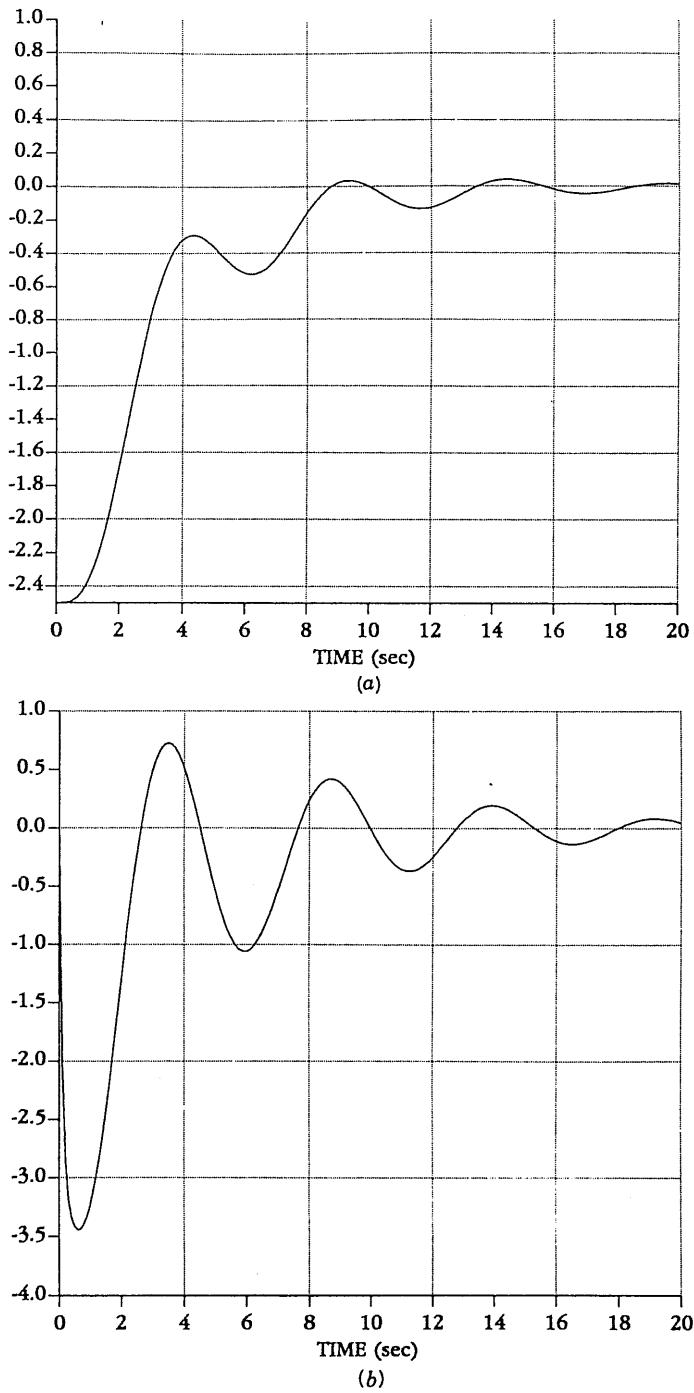


Fig. 5.6-5 Aircraft response during controlled flare: (a) flight-path angle γ (deg); (b) elevator command δ_e (deg).

Using the gains in (29), the compensator in the forward error channel of Fig. 5.6-3 has the transfer function

$$\frac{k_1}{s(s+10)} + \frac{k_2}{s} + k_3 = \frac{6.154(s+0.364)}{s+10}, \quad (30)$$

where the pole at $s = 0$ has been canceled by a zero at $s = 0$ to yield a simplified compensator. Thus there is no integrator in the feedforward path, and the model-following behavior does not rely on the system being of Type I. The ratio of the zero to the poles in the lead compensator is excessive, and the design may be repeated using, for instance, a compensator pole at $s = -5$ instead of $s = -10$ (and no integrator). ■

5.7 LINEAR QUADRATIC DESIGN WITH FULL STATE FEEDBACK

In the previous sections of this chapter we have seen how to design control systems using a variety of modern control techniques that rely only on measuring a system *output*. These output-feedback approaches are very suitable for aircraft controls design since they allow us to design a compensator with any desired dynamical structure. This cannot be accomplished using full state feedback.

In this section we intend to explore full state-variable feedback in the linear quadratic regulator (LQR) for the insight it provides. That is, for the system

$$\dot{x} = Ax + Bu \quad (5.7-1)$$

with $x \in \mathbf{R}^n$, $u \in \mathbf{R}^m$ we want to examine control laws of the form

$$u = -Kx, \quad (5.7-2)$$

which results in the closed-loop system

$$\dot{x} = (A - BK)x \equiv A_c x. \quad (5.7-3)$$

In the previous sections we defined the measurable output

$$y = Cx \quad (5.7-4)$$

and restricted ourselves to controls of the form

$$u = -Ky = -KCx. \quad (5.7-5)$$

Here we plan to examine the simplifications in the control design equations that come about when $C = I$. As we shall see, we can draw some conclusions that will give more insight into modern control theory.

The Relevance of State Feedback

Although all the states are seldom measurable in aircraft control systems, we have several objectives for looking at state-variable feedback design in this section. First, it is clear that state feedback is just the special case of output feedback with $C = I$. That is, it assumes that all the states can be measured. Thus the theory for state-variable feedback will tell us *the best performance that we can expect* in the closed-loop system by using static output feedback, where all of the states are not available as measurements.

Second, the output-feedback design equations in Tables 5.3-1, 5.4-1, and 5.5-1 are not the LQR equations with which the reader may be familiar. We would like to show how they relate to the more traditional Riccati equation.

If all the states are involved in the feedback, there are some very powerful stability results of which the reader should be aware. Indeed, under some reasonable assumptions it is possible to *guarantee the stability of the closed-loop system* using the optimal LQ state feedback gain. Similar theoretical results for output feedback have not yet been discovered.

Finally, we shall need state feedback in Chapter 6 when we discuss dynamic regulators and LQG/LTR robust design. A limitation of state feedback is that all the states are not generally available, but only the outputs are measured. However, we can design a full state feedback $u = -Kx$, and then a dynamic observer to estimate the states from the measured outputs. Then the state *estimates* \hat{x} may be fed back, instead of the states themselves, in a control law such as $u = -K\hat{x}$. The combination of state feedback plus an observer is called a *dynamic regulator*. It is a compensator of the sort used in classical control, but it is easy to design for multivariable systems, overcoming a deficiency of the classical approach, where multiloop and MIMO systems are hard to deal with.

The Riccati Equation and Kalman Gain

By setting $C = I$ all of our work in Sections 5.3 and 5.4 applies to state feedback. That is, all the work of deriving the control design equations for state feedback has already been done. Let us see how the LQR design equations simplify in the case of full state feedback.

To regulate the performance output

$$z = Hx \quad (5.7-6)$$

to zero, let us select the PI

$$J = \frac{1}{2} \int_0^\infty (x^T Q x + u^T R u) dt \quad (5.7-7)$$

with $Q = H^T H \geq 0$, $R > 0$.

The output feedback gain K in (5.7-5) that minimizes the PI may be found using the design equations in Table 5.3-1. To obtain the optimal state feedback in (5.7-2), we may simply set $C = I$ in the table. Then there results

$$0 = A_c^T P + P A_c + Q + K^T R K \quad (5.7-8)$$

$$0 = A_c S + S A_c^T + X \quad (5.7-9)$$

$$K = R^{-1} B^T P S S^{-1}, \quad (5.7-10)$$

where the initial state autocorrelation is

$$X = E\{x(0)x^T(0)\}. \quad (5.7-11)$$

The problems in computing the output feedback gains include the need to know X and the selection of an initial stabilizing gain K_0 for the algorithm in Table 5.3-2. Moreover, although we gave conditions for the convergence to a local minimum of the algorithm in that table, little is known about the necessary and sufficient conditions for the existence of an output feedback gain that satisfies the design equations and stabilizes the plant.

All of these problems vanish in the case of state feedback, as we shall now show. According to (5.7-10),

$$K = R^{-1} B^T P; \quad (5.7-12)$$

that is, the solution S to (5.7-9) is not needed to solve for the optimal state feedback gain. The gain K is called the *Kalman gain*. Using (5.7-12) in (5.7-8) yields

$$0 = A_c^T P + P A_c + Q + P B R^{-1} B^T P \quad (5.7-13)$$

or, according to (5.7-3)

$$\begin{aligned} 0 &= (A - B R^{-1} B^T P)^T P + P(A - B R^{-1} B^T P) \\ &\quad + Q + P B R^{-1} B^T P \\ 0 &= A^T P + P A + Q - P B R^{-1} B^T P. \end{aligned} \quad (5.7-14)$$

This matrix quadratic equation is called the *algebraic Riccati equation (ARE)*. It is named after Count J. F. Riccati, who used a related equation in the

TABLE 5.7-1. LQR with State Feedback

System Model

$$\dot{x} = Ax + Bu$$

Control

$$u = -Kx$$

Performance Index

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt$$

Optimal LQ Design Equations

ALGEBRAIC RICCATI EQUATION (ARE)

$$0 = A^T P + PA + Q - PBR^{-1}B^T P \quad (5.7-16)$$

KALMAN GAIN

$$K = R^{-1}B^T P \quad (5.7-17)$$

Optimal Cost

$$J = \frac{1}{2} x^T(0) Px(0)$$

study of heat flow [Riccati, 1724]. Since the equation is equal to its own transpose (verify!), the solution P is symmetric ($P = P^T$).

Since S is not needed to find the optimal state-feedback gain K , this gain does not depend on X in (5.7-9). That is, contrary to the case with output feedback; to compute the optimal state feedback gains *no information about the initial state $x(0)$ is needed*. Thus it is not required to take expected values of the PI as we did in Section 5.3. Therefore, according to the development in that section, the optimal cost is given by

$$J = \frac{1}{2} x^T(0) Px(0). \quad (5.7-15)$$

The state-feedback LQR is summarized in Table 5.7-1.

Setting $C = I$ has allowed us to replace the solution of three coupled matrix equations by the solution of *one nonlinear matrix equation* for P . Then the Kalman gain is given in terms of P by (5.7-17). The importance of this is that there are many good techniques for solving the Riccati equation using *standard software packages* (e.g., ORACLS [Armstrong, 1980], MATRIX_x [1989], PC-MATLAB [Moler et al., 1987], and IMSL [1980]). On the other hand, the specialized software for solving the output-feedback problem in

Tables 5.3-1, 5.4-1, or 5.5-1 can be used to solve the full state-feedback problem by setting $C = I$.

Guaranteed Closed-Loop Stability

The theory for the LQ regulator with state feedback is well developed. In fact, the next stability result is so fundamental that we set it apart as a theorem [Lewis, 1986]. The notion of detectability was introduced while discussing Table 5.3-2. We say that (A, H) is detectable if there exists an L so that $A - LH$ is stable; this amounts to the observability of the unstable modes of A . We say that (A, B) is *stabilizable* if there exists a feedback gain K such that $A_c = A - BK$ is stable. This amounts to the controllability of the unstable modes of A .

Theorem. Let H be any matrix so that $Q = H^T H$. Suppose that (H, A) is detectable. Then (A, B) is stabilizable if and only if:

- (a) There exists a unique positive semidefinite solution P to the Riccati equation, and
- (b) The closed-loop system (5.7-3) is asymptotically stable if the Kalman gain K is computed using (5.7-17) in terms of this positive semidefinite solution P . ■

This result is at the heart of modern control theory. Exactly as in classical control, it allows us to examine *open-loop* properties (i.e., detectability and stabilizability) and draw conclusions about the closed-loop system. As long as (H, A) is detectable, so that all the unstable modes appear in the PI, and (A, B) is stabilizable, so that the control $u(t)$ has sufficient influence on the system, the LQ regulator using state feedback will *guarantee* a stable closed-loop system. A similar easily understandable result has not yet been discovered for output feedback.

Detectability is implied by the stronger condition of observability, which is easy to check by verifying that the observability matrix has full rank n (see Section 5.3). Stabilizability is implied by controllability, which is easy to check by verifying that the controllability matrix has full rank n (see Section 5.2). Thus the controllability of (A, B) and the observability of (H, A) guarantee closed-loop stability of the LQ regulator with state feedback.

This theorem, coupled with the availability of good software for solving the ARE, means that it is always straightforward to find a state-variable feedback gain K that stabilizes any stabilizable plant, no matter how many inputs or outputs it has.

Since output feedback amounts to a partial state feedback, it is clear that if the conditions of the theorem do not hold, we should not expect to be able to stabilize the plant using any output feedback. (Unless time-dependent weighting of the form t^k is used in the PI to avoid the observability

requirement; see Section 5.5). Thus in the case of output-feedback design these conditions should hold *as a minimum*. In fact, we saw that the Algorithm of Table 5.3-2 requires the detectability of (\sqrt{Q}, A) and the output stabilizability of the system. Output stabilizability is a stronger condition than stabilizability.

In the case of a full state feedback, it is possible in simple examples to give a direct correlation between the PI weighting matrices and the closed-loop poles. Let us investigate this connection for systems obeying Newton's laws.

Example 5.7-1: LQR with State Feedback for Systems Obeying Newton's Laws. In this example we shall see that in the case of full state feedback for simple systems, there is a direct connection between the PI weights and the closed-loop damping ratio and natural frequency.

Systems obeying Newton's laws may be described by the state equation

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u = Ax + Bu, \quad (1)$$

where the state is $x = [d \ v]^T$, with $d(t)$ the position and $v(t)$ the velocity, and the control $u(t)$ is an acceleration input. Indeed, note that (1) says nothing other than $\ddot{d} = u$, or $a = F/m$.

Let the PI be

$$J = \frac{1}{2} \int_0^\infty (x^T Q x + u^2) dt \quad (2)$$

with $Q = \text{diag}\{q_d^2, q_v\}$. In this example, we shall see the effect of q_d and q_v . Note that it is not useful to include a separate control weighting r , since only the ratios q_d^2/r and q_v/r are important in J .

Since the Riccati solution P is symmetric, we may assume that

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \quad (3)$$

for some scalars p_1, p_2, p_3 to be determined. Using A , B , Q , and $r = 1$ in the Riccati equation in Table 5.7-1 yields

$$0 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} + \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} q_d^2 & 0 \\ 0 & q_v \end{bmatrix} - \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}. \quad (4)$$

The reader should verify that this may be multiplied out to obtain the three

scalar equations

$$0 = -p_2^2 + q_d^2 \quad (5a)$$

$$0 = p_1 - p_2 p_3 \quad (5b)$$

$$0 = 2p_2 - p_3^2 + q_v. \quad (5c)$$

Solving these equations in the order (5a), (5c), (5b) gives

$$p_2 = q_d \quad (6a)$$

$$p_3 = \sqrt{2} \sqrt{q_d + \frac{q_v}{2}} \quad (6b)$$

$$p_1 = q_d \sqrt{2} \sqrt{q_d + \frac{q_v}{2}}, \quad (6c)$$

where we have selected the signs that make P positive definite.

According to Table 5.7-1, the Kalman gain is equal to

$$K = R^{-1}B^T P = [0 \ 1] \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} = [p_2 \ p_3]. \quad (7)$$

Therefore,

$$K = \begin{bmatrix} q_d & \sqrt{2} \sqrt{q_d + \frac{q_v}{2}} \end{bmatrix}. \quad (8)$$

It should be emphasized that in the case of state feedback, we have been able to find an *explicit expression* for K in terms of the PI weights. This is not possible for output feedback.

Using (8), the closed-loop system matrix is found to be

$$A_c = (A - BK) = \begin{bmatrix} 0 & 1 \\ -q_d & -\sqrt{2} \sqrt{q_d + \frac{q_v}{2}} \end{bmatrix}. \quad (9)$$

Therefore, the closed-loop characteristic polynomial is

$$\Delta_c(s) = |sI - A_c| = s^2 + 2\xi\omega s + \omega^2, \quad (10)$$

with the optimal natural frequency ω and damping ratio ξ given by

$$\omega = \sqrt{q_d}, \quad \xi = \frac{1}{\sqrt{2}} \sqrt{1 + \frac{q_v}{2q_d}}. \quad (11)$$

It is now clear how selection of the weights in the PI affects the closed-loop behavior. Note that if no velocity weighting q_v is used, the damping ratio becomes the familiar $1/\sqrt{2}$.

Note that (A, B) is reachable since

$$U = [B \ AB] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (12)$$

is nonsingular. The observability matrix is

$$O = \begin{bmatrix} \sqrt{Q} \\ \sqrt{Q} A \end{bmatrix} = \begin{bmatrix} q_d & 0 \\ 0 & \sqrt{q_v} \\ 0 & q_d \\ 0 & 0 \end{bmatrix}. \quad (13)$$

Therefore, observability is guaranteed if and only if the position weighting q_d is greater than zero. Then the theorem says that we should be able to rely on a stable closed-loop system. Examining (11) makes it clear that this is indeed the case. ■

5.8 SUMMARY

In this chapter we showed how to use modern control techniques to design multivariable and multiloop aircraft flight control systems. The approach is based on the state-variable model and a mathematical performance criterion selected according to the performance objectives. The matrix of control gains is determined by solving explicit matrix equations using computer software. Using such an approach, all the feedback loops are closed simultaneously to yield the guaranteed performance desired. This is in contrast to the classical techniques of Chapter 4, which relied on trial-and-error successive loop closures to find the control gains individually.

Two basic modern design techniques were covered. In Section 5.2 we discussed eigenstructure assignment techniques that take advantage of the freedom inherent in design for systems with more than one input and/or output to assign the closed-loop poles and eigenvectors. In the remainder of the chapter we covered linear quadratic (LQ) techniques, where the control gains are selected to minimize generalized quadratic performance indices (PIs). Design equations were derived for the control gains minimizing these PIs and listed in tabular form for easy reference. The design equations may be solved for the control gains using software like that described in Appendix B.

In Section 5.5 the thrust was to introduce modified nonstandard PIs allowing LQ designs with only a small number of design parameters that

require tuning for suitable performance. The point was made that successful control system design hinges on the selection of a suitable PI.

Our primary thrust was to use output feedback to allow the design of a compensator with any desired structure. The PI was an integral of the squares of the states and control inputs; thus the LQ techniques used in this chapter are *time-domain* techniques.

REFERENCES

- AFWAL-TR-84-3008*, "AFTI/F-16 development and integration program, DFCS phase final technical report," General Dynamics, Ft. Worth, Tex., Dec. 1984.
- Andry, A. N., Jr., E. Y. Shapiro, and J. C. Chung, "Eigenstructure assignment for linear systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-19, no. 5, pp. 711-729, Sept. 1983.
- Andry, A. N., Jr., J. C. Chung, and E. Y. Shapiro, "Modalized observers," *IEEE Trans. Autom. Control*, vol. AC-29, no. 7, July 1984, pp. 669-672.
- Armstrong, E. S., *ORACLS: A Design System for Linear Multivariable Control*. New York: Marcel Dekker, 1980.
- Bartels, R. H., and G. W. Stewart, "Solution of the matrix equation $AX + XB = C$," *Commun. ACM*, vol. 15, no. 9, Sept. 1972, pp. 820-826.
- Blakelock, J. H., *Automatic Control of Aircraft and Missiles*. New York: Wiley, 1965.
- Broussard, J., and N. Halyo, "Active flutter control using discrete optimal constrained dynamic compensators," *Proc. Am. Control Conf.*, June 1983, pp. 1026-1034.
- Bryson, A. E., Jr., "New concepts in control theory, 1959-1984," *J. Guid.*, vol. 8, no. 4, July-Aug. 1985, pp. 417-425.
- Bryson, A. E., Jr., and Y.-C. Ho, *Applied Optimal Control*. New York: Hemisphere, 1975.
- Davison, E. J., and I. J. Ferguson, "The design of controllers for the multivariable robust servomechanism problem using parameter optimization methods," *IEEE Trans. Autom. Control*, vol. AC-26, no. 1, Feb. 1981, pp. 93-110.
- D'Azzo, J. J., and C. H. Houpis, *Linear Control System Analysis and Design*. New York: McGraw-Hill, 1988.
- Doyle, J. C., and G. Stein, "Multivariable feedback design: concepts for a classical/modern synthesis," *IEEE Trans. Autom. Control*, vol. AC-26, no. 1, Feb. 1981, pp. 4-16.
- Franklin, G. F., J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Reading, Mass.: Addison-Wesley, 1986.
- Gangsaas, D., K. R. Bruce, J. D. Blight, and U.-L. Ly, "Application of modern synthesis to aircraft control: three case studies," *IEEE Trans. Autom. Control*, vol. AC-31, no. 11, Nov. 1986, pp. 995-1014.
- Grimble, M. J., and M. A. Johnson, *Optimal Control and Stochastic Estimation: Theory and Applications*, Vol. 1, New York: Wiley, 1988.
- Harvey, C. A., and G. Stein, "Quadratic weights for asymptotic regulator properties," *IEEE Trans. Autom. Control*, vol. AC-23, no. 3, 1978, pp. 378-387.

- IMSL, *Library Contents Document*, 8th ed., International Mathematical and Statistical Libraries, Inc., NBC Building, 7500 Bellaire Blvd., Houston, TX 77036, 1980.
- Kailath, T., *Linear Systems*. Englewood Cliffs, N.J.: Prentice Hall, 1980.
- Kalman, R., "Contributions to the theory of optimal control," *Bol. Soc. Math. Mex.*, vol. 5, 1958, pp. 102-119.
- Kalman, R. E., "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, 1960, pp. 34-35.
- Kreindler, E., and D. Rothschild, "Model-following in linear-quadratic optimization," *AIAA J.*, vol. 14, no. 7, July 1976, pp. 835-842.
- Kwakernaak, H., and R. Sivan, *Linear Optimal Control Systems*. New York: Wiley, 1972.
- Kwon, B.-H., and M.-J. Youn, "Eigenvalue-generalized eigenvector assignment by output feedback," *IEEE Trans. Autom. Control*, vol. AC-32, no. 5, May 1987, pp. 417-421.
- Levine, W. S., and M. Athans, "On the determination of the optimal constant output feedback gains for linear multivariable systems," *IEEE Trans. Autom. Control*, vol. AC-15, no. 1, Feb. 1970, pp. 44-48.
- Lewis, F. L., *Optimal Control*. New York: Wiley, 1986.
- Ly, U.-L., A. E. Bryson, and R. H. Cannon, "Design of low-order compensators using parameter optimization," *Automatica*, vol. 21, no. 3, 1985, pp. 315-318.
- MacFarlane, A. G. J., "The calculation of functionals of the time and frequency response of a linear constant coefficient dynamical system," *Quart. J. Mech. Appl. Math.*, vol. 16, pt. 2, 1963, pp. 259-271.
- MATRIX_x, Integrated Systems, Inc., 2500 Mission College Blvd., Santa Clara, CA 95054, 1989.
- McRuer, D., I. Ashkenas, and D. Graham, *Aircraft Dynamics and Automatic Control*. Princeton N.J.: Princeton University Press, 1973.
- Mil. Spec. 1797*, "Flying qualities of piloted vehicles," 1987.
- Moerder, D. D., and A. J. Calise, "Convergence of a numerical algorithm for calculating optimal output feedback gains," *IEEE Trans. Autom. Control*, vol. AC-30, no. 9, Sept. 1985, pp. 900-903.
- Moler, C., J. Little, and S. Bangert, *PC-Matlab*, The Mathworks, Inc., 20 North Main St., Suite 250, Sherborn, MA 01770, 1987.
- Moore, B. C., "On the flexibility offered by state feedback in multivariable systems beyond closed-loop eigenvalue assignment," *IEEE Trans. Autom. Control*, Oct. 1976, pp. 689-692.
- Nelder, J. A., and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, 1964, pp. 308-313.
- O'Brien, M. J., and J. R. Broussard, "Feedforward control to track the output of a forced model," *Proc. IEEE Conf. Decision Control*, Dec. 1978.
- Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*. New York: Cambridge University Press, 1986.

- Quintana, V. H., M. A. Zohdy, and J. H. Anderson, "On the design of output feedback excitation controllers of synchronous machines," *IEEE Trans. Power Appar. Syst.*, vol. PAS-95, no. 3, 1976, pp. 954-961.
- Riccati, J. F., "Animadversiones in aequationes differentiales secundi gradus," *Actorum Eruditorum quae Lipsiae publicantur, Suppl.*, vol. 8, 1724, pp. 66-73.
- Sobel, K. M., and E. Y. Shapiro, "Eigenstructure assignment for design of multimode flight control systems," *IEEE Control Syst.*, May 1985, pp. 9-14.
- Söderström, T., "On some algorithms for design of optimal constrained regulators," *IEEE Trans. Autom. Control*, vol. AC-23, no. 6, Dec. 1978, pp. 1100-1101.
- Srinathkumar, S., "Eigenvalue/eigenvector assignment using output feedback," *IEEE Trans. Autom. Control*, vol. AC-23, no. 1, Feb. 1978, pp. 79-81.
- Stern, G., and H. A. Henke, "A design procedure and handling-quality criteria for lateral-directional flight control systems," *AF-FDL-TR-70-152*, Wright-Patterson AFB, Ohio, May 1971.
- Stevens, B. L., F. L. Lewis, and F. Al-Sunni, "Aircraft flight controls design using output feedback," *J. Guid. Control Dynam.*, 1991 (to appear).

PROBLEMS

Section 5.2

5.2-1 Eigenstructure Assignment with Full State Feedback. The short-period approximation of an aircraft with the cg far aft might be described by

$$\dot{x} = \begin{bmatrix} -1.10188 & 0.90528 & -0.00212 \\ 4.0639 & -0.77013 & -0.16919 \\ 0 & 0 & -10 \end{bmatrix}x + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 10 & 0 \end{bmatrix}u, \quad (1)$$

which includes an elevator actuator of $10/(s + 10)$. The state is $x = [\alpha \quad q \quad \delta_e]^T$. An extra control input u_2 has been added to illustrate the extra design freedom available in multivariable systems.

- (a) Find the poles.
- (b) To conform to flying-qualities specifications, it is desired to assign closed-loop short-period eigenvalues λ_1 and λ_2 of $-2 \pm j2$. The actuator pole does not matter but may be assigned to $s = -15$ to speed up its response. The desired closed-loop eigenvectors are

$$v_1 = v_2^* = [0.20 + j0.35 \quad -0.98 + j0.07 \quad 0]^T,$$

$$v_3 = [0 \quad 0 \quad 1]^T.$$

Find the state-feedback gain K in $u = -Kx$ to assign the desired eigenstructure.

- 5.2-2 Eigenstructure Assignment with Output Feedback.** In Problem 5.2-1, a more realistic situation occurs when only measurements of α and q are taken. Then the control is $u = -Ky$ with $y = [\alpha \quad q]^T$. Only two poles may now be assigned. Select desired closed-loop poles as λ_1 and λ_2 in Problem 5.1, with the same eigenvectors v_1 and v_2 . Find the required output-feedback gain K . Find the closed-loop poles. What happens to the actuator pole?
- 5.2-3** In Problem 5.2-1, change the control input to $B = [0 \quad 0 \quad 10]^T$ and use feedback of the output $y = [\alpha \quad q]^T$. Now two poles can be assigned, but there is no freedom in selecting the eigenvectors. Select the desired closed-loop poles $\lambda_1 = \lambda_2^* = -2 + j2$. Find the achievable associated eigenvectors v_1 and v_2 . Find the feedback gain K . Find the closed-loop actuator pole.

Section 5.3

- 5.3-1** Fill in the details in the derivation of the design equations in Table 5.3-1.
- 5.3-2 Output-Feedback Design for Scalar Systems**
- Consider the case where $x(t)$, $u(t)$, $y(t)$ are all scalars. Show that the solution S to the second Lyapunov equation in Table 5.3-1 is not needed to determine the output-feedback gain K . Find an explicit solution for P and hence for the optimal gain K .
 - Repeat for the case where $x(t)$ and $y(t)$ are scalars, but $u(t)$ is an m -vector.
- 5.3-3** Use (5.3-28) to eliminate K in the Lyapunov equations of Table 5.3-1, hence deriving two coupled nonlinear equations that may be solved for the optimal auxiliary matrices S and P . Does this simplify the solution of the output-feedback design problem?
- 5.3-4 Software for Output-Feedback Design.** Write a program that finds the gain K minimizing the PI in Table 5.3-1 using the SIMPLEX algorithm in [Press, et al., 1986]. Use it to verify the results of Example 5.3-1. Can you tune the elements of Q and R to obtain better closed-loop responses than the ones given?
- 5.3-5** For the system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u, \quad y = [1 \quad 1]x \quad (1)$$

find the output-feedback gain that minimizes the PI in Table 5.3-1 with $Q = I$. Try various values of R to obtain a good response. You will need the software from Problem 5.3-4. The closed-loop step

response may be plotted using the software described in Appendix B. [Note that system (1) is nothing but Newton's law, since if $x = [p \ v]^T$, then $\ddot{p} = u$, where $u(t)$ may be interpreted as an acceleration input F/m .]

- 5.3-6 Gradient-Based Software for Output-Feedback Design.** Write a program that finds the gain K minimizing the PI in Table 5.3-1 using the Davidon–Fletcher–Powell algorithm in [Press et al., 1986]. Use it to verify the results of Example 5.3-1.

Section 5.4

- 5.4-1** Derive (5.4-31).
- 5.4-2** Derive the necessary conditions in Table 5.4-1.
- 5.4-3** In Example 5.4-1, use the observability matrix to verify that the original proposed value of $Q = H^T H$ has (\sqrt{Q}, A) unobservable while the Q that contains a $(5, 5)$ element has (\sqrt{Q}, A) observable.
- 5.4-4 Software for LQ Output-Feedback Design.** Write a program to solve for the optimal gain K in Table 5.4-1 using the SIMPLEX algorithm in [Press et al., 1986]. Use it to verify Example 5.4-1.
- 5.4-5** In Example 5.4-1 we used an output with four components. There is an extra degree of freedom in the choice of control gains that may not be needed. Redo the example using the software from Problem 5.4-4, with the output defined as $y = [\alpha_F \ q \ \epsilon]^T$.
- 5.4-6** To see whether the angle-of-attack filter in Example 5.4-1 complicates the design, redo the example using $y = [\alpha \ q \ e \ \epsilon]^T$.
- 5.4-7** Redo Example 5.4-1 using root-locus techniques like those in Chapter 4. Based on this, are the gains selected by the LQ algorithm sensible from the point of view of classical control theory?
- 5.4-8 Gradient-Based Software for LQ Output-Feedback Design.** Write a program to solve for the optimal gain K in Table 5.4-1 using the Davidon–Fletcher–Powell algorithms in [Press et al., 1986]. Use it to verify Example 5.4-1.
- 5.4-9 Gain Scheduling.** To implement a control law on an aircraft, it must be gain scheduled over the flight envelope where it will be used. In Section 3.3 a software longitudinal model was given for a transport aircraft. In Section 3.4 it was shown how to use a trim program to obtain linearized state-variable models at different trim conditions. Using the trim software, obtain three state-variable models for the short-period approximation at 0 ft altitude for speeds of 170, 220, and 300 ft/s. Redo the normal acceleration CAS in Example 5.4-1 for

each of these three state-space models. The result is three sets of control gains, each of which is valid for one of the trim conditions. To implement the gain-scheduled control law, write a simple program that selects between the control gains depending on the actual measured speed of the aircraft. Use linear interpolation between the three gain element values for points between the three equilibrium conditions.

Section 5.5

- 5.5-1** Show the validity of (5.5-8) and (5.5-9).
- 5.5-2** Use a technique like that employed in Section 5.3 to derive the expression for the optimal cost in terms of P_k that appears in Table 5.5-1. You will need to successively integrate by parts [MacFarlane, 1963].
- 5.5-3** Derive the necessary conditions in Table 5.5-1.
- 5.5-4 Software for Output-Feedback LQR Design.** Write a program that finds the gain K minimizing the PI in Table 5.3-1 using the SIMPLEX algorithm in [Press et al., 1986]. Include gain-element weighting using (5.5-6). Use this software to verify the results of Example 5.5-1.
- 5.5-5 Software for Output-Feedback LQ Tracker Design.** Write a program that finds the gain K minimizing the PI in Table 5.4-1 using the SIMPLEX algorithm in [Press et al., 1986]. Include gain-element weighting using (5.5-6).
- 5.5-6** In Example 5.4-1 we used an output with four components. There is an extra degree of freedom in the choice of control gains which may not be needed. Using the gain-element weighting software from Problem 5.5-5, redo the example with a large weight on the gain element multiplying $e(t)$ to drive it to zero. Is the performance as good? Try tuning the performance index weights for better performance.
- 5.5-7 Software for Time-Weighted Output-Feedback Tracker Design.** Write a program that finds the gain K minimizing the PI in Table 5.5-1 using the SIMPLEX algorithm in [Press et al., 1986]. Include gain-element weighting using (5.5-6). Use this software to verify the results of Example 5.5-2. Redo the design using weighting of t^3, t^4 . Is there any significant difference from the t^2 case?
- 5.5-8 Root-Locus Design.** Redo Example 5.5-3, finding the control gains using root-locus techniques like those in Chapter 4. Compare this procedure to modern design using software that solves the design equations in Table 5.5-1.

Section 5.6

- 5.6-1** Derive the implicit-model following design equations (5.6.43)–(5.6.45).
- 5.6-2** Using the control gains found in the flare control system of Example 5.6-1, determine the compensator zeros in Fig. 5.6-3.
- 5.6-3** A system obeying Newton's laws is described by the state equations

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u, \quad y = [1 \ 1]x.$$

The state is $x = [p \ v]^T$, with $p(t)$ the position and $v(t)$ the velocity.

Using the CGT approach, design an explicit-model following controller that makes the position follow a quadratic input command $r(t) = r_0 + r_1 t + r_2 t^2$.

- 5.6-4** It is desired to make the scalar plant

$$\dot{x} = x + u, \quad y = x, \quad z = x$$

behave like the scalar model

$$\dot{\underline{x}} = -2\underline{x} + r, \quad \underline{y} = \underline{x}, \quad \underline{z} = \underline{x}$$

with reference input r equal to the unit step. Use explicit model following to design a servosystem:

- (a) Draw the controller structure.
- (b) Select the control gains using LQR design on the augmented system.

Section 5.7

- 5.7-1 Damped Harmonic Oscillator.** The damped harmonic oscillator is described by

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u$$

with ζ the damping ratio and ω_n the natural frequency. This system is useful in modelling systems with an oscillatory mode (e.g., short-period mode, fuel slosh).

- (a) Repeat Example 5.7-1 for this system.
- (b) For several choices of the PI weighting parameters, find the optimal gain and simulate the closed-loop step response. (You can check your results using the software written to solve the design equations in Table 5.3-1, 5.4-1, or 5.5-1 by setting $C = I$ there.)

CHAPTER 6

ROBUSTNESS AND MULTIVARIABLE FREQUENCY-DOMAIN TECHNIQUES

6.1 INTRODUCTION

Modeling Errors and Stability Robustness

In the design of aircraft control systems it is important to realize that the rigid-body equations that are the basis for design in Chapters 4 and 5 are only an approximation to the nonlinear aircraft dynamics. An aircraft has flexible modes that are important at high frequencies; we neglected these in our rigid-body design model. These *unmodeled high-frequency dynamics* can act to destabilize a control system that may have quite suitable behavior in terms only of the rigid-body model.

Moreover, as the aircraft changes its equilibrium flight condition, the linearized rigid-body model describing its perturbed behavior changes. This *parameter variation* is a low-frequency effect that can also act to destabilize the system. To compensate for this variation, one may determine suitable controller gains for linearized models at several design equilibrium points over a flight envelope. Then, these design gains may be scheduled in computer lookup tables for suitable controller performance over the whole envelope. For gain scheduling to work, it is essential for the controller gains at each design equilibrium point to guarantee stability for actual flight conditions near that equilibrium point. Thus it is important to design controllers that have *stability robustness*, which is the ability to provide stability in spite of modeling errors due to high-frequency unmodeled dynamics and plant parameter variations.

Disturbances and Performance Robustness

It is often important to account for disturbances such as wind gusts and also for sensor measurement noise. Disturbances can often act to cause unsatisfactory performance in a system that has been designed without taking them into account. Thus it is important to design controllers that have *performance robustness*, which is the ability to guarantee acceptable performance (in terms, for instance, of percent overshoot, settling time, etc.) even though the system may be subject to disturbances.

Classical Robust Design

In classical control, robustness may be designed into the system from the beginning by providing sufficient gain and phase margin to counteract the effects of inaccurate modeling or disturbances. In terms of the Bode magnitude plot, it is known that the loop gain should be high at low frequencies for performance robustness, but low at high frequencies, where unmodeled dynamics may be present, for stability robustness. The concept of bandwidth is important in this connection, as is the concept of the sensitivity function.

Classical controls design techniques are generally in the frequency domain, so they afford a convenient approach to robust design for single-input/single-output (SISO) systems. However, it is well known that the individual gain margins, phase margins, and sensitivities of all the SISO transfer functions in a multivariable or multiloop system have little to do with its overall robustness. Thus there have been problems in extending classical robust design notions to multi-input/multi-output (MIMO) systems.

Modern Robust Design

Modern control techniques provide a direct way to design multiloop controllers for MIMO systems by closing all the loops simultaneously. Performance is guaranteed in terms of minimizing a quadratic performance index (PI) which, with a sensible problem formulation, generally implies closed-loop stability as well. However, all our work in Chapter 5 assumed that the aircraft model is exactly known and that there are no disturbances. In fact, this is rarely the case.

In this chapter we show that the classical frequency-domain robustness measures are easily extended to MIMO systems in a rigorous fashion by using the notion of the *singular value*. In Section 6.2 we develop the *multivariable loop gain and sensitivity* and describe the *multivariable Bode magnitude plot*. In terms of this plot, we present bounds that *guarantee* both robust stability and robust performance for multivariable systems, deriving notions that are entirely analogous to those in classical control.

In Section 6.3 we give a design technique for robust multivariable controllers using modern output-feedback theory, showing how robustness may be guaranteed. The approach is a straightforward extension of classical techniques. To yield both suitable time-domain performance and robustness, an iterative approach is described that is simple and direct using the software described in Appendix B. We illustrate by designing a pitch rate control system that has good performance despite the presence of flexible modes and wind gusts.

A popular modern approach to the design of robust controllers is *linear quadratic Gaussian/loop-transfer recovery (LQG/LTR)*. This approach has been used extensively by Honeywell in the design of advanced multivariable aircraft control systems. LQG/LTR relies on the *separation principle*, which involves designing a full state-variable feedback (as in Section 5.7) and then an *observer* to provide the state estimates for feedback purposes. The result is a dynamic compensator that is similar to those resulting from classical control approaches. The importance of the separation principle is that compensators can be designed for *multivariable systems* in a straightforward manner by solving matrix equations. In Section 6.4 we discuss observers and the Kalman filter. In Section 6.5 we cover LQG/LTR design.

A recent approach to modern robust design is *H-infinity* design [Francis et al., 1984; Doyle et al., 1989; Kaminer et al., 1990]. However, using *H-infinity* design it is difficult to obtain a controller with a desired structure. For this reason, as well as due to space limitations, we shall not cover *H-infinity* design.

6.2 MULTIVARIABLE FREQUENCY-DOMAIN ANALYSIS

We shall deal with system uncertainties, as in classical control, using robust design techniques which are conveniently examined in the frequency domain. To this point, our work in modern control has been in the time domain, since the LQ performance index is a time-domain criterion.

One problem that arises immediately for MIMO systems is that of extending the SISO Bode magnitude plot. We are not interested in making several individual SISO frequency plots for various combinations of the inputs and outputs in the MIMO system and examining gain and phase margins. Such approaches have been tried and may not always yield much insight on the true behavior of the MIMO system. This is due to the coupling that generally exists between *all* inputs and *all* outputs of a MIMO system.

Thus in this section we introduce the *multivariable loop gain and sensitivity* and the *multivariable Bode magnitude plot*, which will be nothing but the plot versus frequency of the *singular values* of the transfer function matrix. This basic tool allows much of the rich experience of classical control theory to be

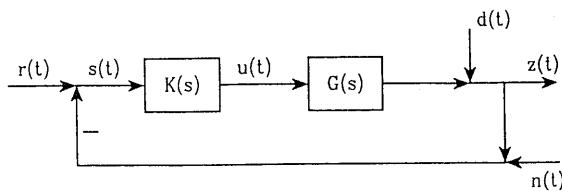


Fig. 6.2-1 Standard feedback configuration.

applied to MIMO systems. Thus we shall discover that for robust performance the minimum singular value of the loop gain should be large at low frequencies, where disturbances are present. On the other hand, for robust stability the maximum singular value of the loop gain should be small at high frequencies, where there are significant modeling inaccuracies. We shall also see that to guarantee stability despite parameter variations in the linearized model due to operating point changes, the maximum singular value should be below an upper limit.

Sensitivity and Cossensitivity

Figure 6.2-1 shows a standard feedback system of the sort that we have seen several times in our work to date. The plant is $G(s)$, and $K(s)$ is the feedback/feedforward compensator, which can be designed by any of the techniques we have covered. The plant output is $z(t) \in \mathbf{R}^q$, the plant control input is $u(t) \in \mathbf{R}^m$, and the reference input is $r(t) \in \mathbf{R}^q$.

We have mentioned in Section 5.4 that perfect tracking may not be achieved unless the number m of control inputs $u(t)$ is greater than or equal to the number q of performance outputs $z(t)$ [Kwakernaak and Sivan, 1972]. Therefore, we shall assume that $m = q$ so that the plant $G(s)$ and compensator $K(s)$ are square. This is only a consequence of sensible design, not a restriction on the sorts of plants that may be considered.

We have added a few items to the figure to characterize uncertainties. The signal $d(t)$ represents a *disturbance* acting on the system of the sort appearing in classical control. This could represent, for instance, wind gusts. The *sensor measurement noise* or errors are represented by $n(t)$. Both of these signals are generally vectors of dimension q . Typically, the disturbances occur at low frequencies, say below some ω_d , while the measurement noise $n(t)$ has its predominant effect at high frequencies, say above some value ω_n . Typical Bode plots for the magnitudes of these terms appear in Fig. 6.2-2 for the case that $d(t)$ and $n(t)$ are scalars. The reference input is generally also a low-frequency signal (e.g., the unit step).

The tracking error is

$$e(t) \equiv r(t) - z(t). \quad (6.2-1)$$

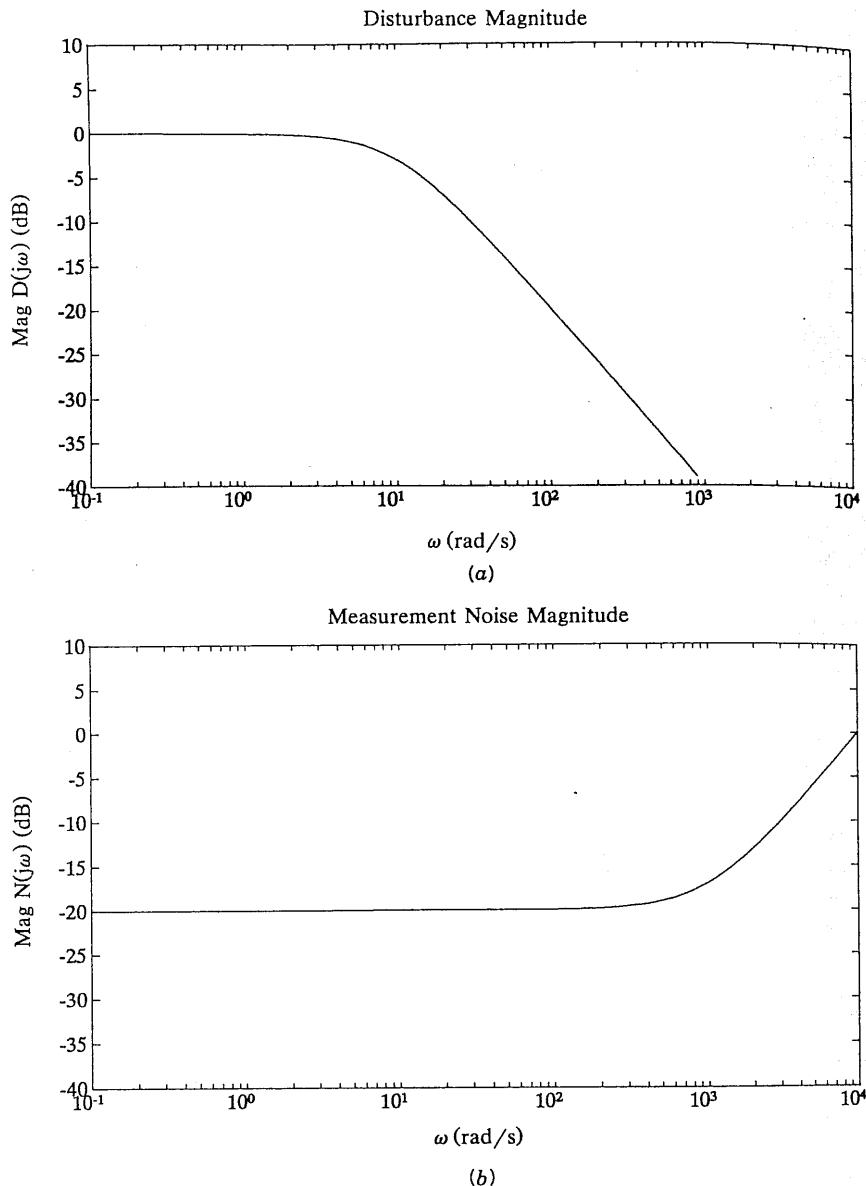


Fig. 6.2-2 Typical Bode plots for the uncertain signals in the system: (a) disturbance magnitude; (b) measurement noise magnitude.

Due to the presence of $n(t)$, $e(t)$ may not be symbolized in Fig. 6.2-1. The signal $s(t)$ is in fact given by

$$s(t) = r(t) - z(t) - n(t) = e(t) - n(t). \quad (6.2-2)$$

Let us perform a frequency-domain analysis on the system to see the effects of the uncertainties on system performance. In terms of Laplace transforms we may write

$$Z(s) = G(s)K(s)S(s) + D(s) \quad (6.2-3)$$

$$S(s) = R(s) - Z(s) - N(s) \quad (6.2-4)$$

$$E(s) = R(s) - Z(s). \quad (6.2-5)$$

Now we may solve for $Z(s)$ and $E(s)$, obtaining the closed-loop transfer function relations (see the problems)

$$Z(s) = (I + GK)^{-1}GK(R - N) + (I + GK)^{-1}D \quad (6.2-6)$$

$$E(s) = [I - (I + GK)^{-1}GK]R + (I + GK)^{-1}GKN - (I + GK)^{-1}D. \quad (6.2-7)$$

It is important to note that unlike the case for SISO systems, care must be taken to perform the matrix operations in the correct order; For instance, $GK \neq KG$. The multiplications by matrix inverses must also be performed in the correct order.

We can put these equations into a more convenient form. According to the matrix inversion lemma (Section 3.7), (6.2-7) may be written as

$$E(s) = (I + GK)^{-1}(R - D) + (I + GK)^{-1}GKN. \quad (6.2-8)$$

Moreover, since GK is square and invertible, we can write

$$\begin{aligned} (I + GK)^{-1}GK &= [(GK)^{-1}(I + GK)]^{-1} = [(GK)^{-1} + I]^{-1} \\ &= [(I + GK)(GK)^{-1}]^{-1} = GK(I + GK)^{-1}. \end{aligned} \quad (6.2-9)$$

Therefore, we may finally write $Z(s)$ and $E(s)$ as

$$Z(s) = GK(I + GK)^{-1}(R - N) + (I + GK)^{-1}D \quad (6.2-10)$$

$$E(s) = (I + GK)^{-1}(R - D) + GK(I + GK)^{-1}N. \quad (6.2-11)$$

To simplify things a bit, define the *system sensitivity*

$$S(s) = (I + GK)^{-1} \quad (6.2-12)$$

and

$$T(s) = GK(I + GK)^{-1} = (I + GK)^{-1}GK. \quad (6.2-13)$$

Since

$$S(s) + T(s) = (I + GK)(I + GK)^{-1} = I \quad (6.2-14)$$

we call $T(s)$ the *complementary sensitivity*, or in short, the *cosensitivity*. Note that the *return difference*

$$L(s) = I + GK \quad (6.2-15)$$

is the inverse of the sensitivity. The *loop gain* is given by $G(s)K(s)$.

These expressions extend the classical notions of loop gain, return difference, and sensitivity to multivariable systems. They are generally square transfer function matrices of dimension $q \times q$. In terms of these new quantities, we have

$$Z(s) = T(s)(R(s) - N(s)) + S(s)D(s) \quad (6.2-16)$$

$$E(s) = S(s)(R(s) - D(s)) + T(s)N(s). \quad (6.2-17)$$

According to the second equation, to ensure small tracking errors, we must have $S(j\omega)$ small at those frequencies ω where the reference input $r(t)$ and disturbance $d(t)$ are large. This will yield good *disturbance rejection*. On the other hand, for satisfactory *sensor noise rejection*, we should have $T(j\omega)$ small at those frequencies ω where $n(t)$ is large.

Unfortunately, a glance at (6.2-14) reveals that $S(j\omega)$ and $T(j\omega)$ cannot simultaneously be small at any one frequency ω . According to Fig. 6.2-2, we should like to have $S(j\omega)$ small at low frequencies, where $r(t)$ and $d(t)$ dominate, and $T(j\omega)$ small at high frequencies, where $n(t)$ dominates.

These are nothing but the multivariable generalizations of the well-known SISO classical notion that a large loop gain $GK(j\omega)$ is required at low frequencies for satisfactory performance and small errors, but a small loop gain is required at high frequencies where sensor noises are present.

Multivariable Bode Plot

These notions are not difficult to understand on a heuristic level. Unfortunately, it is not so straightforward to determine a clear measure for the "smallness" of $S(j\omega)$ and $T(j\omega)$. These are both square matrices of dimension $q \times q$, with q the number of performance outputs $z(t)$ and reference inputs $r(t)$. They are complex functions of the frequency. Clearly, the

classical notion of the Bode magnitude plot, which is defined only for *scalar* complex functions of ω , must be extended to the MIMO case.

Some work was done early on using the frequency-dependent eigenvalues of a square complex matrix as a measure of smallness [Rosenbrock, 1974; MacFarlane, 1970; MacFarlane and Kouvaritakis, 1977]. However, note that the matrix

$$M = \begin{bmatrix} 0.1 & 100 \\ 0 & 0.1 \end{bmatrix} \quad (6.2-18)$$

has large and small components, but its eigenvalues are both at 0.1.

A better measure of the magnitude of square matrices is the *singular value decomposition (SVD)* [Strang, 1980]. Given any matrix M we may write its *singular value decomposition (SVD)* as

$$M = U\Sigma V^*, \quad (6.2-19)$$

with $*$ denoting complex conjugate transpose, U and V square unitary matrices (i.e., $V^{-1} = V^*$, the complex conjugate transpose of V), and

$$\Sigma = \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \sigma_r & \\ & & & & 0 \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix} \quad (6.2-20)$$

with $r = \text{rank}(M)$. The singular values are the σ_i , which are ordered so that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$. The SVD may loosely be thought of as the extension to general matrices (which may be nonsquare or complex) of the Jordan form. If M is a function of $j\omega$, so are U , σ_i , and V .

Since $MM^* = U\Sigma V^*V\Sigma^T U^* = U\Sigma^2 U^*$, it follows that the singular values of M are simply the (positive) square roots of the nonzero eigenvalues of MM^* . A similar proof shows that the nonzero eigenvalues of MM^* and those of M^*M are the same.

We note that the M given above has two singular values, $\sigma_1 = 100.0001$ and $\sigma_2 = 0.0001$. Thus this measure indicates that M has a large and a small component. Indeed, note that

$$\begin{bmatrix} 0.1 & 100 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} -1 \\ 0.001 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.0001 \end{bmatrix} \quad (6.2-21)$$

while

$$\begin{bmatrix} 0.1 & 100 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} 0.001 \\ 1 \end{bmatrix} = \begin{bmatrix} 100.0001 \\ 0.1 \end{bmatrix}. \quad (6.2-22)$$

Thus the singular value σ_2 has the *input direction*

$$\begin{bmatrix} -1 \\ 0.001 \end{bmatrix}$$

associated with it for which the output contains the value σ_2 . On the other hand, the singular value σ_1 has an associated input direction of

$$\begin{bmatrix} 0.001 \\ 1 \end{bmatrix}$$

for which the output contains the value σ_1 .

There are many nice properties of the singular value which make it a suitable choice for defining the magnitude of matrix functions. Among these is the fact that the maximum singular value is an *induced matrix norm*, and norms have several useful attributes. The use of the SVs in the context of modern control was explored in Doyle and Stein [1981] and Safonov et al. [1981].

A major factor is that there are many good software packages which have good routines for computing the singular value (e.g., subroutine LSVDF in IMSL [1980] or Moler et al. [1987]). Thus, plots like those we shall present may easily be obtained by writing only a computer program to drive the available subroutines. Indeed, since the SVD uses unitary matrices, its computation is numerically stable. An efficient technique for obtaining the SVs of a complex matrix as a function of frequency ω is given in Laub [1981].

We note that a complete picture of the behavior of a complex matrix versus ω must take into account the magnitudes of the SVs as well as the *multivariable phase*, which may also be obtained from the SVD [Postlethwaite et al., 1981]. Thus, complete MIMO generalizations of the Bode magnitude and phase plots are available. However, the theory relating to the phase portion of the plot is more difficult to use in a practical design technique, although a MIMO generalization of the Bode gain-phase relation is available [Doyle and Stein, 1981]. Therefore, we shall only employ plots of the SVs versus frequency, which correspond to the Bode magnitude plot for MIMO systems.

The magnitude of a square transfer function matrix $H(j\omega)$ at any frequency ω depends on the direction of the input excitation. Inputs in a certain direction in the input space will excite only the SV(s) associated with that direction. However, for any input, the magnitude of the transfer function $H(j\omega)$ at any given frequency ω may be bounded above by its *maximum singular value*, denoted $\bar{\sigma}(H(j\omega))$, and below by its *minimum singular value*, denoted $\underline{\sigma}(H(j\omega))$. Therefore, all our results, as well as the plots we shall give, need take into account only these two bounding values of "magnitude."

Example 6.2-1: MIMO Bode Magnitude Plots. Here, we consider a simple nonaircraft system to make some points about the singular value plots. Consider the multivariable system

$$\dot{x} = \begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -2 & 6 \\ 0 & 0 & -6 & -2 \end{bmatrix}x + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}u = Ax + Bu \quad (1)$$

$$z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}x = Hx, \quad (2)$$

which as a 2×2 MIMO transfer function of

$$H(s) = H(sI - A)^{-1}B = \frac{M(s)}{\Delta(s)} \quad (3)$$

with

$$\Delta(s) = s^4 + 6s^3 + 50s^2 + 88s + 80$$

$$M(s) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}s^3 + \begin{bmatrix} 5 & 0 \\ 0 & 4 \end{bmatrix}s^2 + \begin{bmatrix} 44 & 0 \\ 0 & 6 \end{bmatrix}s + \begin{bmatrix} 40 & 0 \\ 0 & 4 \end{bmatrix}. \quad (4)$$

By writing a driver program that calls standard software (e.g., subroutine LSVDF in [IMSL, 1980]) to evaluate the SVs at closely spaced values of frequency ω , we may obtain the SV plots versus frequency shown in

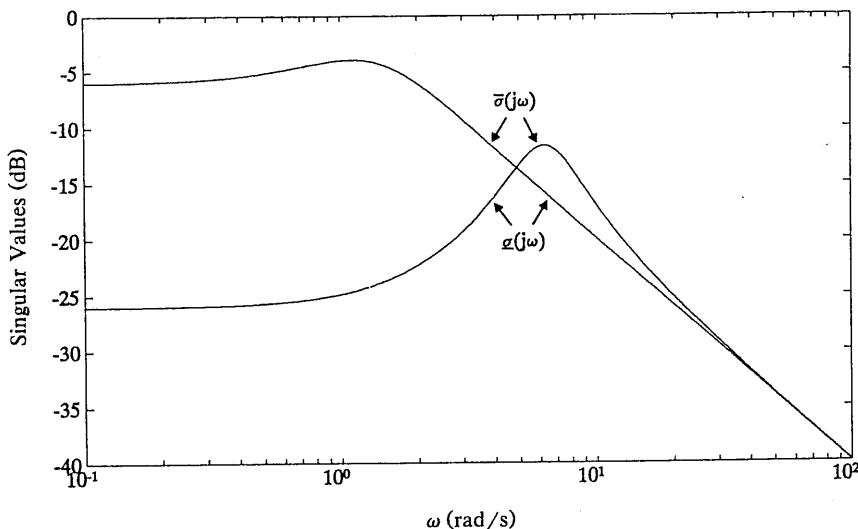


Fig. 6.2-3 MIMO Bode magnitude plot of singular values versus frequency.

Fig. 6.2-3. We call this the *multivariable Bode magnitude plot* for the MIMO transfer function $H(s)$.

Since $H(s)$ is 2×2 , it has two singular values. Note that although each singular value is continuous, the maximum and minimum singular values are not. This is due to the fact that the singular values can cross over each other, as the figure illustrates. ■

Example 6.2-2: Singular Value Plots for F-16 Lateral Dynamics. To illustrate the difference between the singular-value plots and the individual SISO Bode plots of a multivariable system, let us consider the F-16 lateral dynamics of Examples 5.3-1 and 5.5-4. In the latter example, we designed a wing leveler. For convenience, refer to the figure there showing the control system structure. Using the system matrices A and B in that example, which include an integrator in the ϕ channel as well as actuator dynamics and a washout filter, take as the control inputs $u = [u_a \ u_r]^T$, with u_a the aileron servo input and u_r the rudder servo input. Select as outputs $z = [\epsilon \ r_w]^T$, with ϵ the integrator output in the ϕ channel and r_w the washed-out yaw rate.

The individual SISO transfer functions in this two-input/two-output open-loop system are

$$H_{11} = \frac{\epsilon}{u_a} = \frac{14.8}{s(s + 0.0163)(s + 3.615)(s + 20.2)} \quad (1)$$

$$\begin{aligned} H_{12} &= \frac{r_w}{u_a} \\ &= \frac{-36.9s(s + 2.237)[(s + 0.55)^2 + 2.49^2]}{(s + 0.0163)(s + 1)(s + 3.615)(s + 20.2)[(s + 0.4225)^2 + 3.063^2]} \end{aligned} \quad (2)$$

$$H_{21} = \frac{\epsilon}{u_r} = \frac{-2.65(s + 2.573)(s - 2.283)}{s(s + 0.0163)(s + 3.615)(s + 20.2)[(s + 0.4225)^2 + 3.063^2]} \quad (3)$$

$$H_{22} = \frac{r_w}{u_r} = \frac{-0.718s[(s + 0.139)^2 + 0.446^2]}{(s + 0.0163)(s + 1)(s + 20.2)[(s + 0.4225)^2 + 3.063^2]}. \quad (4)$$

The standard Bode magnitude plots for these SISO transfer functions are shown in Fig. 6.2-4. Clearly visible are the resonance due to the dutch roll mode, as well as the integrator in the upper ϕ channel in the figure in Example 5.5-4.

On the other hand, shown in Fig. 6.2-5 are the singular values of this multivariable system. Note that it is not immediately evident how they relate to the SISO plots in Fig. 6.2-4. In the next section we shall see that bounds

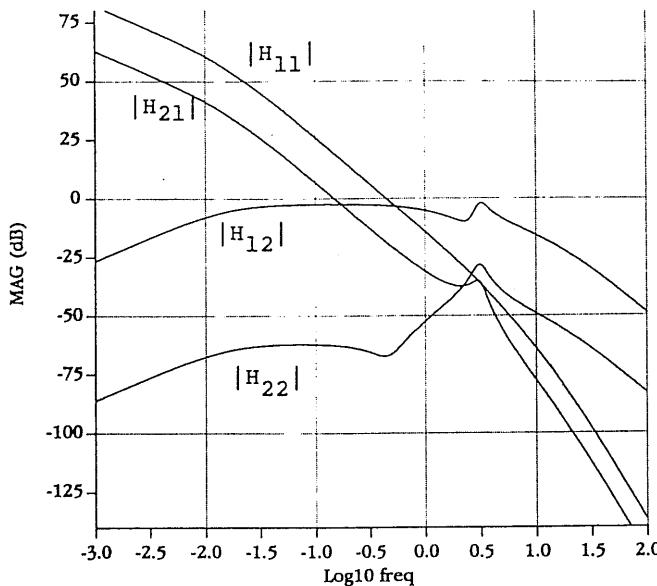


Fig. 6.2-4 SISO Bode magnitude plots for F-16 lateral dynamics.

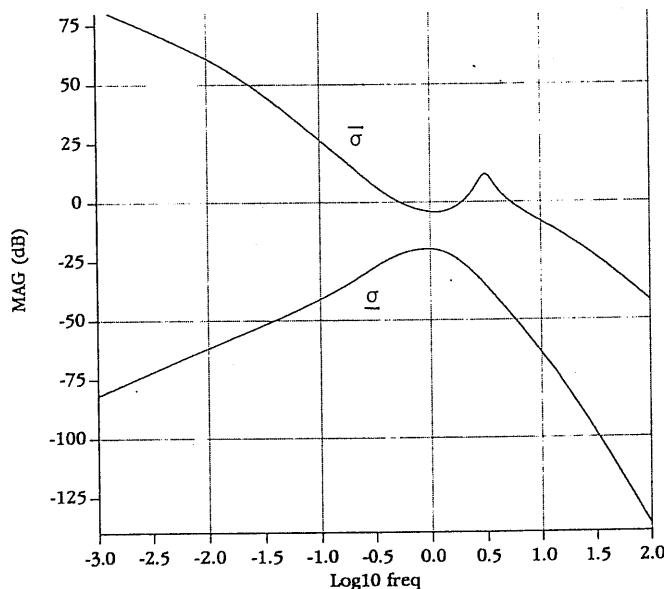


Fig. 6.2-5 Singular values for F-16 lateral dynamics.

for guaranteed robustness are given for MIMO systems in terms of the minimum singular value being large at low frequencies (for performance robustness) and the maximum singular value being small at high frequencies (for stability robustness). The lack of any clear correspondence between Figs. 6.2-4 and 6.2-5 shows that these bounds cannot be expressed in terms of the individual SISO Bode plots. ■

Frequency-Domain Performance Specifications

We have seen how to make a multivariable Bode magnitude plot of a square transfer function matrix. It is now necessary to discuss *performance specifications* in the frequency domain in order to determine what a "desirable" Bode plot means in the MIMO case. The important point is that the low-frequency requirements are generally in terms of the *minimum* singular value being *large*, while the high-frequency requirements are in terms of the *maximum* singular value being *small*.

First, let us point out that the classical notion of *bandwidth* holds in the MIMO case. This is the frequency ω_c for which the loop gain $GK(j\omega)$ passes through a value of 1, or 0 dB. If the bandwidth should be limited due to high-frequency noise considerations, the *largest* SV should satisfy $\bar{\sigma}(GK(j\omega_c)) = 1$, at the specified cutoff frequency ω_c .

L_2 Operator Gain. To relate frequency-domain behavior to time-domain behavior, we may take into account the following considerations [Morari and Zafriou, 1989]. Define the L_2 norm of a vector time function $s(t)$ by

$$\|s\|_2 = \left[\int_0^\infty s^T(t) s(t) dt \right]^{1/2}. \quad (6.2-23)$$

This is related to the total energy in $s(t)$ and should be compared to the LQ performance index.

A linear time-invariant system has input $u(t)$ and output $z(t)$ related by the convolution integral

$$z(t) = \int_{-\infty}^{\infty} h(t - \tau) u(\tau) d\tau, \quad (6.2-24)$$

with $h(t)$ the impulse response. The L_2 operator gain, denoted $\|H\|_2$, of such a system is defined as the smallest value of γ such that

$$\|z\|_2 \leq \gamma \|u\|_2. \quad (6.2-25)$$

This is just the operator norm induced by the L_2 vector norm. An important

result is that the L_2 operator gain is given by

$$\|H\|_2 = \max_{\omega} [\bar{\sigma}(H(j\omega))] \quad (6.2-26)$$

with $H(s)$ the system transfer function. That is, $\|H\|_2$ is nothing but the *maximum value* over ω of the maximum singular value of $H(j\omega)$. Thus $\|H\|_2$ is an *H -infinity norm* in the frequency domain.

This result gives increased importance to $\bar{\sigma}(H(j\omega))$, for if we are interested in keeping $z(t)$ small over a range of frequencies, we should take care that $\bar{\sigma}(H(j\omega))$ is small over that range.

It is now necessary to see how this result may be used in deriving frequency-domain performance specifications. Some facts we shall use in this discussion are

$$\underline{\sigma}(GK) - 1 \leq \underline{\sigma}(I + GK) \leq \underline{\sigma}(GK) + 1 \quad (6.2-27)$$

$$\bar{\sigma}(M) = \frac{1}{\underline{\sigma}(M^{-1})}, \quad (6.2-28)$$

$$\bar{\sigma}(AB) \leq \bar{\sigma}(A)\bar{\sigma}(B) \quad (6.2-29)$$

for any matrices A, B, GK, M , with M nonsingular.

Before we begin a discussion of performance specifications, let us note the following. If $S(j\omega)$ is small, as desired at low frequencies, then

$$\bar{\sigma}(S) = \bar{\sigma}\left[(I + GK)^{-1}\right] = \frac{1}{\underline{\sigma}(I + GK)} \approx \frac{1}{\underline{\sigma}(GK)}. \quad (6.2-30)$$

That is, a large value of $\underline{\sigma}(GK)$ guarantees a small value of $\bar{\sigma}(S)$.

On the other hand, if $T(j\omega)$ is small, as is desired at high frequencies, then

$$\bar{\sigma}(T) = \bar{\sigma}\left[GK(I + GK)^{-1}\right] \approx \bar{\sigma}(GK). \quad (6.2-31)$$

That is, a small value of $\bar{\sigma}(GK)$ guarantees a small value of $\bar{\sigma}(T)$.

This means that specifications that $S(j\omega)$ be small at low frequencies and $T(j\omega)$ be small at high frequencies may equally well be formulated in terms of $\underline{\sigma}(GK)$ being large at low frequencies and $\bar{\sigma}(GK)$ being small at high frequencies. Thus all of our performance specifications will be in terms of the *minimum and maximum SVs of the loop gain $GK(j\omega)$* . The practical significance of this is that we need only compute the SVs of $GK(j\omega)$, not those of $S(j\omega)$ and $T(j\omega)$. These notions are symbolized in Fig. 6.2-6, where it should be recalled that $S + T = I$.

Now, we shall first consider low-frequency specifications on the singular-value plot, and then high-frequency specifications. According to our

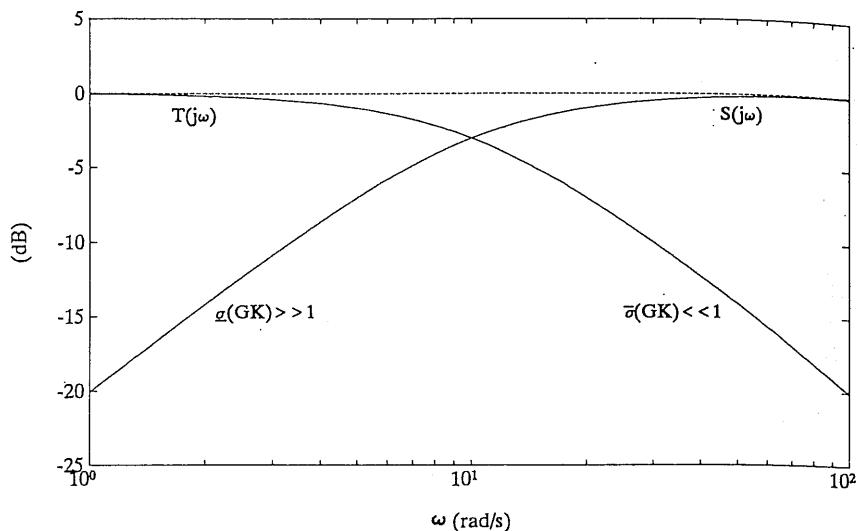


Fig. 6.2-6 Magnitude specifications on $S(j\omega)$, $T(j\omega)$, and $GK(j\omega)$.

discussion relating to (6.2-17), the former will involve the reference input $r(t)$ and disturbances $d(t)$, while the latter will involve the sensor noise $n(t)$.

Low-Frequency Specifications. For low frequencies let us suppose that the sensor noise $n(t)$ is zero so that (6.2-17) becomes

$$E(s) = S(s)(R(s) - D(s)). \quad (6.2-32)$$

Thus, to keep $\|e(t)\|_2$ small, it is only necessary to ensure that the L_2 operator norm $\|S\|_2$ is small at all frequencies where $R(j\omega)$ and $D(j\omega)$ are appreciable. This may be achieved by ensuring that, at such frequencies, $\bar{\sigma}(S(j\omega))$ is small. As we have just seen, this may be guaranteed if we select

$$\underline{\sigma}(GK(j\omega)) \gg 1 \quad \text{for } \omega \leq \omega_d, \quad (6.2-33)$$

where $D(s)$ and $R(s)$ are appreciable for $\omega \leq \omega_d$.

Thus, exactly as in the classical case [Franklin et al., 1986], we are able to specify a low-frequency performance bound that guarantees *performance robustness*; that is, good performance in the face of low-frequency disturbances. For instance, to ensure that disturbances are attenuated by a factor of 0.01, we should ensure $\underline{\sigma}(GK(j\omega))$ is greater than 40 dB at low frequencies $\omega \leq \omega_d$.

At this point it is worth examining Fig. 6.2-9, which illustrates the frequency-domain performance specifications we are beginning to derive.

Another low-frequency performance bound may be derived from steady-state error considerations. Thus, suppose that $d(t) = 0$ and the reference input is a unit step of magnitude r so that $R(s) = r/s$. Then, according to (6.2-32) and the final value theorem [Franklin et al., 1986], the steady-state error e_∞ is given by

$$e_\infty = \lim_{s \rightarrow 0} sE(s) = rS(0). \quad (6.2-34)$$

To ensure that the largest component of e_∞ is less than a prescribed small acceptable value δ_∞ , we should therefore select

$$\underline{\sigma}(GK(0)) > \frac{r}{\delta_\infty}. \quad (6.2-35)$$

The ultimate objective of all our concerns is to manufacture a compensator $K(s)$ in Fig. 6.2-1 that gives desirable performance. Let us now mention two low-frequency considerations that are important in the initial stages of the design of the compensator $K(s)$.

To make the steady-state error in response to a unit step at $r(t)$ exactly equal to zero, we may ensure that there is an integrator *in each path* of the system $G(s)$ so that it is of type 1 [Franklin et al., 1986]. Thus, suppose that the system to be controlled is given by

$$\begin{aligned} \dot{x} &= Ax + Bu \\ z &= Hx. \end{aligned} \quad (6.2-36)$$

To add an integrator to each control path, we may augment the dynamics so that

$$\frac{d}{dt} \begin{bmatrix} x \\ \epsilon \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \epsilon \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u, \quad (6.2-37)$$

with ϵ the integrator outputs (see Fig. 6.2-7). The system $G(s)$ in Fig. 6.2-1 should now be taken as (6.2-37), which contains the integrators as a precompensator.

Although augmenting each control path with an integrator results in zero steady-state error, in some applications this may result in an unnecessarily complicated compensator. Note that the steady-state error may be made as small as desired without integrators by selecting $K(s)$ so that (6.2-35) holds.

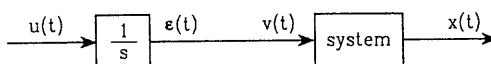


Fig. 6.2-7 Plant augmented with integrators.

A final concern on the low-frequency behavior of $G(s)$ needs to be addressed. It is desirable in many situations to have $\underline{\sigma}(GK)$ and $\bar{\sigma}(GK)$ close to the same value. Then the speed of the responses will be nearly the same in all channels of the system. This is called the issue of *balancing the singular values at low frequency*. The SVs of $G(s)$ in Fig. 6.2-1 may be balanced at low frequencies, as follows.

Suppose that the plant has the state-variable description (6.2-36), and let us add a square constant precompensator gain matrix P , so that

$$v = Pu \quad (6.2-38)$$

is the relation between the control input $u(t)$ in Fig. 6.2-1 and the actual plant input $v(t)$. The transfer function of the plant plus precompensator is now

$$G(s) = H(sI - A)^{-1}BP. \quad (6.2-39)$$

As s goes to zero, this approaches

$$G(0) = H(-A)^{-1}BP,$$

as long as A has no poles at the origin. Therefore, we may ensure that $G(0)$ has all SVs equal to a prescribed value of γ by selecting

$$P = \gamma [H(-A)^{-1}B]^{-1}, \quad (6.2-40)$$

for then $G(0) = \gamma I$.

The transfer function of (6.2-36) is

$$H(s) = H(sI - A)^{-1}B, \quad (6.2-41)$$

whence we see that the required value of the precompensator gain is

$$P = \gamma H^{-1}(0). \quad (6.2-42)$$

This is nothing but the (scaled) reciprocal dc gain.

Example 6.2-3: Precompensator for Balancing and Zero Steady-State Error. Let us design a precompensator for the system in Example 6.2-1 using the notions just discussed. Substituting the values of A , B , and H in (6.2-40) with $\gamma = 1$ yields

$$P = [H(-A)^{-1}B]^{-1} = \begin{bmatrix} 2 & 0 \\ 0 & 20 \end{bmatrix}. \quad (1)$$

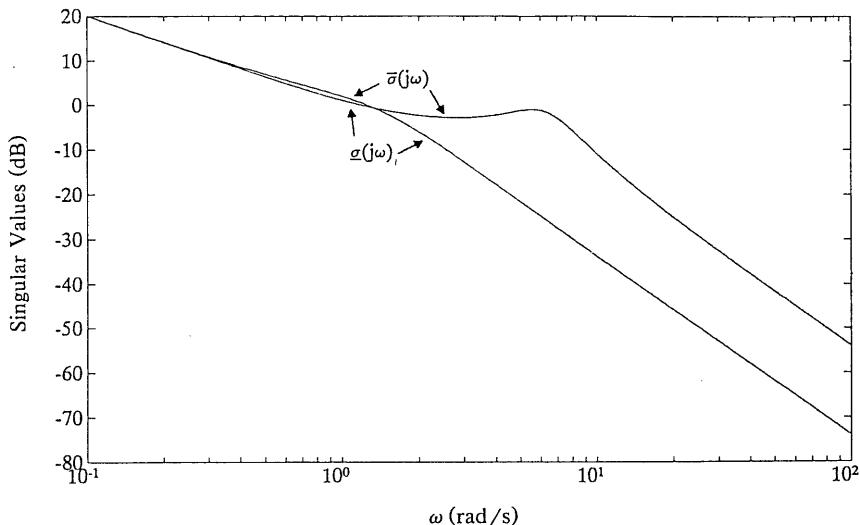


Fig. 6.2-8 MIMO Bode magnitude plot for augmented plant.

To ensure zero-steady-state error as well as equal singular values at low frequencies, we may incorporate integrators in each input channel along with the gain matrix P by writing the augmented system

$$\frac{d}{dt} \begin{bmatrix} x \\ \epsilon \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \epsilon \end{bmatrix} + \begin{bmatrix} 0 \\ P \end{bmatrix} u. \quad (2)$$

The singular-value plots for this plant plus precompensator appear in Fig. 6.2-8. At low frequencies there is now a slope of -20 dB/decade as well as equality of $\underline{\sigma}$ and $\bar{\sigma}$. Thus the augmented system is both balanced and of Type 1. Compare Fig. 6.2-8 to the singular-value plot of the uncompensated system in Fig. 6.2-3. The remaining step is the selection of the feedback gain matrix for the augmented plant (2) so that the desired performance is achieved. ■

High-Frequency Specifications. We now turn to a discussion of high-frequency performance specifications. The sensor noise is generally appreciable at frequencies above some known value ω_n (see Fig. 6.2-2). Thus, according to (6.2-17), to keep the tracking error norm $\|e\|_2$ small in the face of measurement noise we should ensure that the operator norm $\|T\|_2$ is small at high frequencies above this value. By (6.2-31) this may be guaranteed if

$$\bar{\sigma}(GK(j\omega)) \ll 1 \quad \text{for } \omega \geq \omega_n. \quad (6.2-43)$$

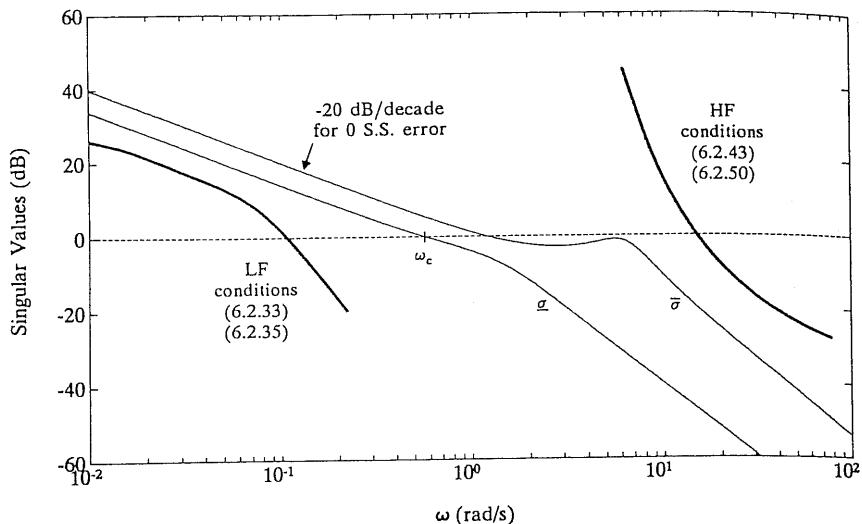


Fig. 6.2-9 Frequency-domain performance specifications.

(see Fig. 6.2-9). For instance, to ensure that sensor noise is attenuated by a factor of 0.1, we should guarantee that $\bar{\sigma}(GK(j\omega)) < -20$ dB for $\omega \geq \omega_n$.

One final high-frequency robustness consideration needs to be mentioned. It is unusual for the plant model to be exactly known. There are two basic sorts of modeling inaccuracies that concern us in aircraft controls. The first is plant parameter variation due to changes in the linearization equilibrium point of the nonlinear model. This is a low-frequency phenomenon and will be discussed in the next subsection. The second sort of inaccuracy is due to unmodeled high-frequency dynamics; this we discuss here.

We are assuming a rigid-body aircraft model for the purpose of controls design, and in so doing are neglecting flexible and vibrational modes at high frequencies. Thus although our design may guarantee closed-loop stability for the *assumed mathematical model* $G(s)$, stability is not assured for the *actual plant* $G'(s)$ with flexible modes. To guarantee *stability robustness* in the face of plant parameter uncertainty, we may proceed as follows.

The model uncertainties may be of two types. The actual plant model G' and the assumed plant model G may differ by *additive uncertainties* so that

$$G'(j\omega) = G(j\omega) + \Delta G(j\omega), \quad (6.2-44)$$

where the unknown discrepancy satisfies a known bound

$$\bar{\sigma}(\Delta G(j\omega)) < a(\omega), \quad (6.2-45)$$

with $a(\omega)$ known for all ω .

On the other hand, the actual plant model $G'(s)$ and the assumed plant model $G(s)$ may differ by *multiplicative uncertainties* so that

$$G'(j\omega) = [I + M(j\omega)]G(j\omega), \quad (6.2-46)$$

where the unknown discrepancy satisfies a known bound

$$\bar{\sigma}(M(j\omega)) < m(\omega), \quad (6.2-47)$$

with $m(\omega)$ known for all ω . We shall show several ways of finding the bound $m(\omega)$. In Example 6.2-4 we show how to construct a reduced-order model for the system, which may then be used for controls design. There $m(\omega)$ is determined from the neglected dynamics. In Example 6.3-1 we show how $m(\omega)$ may be determined in terms of the aircraft's neglected flexible modes. In the next subsection we show how to determine $m(\omega)$ in terms of plant parameter variations in the linearized model due to operating point changes.

Since we may write (6.2-44) as

$$G'(j\omega) = [I + \Delta G(j\omega)G^{-1}(j\omega)]G(j\omega) \equiv [I + M(j\omega)]G(j\omega), \quad (6.2-48)$$

we shall confine ourselves to a discussion of multiplicative uncertainties, following [Doyle and Stein, 1981].

Suppose that we have designed a compensator $K(s)$ so that the closed-loop system in Fig. 6.2-1 is stable. We should now like to derive a frequency-domain condition that guarantees the stability of the *actual* closed-loop system, which contains not $G(s)$, but $G'(s)$ satisfying (6.2-46)/(6.2-47). For this, the multivariable Nyquist condition [Rosenbrock, 1974] may be used.

Thus it is required that the encirclement count of the map $|I + G'K|$ be equal to the negative number of unstable open-loop poles of $G'K$. By assumption, this number is the same as that of GK . Thus the number of encirclements of $|I + G'K|$ must remain unchanged for all G' allowed by (6.2-47). This is assured if and only if $|I + G'K|$ remains nonzero as G is warped continuously toward G' , or equivalently,

$$0 < \underline{\sigma}[I + [I + \epsilon M(s)]G(s)K(s)]$$

for all $0 \leq \epsilon \leq 1$, all $M(s)$ satisfying (6.2-47), and all s on the standard Nyquist contour.

Since G' vanishes on the infinite radius segment of the Nyquist contour, and assuming for simplicity that no indentations are required along the $j\omega$ -axis portion, this reduces to the following equivalent conditions:

$$0 < \underline{\sigma}[I + G(j\omega)K(j\omega) + \epsilon M(j\omega)G(j\omega)K(j\omega)]$$

for all $0 \leq \epsilon \leq 1$, $0 \leq \omega < \infty$, all M ,

$$\text{iff } 0 < \underline{\sigma} \left[\{I + \epsilon MGK(I + GK)^{-1}\}(I + GK) \right]$$

$$\text{iff } 0 < \underline{\sigma} \left[I + MGK(I + GK)^{-1} \right]$$

all $0 \leq \omega < \infty$, and all M ,

$$\text{iff } \bar{\sigma} \left[GK(I + GK)^{-1} \right] < \frac{1}{m(\omega)} \quad (6.2-49)$$

for all $0 \leq \omega < \infty$. Thus stability robustness translates into a requirement that the cosensitivity $T(j\omega)$ be bounded above by the reciprocal of the multiplicative modeling discrepancy bound $m(\omega)$.

In the case of high-frequency unmodeled dynamics, $1/m(\omega)$ is small at high ω , so that according to (6.2-31), we may simplify (6.2-49) by writing it in terms of the loop gain as

$$\bar{\sigma}(GK(j\omega)) < \frac{1}{m(\omega)} \quad (6.2-50)$$

for all ω such that $m(\omega) \gg 1$.

This bound for stability robustness is illustrated in Fig. 6.2-9.

An example will be useful at this point.

Example 6.2-4: Model Reduction and Stability Robustness. In some situations we have a high-order aircraft model that is inconvenient to use for controller design. Examples occur in engine control and spacecraft control. In such situations, it is possible to compute a reduced-order model of the system which may then be used for controller design. Here we shall show a convenient technique for model reduction as well as an illustration of the stability robustness bound $m(\omega)$. The technique described here is from [Athans et al., 1986].

a. *Model Reduction by Partial-Fraction Expansion.* Suppose that the actual plant is described by

$$\dot{x} = Ax + Bu \quad (1a)$$

$$z = Hx. \quad (1b)$$

with $x \in \mathbf{R}^n$. If A is simple with eigenvalues λ_i , right eigenvectors u_i , and left eigenvectors v_i so that

$$Au_i = \lambda_i u_i, \quad v_i^T A = \lambda_i v_i^T, \quad (2)$$

then the transfer function

$$G'(s) = H(sI - A)^{-1}B \quad (3)$$

may be written as the partial-fraction expansion (Section 5.2)

$$G'(s) = \sum_{i=1}^n \frac{R_i}{s - \lambda_i}, \quad (4)$$

with residue matrices given by

$$R_i = Hu_i v_i^T B. \quad (5)$$

If the value of n is large, it may be desirable to find a *reduced-order approximation* to (1) for which a simplified compensator $K(s)$ in Fig. 6.2-1 may be designed. Then, if the approximation is a good one, the compensator $K(s)$ should work well when used on the actual plant $G'(s)$.

To find a reduced-order approximation $G(s)$ to the plant, we may proceed as follows. Decide which of the eigenvalues λ_i in (4) are to be retained in $G(s)$. This may be done using engineering judgment, by omitting high-frequency modes, by omitting terms in (4) that have small residues, and so on. Let the r eigenvalues to be retained in $G(s)$ be $\lambda_1, \lambda_2, \dots, \lambda_r$.

Define the matrix

$$Q = \text{diag}\{Q_i\}, \quad (6)$$

where Q is an $r \times r$ matrix and the blocks Q_i are defined as

$$Q_i = \begin{cases} 1, & \text{for each real eigenvalue retained} \\ \begin{bmatrix} \frac{1}{2} & -\frac{j}{2} \\ \frac{1}{2} & \frac{j}{2} \end{bmatrix}, & \text{for each complex pair retained.} \end{cases} \quad (7)$$

Compute the matrices

$$V \equiv Q^{-1} \begin{bmatrix} v_i^T \\ \vdots \\ v_r^T \end{bmatrix} \quad (8)$$

$$U \equiv [u_1 \quad \cdots \quad u_r]Q. \quad (9)$$

In terms of these constructions, the reduced-order system is nothing but a projection of (1) onto a space of dimension r with state defined by

$$w = Vx. \quad (10)$$

The system matrices in the reduced-order approximate system

$$\dot{w} = Fw + Gu \quad (11a)$$

$$z = Jw + Du \quad (11b)$$

are given by

$$\begin{aligned} F &= VAU \\ G &= VB \\ J &= HU, \end{aligned} \quad (12)$$

with the direct-feed matrix given in terms of the residues of the neglected eigenvalues as

$$D = \sum_{i=r+1}^n -\frac{R_i}{\lambda_i}. \quad (13)$$

The motivation for selecting such a D matrix is as follows. The transfer function

$$G(s) = J(sI - F)^{-1}G + D$$

of the reduced system (11) is given as (verify!)

$$G(s) = \sum_{i=1}^r \frac{R_i}{s - \lambda_i} + \sum_{i=r+1}^n -\frac{R_i}{\lambda_i}. \quad (14)$$

Evaluating $G(j\omega)$ and $G'(j\omega)$ at $\omega = 0$, it is seen that they are equal at dc. Thus the modeling errors induced by taking $G(s)$ instead of the actual $G'(s)$ occur at higher frequencies. Indeed, they depend on the frequencies of the neglected eigenvalues of (1).

To determine the $M(s)$ in (6.2-46) that is induced by the order reduction, note that

$$G' = (I + M)G \quad (15)$$

so that

$$M = (G' - G)G^{-1} \quad (16)$$

or

$$M(s) = \left[\sum_{i=r+1}^n \frac{R_i}{\lambda_i} \frac{s}{s - \lambda_i} \right] G^{-1}(s). \quad (17)$$

Then the high-frequency robustness bound is given in terms of

$$m(j\omega) = \bar{\sigma}(M(j\omega)). \quad (18)$$

Note that $M(j\omega)$ tends to zero as ω becomes small, reflecting our perfect certainty of the actual plant at dc.

b. An Example. Let us take an example to illustrate the model-reduction procedure, and show also how to compute the upper bound $m(\omega)$ in (6.2-46)/(6.2-47) on the high-frequency modeling errors thereby induced. To make it easy to see what is going on, we shall take a Jordan-form system.

Let there be prescribed the MIMO system

$$\dot{x} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -10 \end{bmatrix}x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2 & 0 \end{bmatrix}u = Ax + Bu \quad (19a)$$

$$z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}x = Cx. \quad (19b)$$

The eigenvectors are given by $u_i = e_i$, $v_i = e_i$, $i = 1, 2, 3$, with e_i the i th column of the 3×3 identity matrix. Thus the transfer function is given by the partial-fraction expansion

$$G'(s) = \frac{R_1}{s+1} + \frac{R_2}{s+2} + \frac{R_3}{s+10}, \quad (20)$$

with

$$R_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_3 = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}. \quad (21)$$

To find the reduced-order system that retains the poles at $\lambda = -1$ and $\lambda = -2$, define

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (22)$$

and compute the approximate system

$$\dot{w} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}w + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}u = Fw + Gu \quad (23a)$$

$$z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}w + \begin{bmatrix} 0 & 0 \\ 0.2 & 0 \end{bmatrix}u = Jw + Du. \quad (23b)$$

This has a transfer function of

$$G(s) = \frac{R_1}{s+1} + \frac{R_2}{s+2} + D. \quad (24)$$

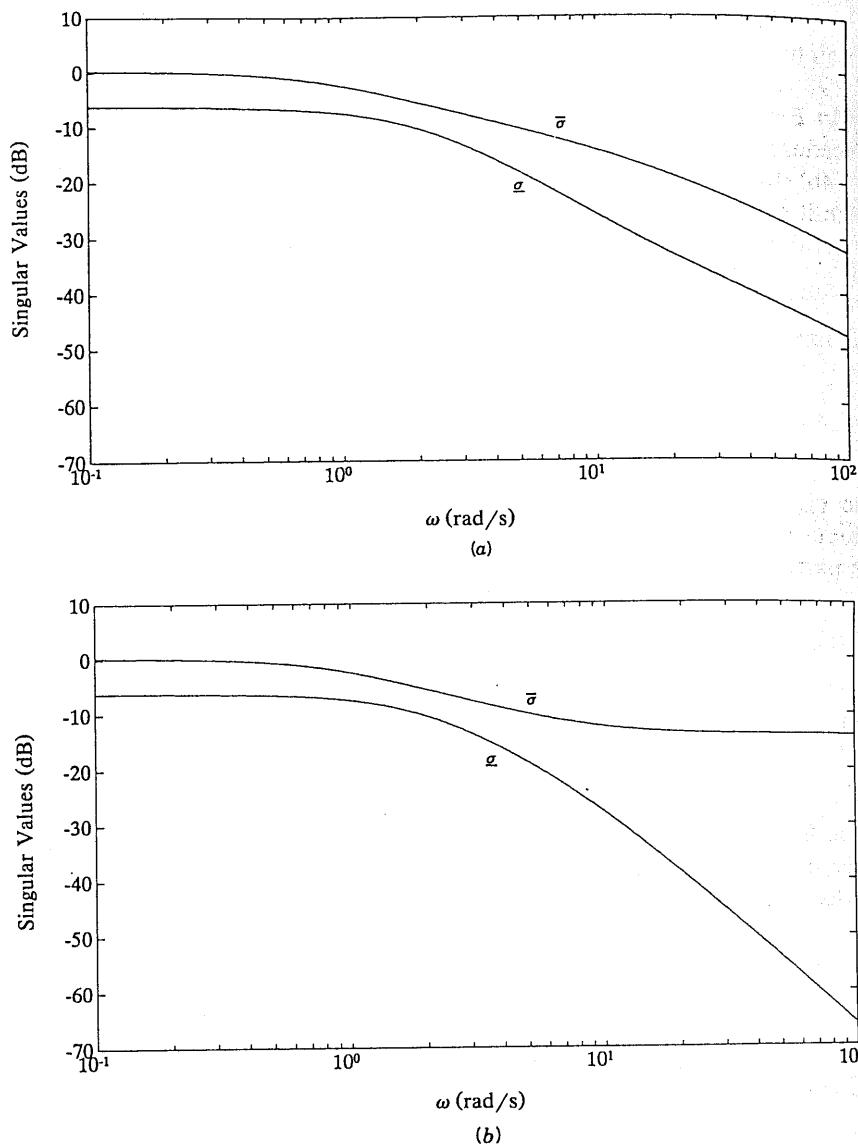


Fig. 6.2-10 MIMO Bode magnitude plots of singular values: (a) actual plant; (b) reduced-order approximation.

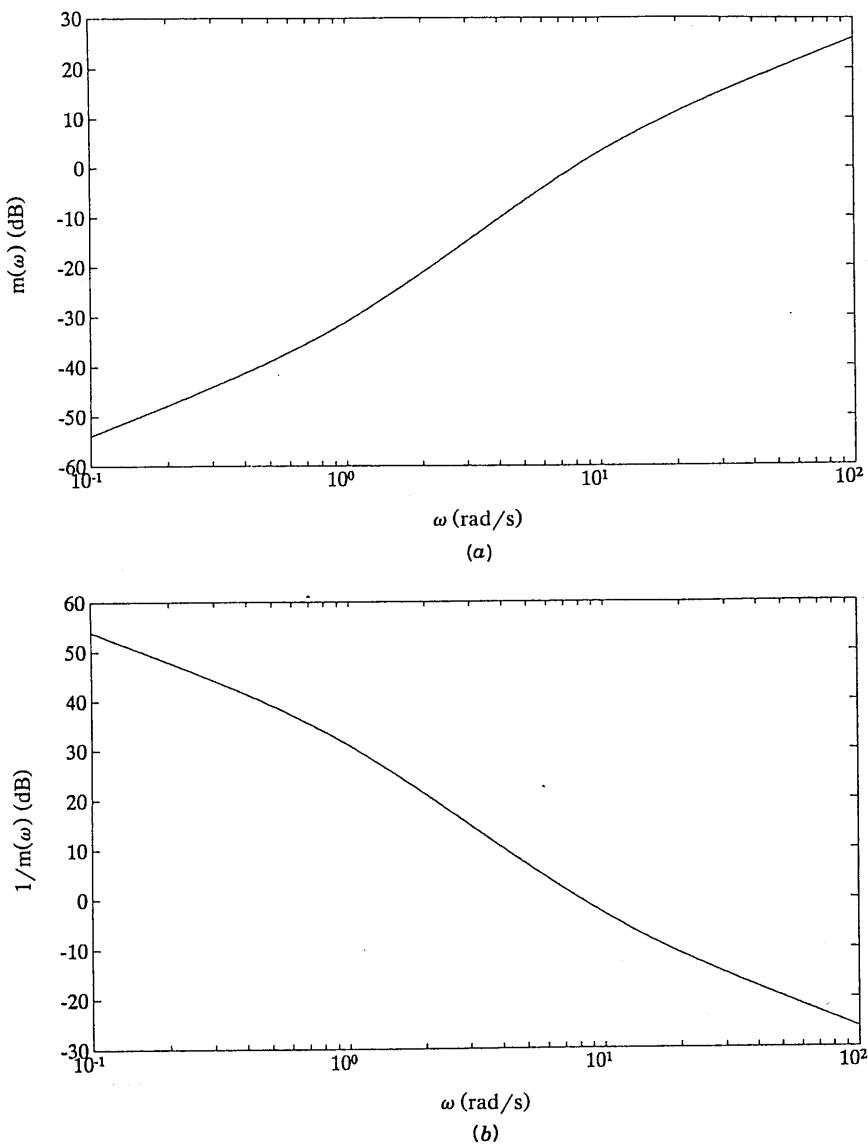


Fig. 6.2-11 High-frequency stability-robustness bound: (a) $m(\omega)$; (b) $1/m(\omega)$.

Singular-value plots of the actual plant (19) and the reduced-order approximation (23) are shown in Fig. 6.2-10.

The multiplicative error is given by

$$M = (G' - G)G^{-1} = \begin{bmatrix} 0 & 0 \\ -\frac{0.2s(s+1)}{s+10} & 0 \end{bmatrix}, \quad (25)$$

whence

$$m(\omega) = \bar{\sigma}(M(j\omega)) = \frac{0.2\omega\sqrt{\omega^2 + 1}}{\sqrt{\omega^2 + 100}}, \quad (26)$$

and the high-frequency bound on the loop gain $GK(j\omega)$ is given by

$$\frac{1}{m(j\omega)} = \frac{5\sqrt{\omega^2 + 100}}{\omega\sqrt{\omega^2 + 1}}. \quad (27)$$

This bound is plotted in Fig. 6.2-11. Note that the modeling errors become appreciable (i.e., of magnitude 1) at a frequency of 8.0 rad/s. Above this frequency, we should ensure that constraint (6.2-50) on the loop-gain magnitude holds to guarantee stability-robustness. This will be a restriction on any compensator $K(s)$ designed using the reduced-order plant (23). ■

Robustness Bounds for Plant Parameter Variations

The aircraft is nonlinear, but for controller design we use linearized models obtained at some operating point. In practice, it is necessary to determine linear models at several design operating points over a specified flight envelope, and determine optimal control gains for each one. Then these design control gains are tabulated and scheduled using microprocessors, so that the gains most appropriate for the actual operating point of the aircraft are used in the controller. It is usual to determine which of the design operating points are closest to the actual operating point and use some sort of linear combination of the control gains corresponding to these design points.

It is important for the control gains to stabilize the aircraft at all points near the design operating point for this gain scheduling procedure to be effective. In passing from operating point to operating point, the parameters of the state-variable model vary. Using (6.2-49), we may design controllers that guarantee robust stability despite plant parameter variations.

Suppose that the nominal perturbed model used for design is

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (6.2-51)$$

which has the transfer function

$$G(s) = C(sI - A)^{-1}B. \quad (6.2-52)$$

However, due to operating point changes the actual aircraft perturbed motion is described by

$$\begin{aligned}\dot{x} &= (A + \Delta A)x + (B + \Delta B)u \\ y &= (C + \Delta C)x,\end{aligned} \quad (6.2-53)$$

where the plant parameter variation matrices are $\Delta A, \Delta B, \Delta C$. It is not difficult to show (see Stevens et al. [1987] and the problems) that this results in the transfer function

$$G'(s) = G(s) + \Delta G(s)$$

with

$$\begin{aligned}\Delta G(s) &= C(sI - A)^{-1}\Delta B + \Delta C(sI - A^{-1})B \\ &\quad + C(sI - A)^{-1}\Delta A(sI - A)^{-1}B,\end{aligned} \quad (6.2-54)$$

where second-order effects have been neglected. Hence (6.2-48) may be used to determine the multiplicative uncertainty bound $m(\omega)$. The cosensitivity $T(j\omega)$ should then satisfy the upper bound (6.2-49) for guaranteed stability in the face of the parameter variations $\Delta A, \Delta B, \Delta C$.

Since $(sI - A)^{-1}$ has a relative degree of at least 1, the high-frequency roll-off of $\Delta G(j\omega)$ is at least -20 dB/decade. Thus plant parameter variations yield an upper bound for the cosensitivity at low frequencies.

Using (6.2-54) it is possible to design robust controllers over a range of operating points that do not require gain scheduling. Compare with [Minto et al., 1990].

6.3 ROBUST OUTPUT-FEEDBACK DESIGN

We should now like to incorporate the robustness concepts introduced in Section 6.2 into the LQ output feedback design procedure for aircraft control systems. This may be accomplished using the following steps.

1. If necessary, augment the plant with added dynamics to achieve the required steady-state error behavior, or to achieve balanced singular values at dc. Use the techniques of Example 6.2-3.
2. Select a performance index, the PI weighting matrices Q and R , and, if applicable, the time weighting factor k in t^k .

3. Determine the optimal output feedback gain K using, for instance, Table 5.4-1 or 5.5-1.
4. Simulate the time responses of the closed-loop system to verify that they are satisfactory. If not, select different Q , R , and k and return to step 3.
5. Determine the low-frequency and high-frequency bounds required for performance robustness and stability robustness. Plot the loop gain singular values to verify that the bounds are satisfied. If they are not, select new Q , R , and k and return to step 3.

An example will illustrate the robust output-feedback design procedure.

Example 6.3-1: Pitch-Rate Control System Robust to Wind Gusts and Unmodeled Flexible Mode. Here we shall illustrate the design of a pitch-rate control system that is robust in the presence of vertical wind gusts and the unmodeled dynamics associated with a flexible mode. It would be worthwhile first to review the pitch rate CAS designed in Examples 4.5-1 and 5.5-3.

a. *Control System Structure.* The pitch rate CAS system is described in Example 5.5-3. The state and measured output are

$$x = \begin{bmatrix} \alpha \\ q \\ \delta_e \\ \alpha_F \\ \epsilon \end{bmatrix}, \quad y = \begin{bmatrix} \alpha_F \\ q \\ \epsilon \end{bmatrix} \quad (1)$$

with α_F the filtered angle of attack and ϵ the output of the integrator added to ensure zero steady-state error. The performance output $z(t)$ that should track the reference input $r(t)$ is $q(t)$.

Linearizing the F-16 dynamics about the nominal flight condition in Table 3.4-1 (502 ft/s, level flight, $x_{cg} = 0.35\bar{c}$) yields

$$\dot{x} = Ax + Bu + Gr \quad (2)$$

$$y = Cx + Fr \quad (3)$$

$$z = Hx \quad (4)$$

with the system matrices given in Example 5.5-3.

The control input is

$$u = -Ky = -[k_\alpha \ k_q \ k_I]y = -k_\alpha\alpha_F - k_qq - k_I\epsilon. \quad (5)$$

It is desired to select the control gains to guarantee a good response to a step

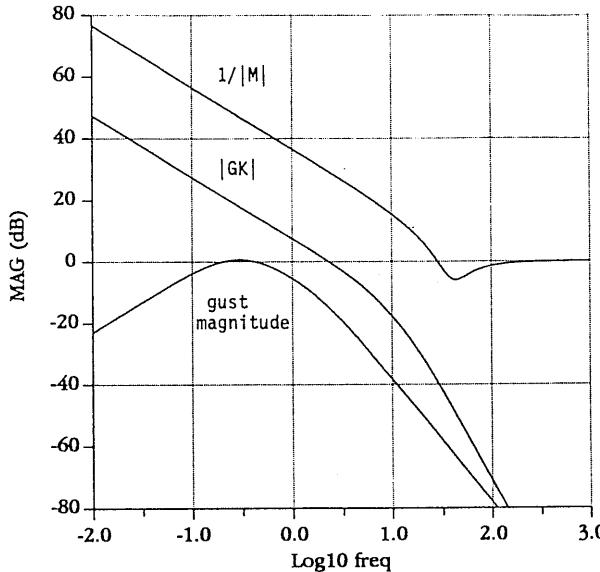


Fig. 6.3-1 Frequency-domain magnitude plots and robustness bounds.

command r in the presence of vertical wind gusts and the unmodeled dynamics of the first flexible mode.

b. Frequency-Domain Robustness Bounds. According to Mil. Spec. 1797 [1987], the vertical wind gust noise has a spectral density given in Dryden form as

$$\Phi_w(\omega) = 2L\sigma^2 \frac{1 + 3L^2\omega^2}{(1 + L^2\omega^2)^2} \quad (6)$$

with ω is the frequency in rad/s, σ the turbulence intensity, and L the turbulence scale length divided by true airspeed. Assuming that the vertical gust velocity is a disturbance input that changes the angle of attack, the software described in Chapter 3 can be used to find a control input matrix from gust velocity to x . Then, using stochastic techniques like those in Example 6.4-2, the magnitude of the gust disturbance versus frequency can be found. It is shown in Fig. 6.3-1. We took $\sigma = 10$ ft/s and $L = (1700 \text{ ft})/(502 \text{ ft/s}) = 3.49$ s.

Let the transfer function of the rigid dynamics from $u(t)$ to $z(t)$ be denoted by $G(s)$. Then the transfer function including the first flexible mode is given by [Blakelock, 1965]

$$G'(s) = G(s)F(s) \quad (7)$$

where

$$F(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (8)$$

with $\omega_n = 40$ rad/s and $\zeta = 0.3$. According to Section 6.2, therefore, the multiplicative uncertainty is given by

$$M(s) = F(s) - I = \frac{-s(s + 2\zeta\omega_n)}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \quad (9)$$

The magnitude of $1/M(j\omega)$ is shown in Fig. 6.3-1.

We should like to perform our controls design using only the rigid dynamics $G(s)$. Then, for performance robustness in the face of the gust disturbance and stability robustness in the face of the first flexible mode, the loop-gain singular values should lie within the bounds implied by the gust disturbance magnitude and $1/|M(j\omega)|$.

c. Controls Design and Robustness Verification. In Example 5.5-3c we performed a derivative-weighting design and obtained the control gains

$$K = [-0.0807 \quad -0.475 \quad 1.361]. \quad (10)$$

The resulting step response is reproduced in Fig. 6.3-2, and the closed-loop

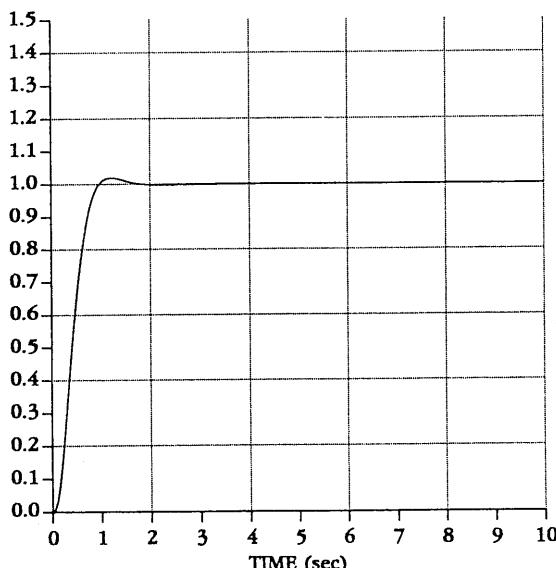


Fig. 6.3-2 Optimal pitch-rate step response.

poles were

$$\begin{aligned} s = & -3.26 \pm j2.83 \\ & -1.02 \\ & -10.67, -14.09. \end{aligned} \quad (11)$$

To verify that the robustness bounds hold for this design, it is necessary to find the loop-gain $GK(s)$ of the closed-loop system. Thus in the figure of Example 5.5-3 it is necessary to find the loop transfer function from $e(t)$ around to $e(t)$ [i.e., from $e(t)$ to $-z(t)$]. With respect to this loop gain, note that some of the elements in (10) are feedforward gains while some are feedback gains.

The magnitude of $GK(j\omega)$ is plotted in Fig. 6.3-1. Note that the robustness bounds are satisfied. Therefore, this design is robust in the presence of vertical turbulence velocities up to 10 ft/s as well as the first flexible mode. ■

6.4 OBSERVERS AND THE KALMAN FILTER

The central theme in Chapter 5 was controls design using partial state or output feedback. We saw in Section 5.4 that by using output feedback a compensator of any desired structure may be used, with the feedback gains being selected by modern LQ techniques. Thus output-feedback design is very suitable for aircraft control. In Section 6.3 we saw how to verify the robustness of the closed-loop system using multivariable Bode plots.

On the other hand, in Section 5.7 we saw that the design equations for full state-variable feedback were simpler than those for output feedback. In fact, in state-variable design it is only necessary to solve the matrix Riccati equation, for which there are many good techniques (ORACLS [Armstrong, 1980], PC-MATLAB [Moler et al., 1987], and MATRIX_x [1989]). By contrast, in output-feedback design it is necessary to solve three coupled nonlinear equations (see Table 5.3-1), which must generally be done using iterative techniques [Moerder and Calise, 1985; Press et al., 1986].

Moreover, in the case of full state feedback, if the system (A, B) is reachable and (\sqrt{Q}, A) is observable (with Q the state weighting in the PI), the Kalman gain is guaranteed to stabilize the plant and yield a global minimum value for the PI. This is a fundamental result of modern control theory, and no such result yet exists for output feedback. The best that may be said is that if the plant is output stabilizable, the algorithm of Table 5.3-2 yields a local minimum for the PI and a stable plant.

Another issue is that the LQ regulator with full state feedback enjoys some important robustness properties that are not guaranteed using output feedback. Specifically, as we shall see in Section 6.5, it has an infinite gain margin and 60% of phase margin.

Thus, state-feedback design offers some advantages over output feedback if the structure of the compensator is of no concern. Although this is rarely the case in aircraft controls, it is nevertheless instructive to pursue a compensator design technique based on state feedback.

Since all the states are seldom available, the first order of business is to estimate the full state $x(t)$ given only partial information in the form of the measured outputs $y(t)$. This is the *observer design* problem. Having estimated the state, we may then use the *estimate* of the state for feedback purposes, designing a feedback gain *as if* all the states were measurable. The combination of the observer and the state feedback gain is then a dynamic regulator similar to those used in classical control, as we shall show in the last portion of this section. In the modern approach, however, it is straightforward to design multivariable regulators with desirable properties by solving matrix equations due to the fundamental *separation principle*, which states that the feedback gain and observer may be designed separately and then concatenated.

One of our prime objectives in this section and the next is to discuss the linear quadratic Gaussian/loop-transfer recovery (LQG/LTR) technique for controls design. This is an important modern technique for the design of robust aircraft control systems. It relies on full state-feedback design, followed by the design of an observer that allows full recovery of the guaranteed robustness properties of the LQ regulator with state feedback.

Of course, observers and filters have important applications in aircraft in their own right. For instance, the angle of attack is difficult to measure accurately; however, using an observer or Kalman filter it is not difficult to estimate the angle of attack very precisely by measuring pitch rate and normal acceleration (see Example 6.4-2).

Observer Design

In aircraft control, all of the states are rarely available for feedback purposes. Instead, only the measured outputs are available. Using modern control theory, if the measured outputs capture enough information about the dynamics of the system, it is possible to use them to *estimate* or *observe* all the states. Then these state estimates may be used for feedback purposes.

To see how a state observer can be constructed, consider the aircraft equations in state-space form

$$\dot{x} = Ax + Bu \quad (6.4-1)$$

$$y = Cx \quad (6.4-2)$$

with $x(t) \in \mathbf{R}^n$ the state, $u(t) \in \mathbf{R}^m$ the control input, and $y(t) \in \mathbf{R}^p$ the available measured outputs.

Let the estimate of $x(t)$ be $\hat{x}(t)$. We claim that the state observer is a dynamical system described by

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \quad (6.4-3)$$

or

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly \equiv A_0\hat{x} + Bu + Ly. \quad (6.4-4)$$

That is, the observer is a system with two inputs, namely $u(t)$ and $y(t)$, both of which are known.

Since $\hat{x}(t)$ is the state estimate, we could call

$$\hat{y} = C\hat{x} \quad (6.4-5)$$

the estimated output. It is desired that $\hat{x}(t)$ be close to $x(t)$. Thus, if the observer is working properly, the quantity $y - \hat{y}$ that appears in (6.4-3) should be small. In fact,

$$\tilde{y} = y - \hat{y} \quad (6.4-6)$$

is the *output estimation error*.

It is worth examining Fig. 6.4-1, which depicts the state observer. Note that the observer consists of two parts: a *model of the system* involving (A, B, C) , and an *error-correcting portion* that involves the output error multiplied by L . We call matrix L the *observer gain*.

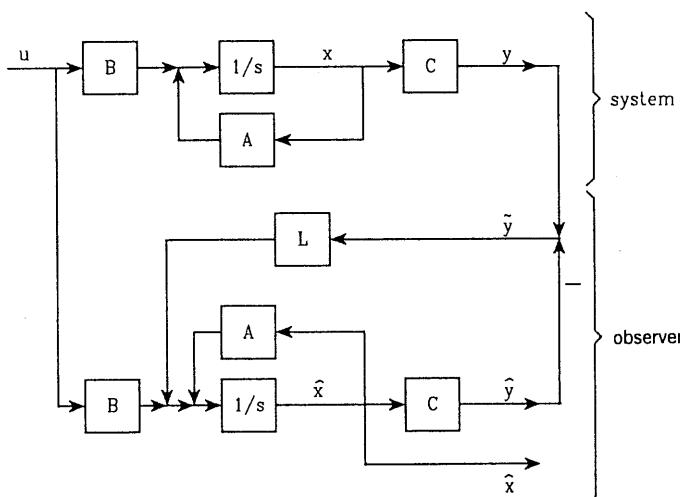


Fig. 6.4-1 State observer.

To demonstrate that the proposed dynamical system is indeed an observer, it is necessary to show that it manufactures an estimate $\hat{x}(t)$ that is close to the actual state $x(t)$. For this purpose, define the (*state*) estimation error as

$$\tilde{x} = x - \hat{x}. \quad (6.4-7)$$

By differentiating (6.4-7) and using (6.4-1) and (6.4-4), it is seen that the estimation error has dynamics given by

$$\dot{\tilde{x}} = (A - LC)\tilde{x} = A_0\tilde{x}. \quad (6.4-8)$$

The initial estimation error is $\tilde{x}(0) = x(0) - \hat{x}(0)$, with $\hat{x}(0)$ the initial estimate, which is generally taken as zero.

It is required that the estimation error vanish with time for any $\tilde{x}(0)$, for then $\hat{x}(t)$ will approach $x(t)$. This will occur if $A_0 = (A - LC)$ is asymptotically stable. Therefore, as long as we select the observer gain L so that $(A - LC)$ is stable, (6.4-3) is indeed an observer for the state in (6.4-1). The observer design problem is to select L so that the error vanishes suitably quickly. It is a well-known result of modern control theory that the poles of $(A - LC)$ may be arbitrarily assigned to desired locations if and only if (C, A) is observable.

Since, according to Fig. 6.4-1, we are injecting the output into the state derivative, L is called an *output injection*. Observers of the sort we are mentioning here are called *output-injection observers*, and their design could be called output-injection design.

It is important to discuss the output-injection problem of selecting L so that $(A - LC)$ is stable, for it is a problem we have already solved under a different guise. The state-feedback control law for system (6.4-1) is

$$u = -Kx, \quad (6.4-9)$$

which results in the closed-loop system

$$\dot{x} = (A - BK)x. \quad (6.4-10)$$

The state-feedback design problem is to select K for desired closed-loop properties. We have shown how this may be accomplished in Section 5.7. Thus if we select the feedback gain as the Kalman gain

$$K = R^{-1}B^T P \quad (6.4-11)$$

with P the positive-definite solution to the algebraic Riccati equation (ARE)

$$0 = A^T P + PA + Q - PBR^{-1}B^T P, \quad (6.4-12)$$

then if (A, B) is reachable and (\sqrt{Q}, A) is observable, the closed-loop system

is guaranteed to be stable. The matrices Q and R are design parameters that will determine the closed-loop dynamics, as we have seen in the examples of Chapter 5.

Now, compare (6.4-8) and (6.4-10). They are very similar. In fact,

$$(A - LC)^T = A^T - C^T L^T, \quad (6.4-13)$$

which has the free matrix L^T to the right, exactly as in the state feedback problem involving $(A - BK)$. This important fact is called *duality*; that is, state feedback and output injection are duals. [Note that $A - LC$ and $(A - LC)^T$ have the same poles.]

The important result of duality for us is that *the same theory we have developed for selecting the state-feedback gain may be used to select the output-injection gain L*. In fact, compare (6.4-13) to $(A - BK)$. Now, in the design equations (6.4-11) and (6.4-12) let us replace A , B , and K everywhere they occur by A^T , C^T , and L^T , respectively. The result is

$$\begin{aligned} L^T &= R^{-1}CP \\ 0 &= AP + PA^T + Q - PC^TR^{-1}CP. \end{aligned} \quad (6.4-14)$$

The first of these may be rewritten as

$$L = PC^TR^{-1}. \quad (6.4-15)$$

We call (6.4-14) the *observer ARE*.

Let us note the following connection between reachability and observability. Taking the transpose of the reachability matrix yields

$$\begin{aligned} U^T &= [B \quad AB \quad A^2B \quad \cdots \quad A^{n-1}B]^T \\ &= \begin{bmatrix} B^T \\ B^TA \\ \vdots \\ B^T(A^T)^{n-1} \end{bmatrix}. \end{aligned} \quad (6.4-16)$$

However, the observability matrix is

$$V = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}. \quad (6.4-17)$$

Comparing U^T and V , it is apparent that they have the same form. In fact, since U and U^T have the same rank it is evident that (A, B) is reachable if and only if (B^T, A^T) is observable. This is another aspect of duality.

Taking into account these notions, an essential result on output injection is the following. It is the dual of the guaranteed stability using the Kalman gain discussed in Section 5.7. Due to its importance, we formulate it as a theorem.

Theorem. Let (C, A) be observable and (A, \sqrt{Q}) be reachable. Then the error system (6.4-8) using the gain L given by (6.4-15), with P the unique positive definite solution to (6.4-14), is asymptotically stable. ■

Stability of the error system guarantees that the state estimate $\hat{x}(t)$ will approach the actual state $x(t)$. By selecting L to place the poles of $(A - LC)$ far enough to the left in the s -plane, the estimation error $\tilde{x}(t)$ can be made to vanish as quickly as desired.

The power of this theorem is that we may treat Q and R as design parameters which may be tuned until suitable observer behavior results for the gain computed from the observer ARE. As long as we select Q and R to satisfy the theorem, observer stability is assured. An additional factor, of course, is that software for solving the observer ARE is readily available (e.g., ORACLS [Armstrong, 1980], PC-MATLAB [Moler et al., 1987], and MATRIX_x [1989]).

We have assumed that the system matrices (A, B, C) are exactly known. Unfortunately, in reality this is rarely the case. In aircraft control, for instance, (6.4-1)–(6.4-2) represent a model of a nonlinear system at an equilibrium point. Variations in the operating point will result in variations in the elements of A , B , and C . However, if the poles of $(A - LC)$ are selected far enough to the left in the s -plane (i.e., fast enough), the estimation error will be small despite uncertainties in the system matrices. That is, the observer has some robustness to modeling inaccuracies.

It is worth mentioning that there are many other techniques for the selection of the observer gain L . In the single-output case the observability matrix V is square. Then Ackermann's formula [Franklin et al., 1986] may be used to compute L . If

$$\Delta_0(s) = |sI - (A - LC)| \quad (6.4-18)$$

is the desired observer characteristic polynomial, the required observer gain is given by

$$L = \Delta_0(A)V^{-1}e_n, \quad (6.4-19)$$

with $e_n = [0 \quad \cdots \quad 0 \quad 1]^T$ the last column of the $n \times n$ identity matrix.

A general rule of thumb is that for suitable accuracy in the state estimate $\hat{x}(t)$, the slowest observer pole should have a real part 5 to 10 times larger than the real part of the fastest system pole. That is, the observer time constants should be 5 to 10 times larger than the system time constants.

Example 6.4-1: Observer Design for Double Integrator System. In Example 5.7-1 we discussed state-feedback design for systems obeying Newton's laws,

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}x + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u = Ax + Bu, \quad (1)$$

where the state is $x = [d \ v]^T$, with $d(t)$ the position and $v(t)$ the velocity, and the control $u(t)$ is an acceleration input. Let us take position measurements so that the measured output is

$$y = [1 \ 0]x = Cx. \quad (2)$$

We should like to design an observer that will reconstruct the full state $x(t)$ given only position measurements. Let us note that simple differentiation of $y(t) = d(t)$ to obtain $v(t)$ is unsatisfactory, since differentiation increases sensor noise. In fact, the observer is a *low-pass* filter that provides estimates while rejecting high-frequency noise. We shall discuss two techniques for observer design.

a. *Riccati Equation Design.* There is good software available in standard design packages for solving the observer ARE (e.g., ORACLS [Armstrong, 1980] and PC-MATLAB [Moler et al., 1987]). However, in this example we want to solve the ARE analytically to show the relation between the design parameters Q and R and the observer poles.

Selecting $R = 1$ and $Q = \text{diag}\{q_d, q_v^2\}$ with q_d and q_v nonnegative, we may assume that

$$P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \quad (3)$$

for some scalars p_1 , p_2 , and p_3 to be determined. The observer ARE (6.4-14) becomes

$$0 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} + \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} q_d & 0 \\ 0 & q_v^2 \end{bmatrix} - \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}. \quad (4)$$

which may be multiplied out to obtain the three scalar equations

$$0 = 2p_2 - p_1^2 + q_d \quad (5a)$$

$$0 = p_3 - p_1 p_2 \quad (5b)$$

$$0 = -p_2^2 + q_v^2. \quad (5c)$$

Solving these equations gives

$$p_2 = q_v \quad (6a)$$

$$p_1 = \sqrt{2} \sqrt{q_v + \frac{q_d}{2}} \quad (6b)$$

$$p_3 = q_v \sqrt{2} \sqrt{q_v + \frac{q_d}{2}}, \quad (6c)$$

where we have selected the signs that make P positive definite.

According to (6.4-15), the observer gain is equal to

$$L = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}. \quad (7)$$

Therefore,

$$L = \begin{bmatrix} \sqrt{2} \sqrt{q_v + \frac{q_d}{2}} \\ q_v \end{bmatrix}. \quad (8)$$

Using (8), the error system matrix is found to be

$$A_0 = (A - LC) = \begin{bmatrix} -\sqrt{2} \sqrt{q_v + \frac{q_d}{2}} & 1 \\ -q_v & 0 \end{bmatrix}. \quad (9)$$

Therefore, the observer characteristic polynomial is

$$\Delta_0(s) = |sI - A_0| = s^2 + 2\xi\omega s + \omega^2, \quad (10)$$

with the observer natural frequency ω and damping ratio ξ given by

$$\omega = \sqrt{q_v}, \quad \xi = \frac{1}{\sqrt{2}} \sqrt{1 + \frac{q_d}{2q_v}}. \quad (11)$$

It is now clear how selection of Q affects the observer behavior. Note that if $q_d = 0$, the damping ratio becomes the familiar $1/\sqrt{2}$.

The reader should verify that the system is observable and that (A, \sqrt{Q}) is reachable as long as $q_v \neq 0$. A comparison with Example 5.7-1, where a state feedback was designed for Newton's system, reveals some interesting aspects of duality.

b. Ackermann's Formula Design. Riccati equation observer design is useful whether the plant has only one or multiple outputs. If there is only one output, we may use Ackermann's formula (6.4-19).

Let the desired observer polynomial be

$$\Delta_0(s) = s^2 + 2\zeta\omega s + \omega^2 \quad (12)$$

for some specified damping ratio ζ and natural frequency ω . Then

$$\Delta_0(A) = A^2 + 2\zeta\omega A + \omega^2 I = \begin{bmatrix} \omega^2 & 2\zeta\omega \\ 0 & \omega^2 \end{bmatrix} \quad (13)$$

$$V = \begin{bmatrix} C \\ CA \end{bmatrix} = I, \quad (14)$$

so that the observer gain is

$$L = \begin{bmatrix} 2\zeta\omega \\ \omega^2 \end{bmatrix}. \quad (15)$$

One may verify that the characteristic polynomial of $A_0 = A - LC$ is indeed (12).

c. *Simulation.* To design an observer with a complex pole pair having damping ratio of $\zeta = 1/\sqrt{2}$ and natural frequency of $\omega = 1$ rad/s, the

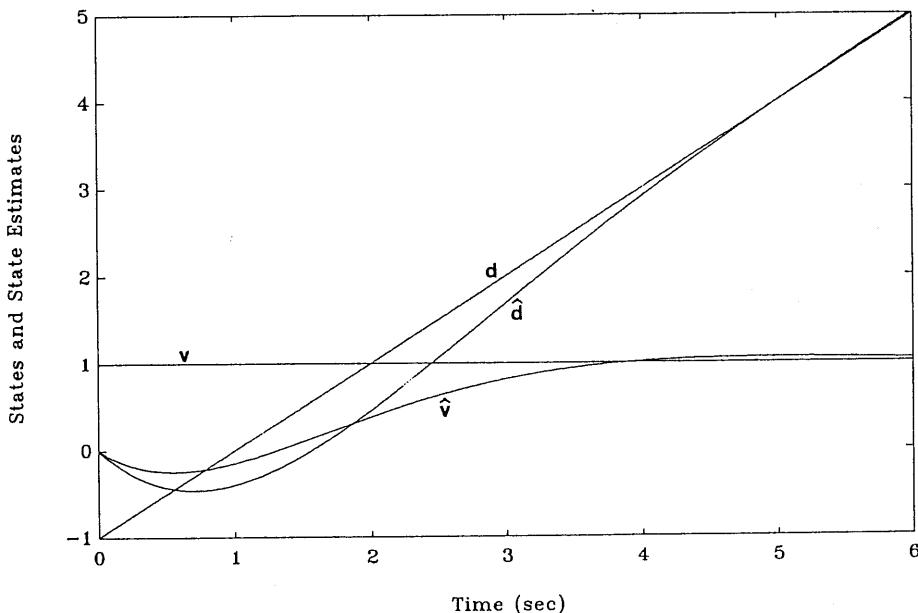


Fig. 6.4-2 Actual and estimated states.

observer gain was selected as

$$L = \begin{bmatrix} \sqrt{2} \\ 1 \end{bmatrix}. \quad (16)$$

A simulation was performed. The time histories of the actual states and their estimates are shown in Fig. 6.4-2. The initial conditions were $d(0) = -1$, $v(0) = 1$ and the input was $u(t) = 0$. The observer was started with initial states of $\hat{d}(0) = 0$, $\hat{v}(0) = 0$.

The Kalman Filter

Throughout Chapter 5 we assumed that the system is exactly known and that no modeling inaccuracies, disturbances, or noises are present. In fact, nature is seldom so cooperative. In Sections 6.2 and 6.3 we showed how to take account of uncertainties in the model and the environment using a robust frequency-domain approach. An alternative is to treat uncertainties using *probability theory*.

In this subsection we develop the Kalman filter, which is based on a probabilistic treatment of process and measurement noises. The Kalman filter is an observer that is used for navigation and other applications that require the reconstruction of the state from noisy measurements. Since it is fundamentally a low-pass filter, it has good noise rejection capabilities. In Example 6.4-2 we show how to use the Kalman filter to estimate the angle of attack in the face of gust disturbances. In Section 6.5 we show how to use a state-variable feedback and a Kalman filter to design robust aircraft controllers by using the LQG/LTR technique.

We begin with a brief review of probability theory. It is not necessary to follow the derivation to use the Kalman filter: it is only necessary to solve the design equations in Table 6.4-1. Thus one could skip the review that follows. However, an understanding of the theory will result in more sensible application of the filter. Supplemental references are [Gelb, 1974; Lewis, 1986b].

A Brief Review of Probability Theory. Suppose that the plant is described by the stochastic dynamical equation

$$\dot{x} = Ax + Bu + Gw \quad (6.4-20)$$

$$y = Cx + v \quad (6.4-21)$$

with state $x(t) \in \mathbf{R}^n$, control input $u(t) \in \mathbf{R}^m$, and measured output $y(t) \in \mathbf{R}^p$. Signal $w(t)$ is an unknown *process noise* that acts to disturb the plant. It

could represent the effects of wind gusts, for instance, or unmodeled high-frequency plant dynamics. Signal $v(t)$ is an unknown *measurement noise* that acts to impair the measurements; it could represent sensor noise.

Since (6.4-20) is driven by process noise, the state $x(t)$ is now also a random process, as is $y(t)$. To investigate average properties of random processes we will require several concepts from probability theory [Papoulis, 1984]. The point is that although $w(t)$ and $v(t)$ represent unknown random processes, we do in fact know something about them which can help us in controls design. For instance, we may know their average values or total energy content. The concepts we shall now define allow us to incorporate this general sort of knowledge into our theory.

Given a random vector $z \in \mathbf{R}^n$, we denote by $f_z(\zeta)$ the *probability density function (PDF)* of z . The PDF represents the probability that z takes on a value within the differential region $d\zeta$ centered at ζ . Although the value of z may be unknown, it is quite common in many situations to have a good feel for its PDF.

The *expected value* of a function $g(z)$ of a random vector z is defined as

$$E\{g(z)\} = \int_{-\infty}^{\infty} g(\zeta) f_z(\zeta) d\zeta. \quad (6.4-22)$$

The *mean* or *expected value* of z is defined by

$$E\{z\} = \int_{-\infty}^{\infty} \zeta f_z(\zeta) d\zeta, \quad (6.4-23)$$

which we shall symbolize by \bar{z} to economize on notation. Note that $\bar{z} \in \mathbf{R}^n$.

The covariance of z is given by

$$P_z = E\{(z - \bar{z})(z - \bar{z})^T\}. \quad (6.4-24)$$

Note that P_z is an $n \times n$ constant matrix.

An important class of random vectors is characterized by the *gaussian* or *normal* PDF

$$f_z(\zeta) = \frac{1}{\sqrt{(2\pi)^n |P_z|}} e^{-(\zeta - \bar{z})^T P_z^{-1} (\zeta - \bar{z})/2}. \quad (6.4-25)$$

In the scalar case $n = 1$ this reduces to the more familiar

$$f_z(\zeta) = \frac{1}{\sqrt{2\pi P_z}} e^{-(\zeta - \bar{z})^2 / 2P_z}, \quad (6.4-26)$$

which is illustrated in Fig. 6.4-3. Such random vectors take on values near the mean \bar{z} with greatest probability and have a decreasing probability of taking

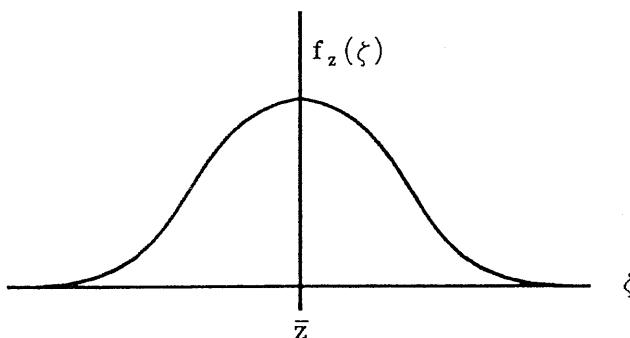


Fig. 6.4-3 Gaussian PDF.

on values farther away from \bar{z} . Many naturally occurring random variables are gaussian.

If the random vector is a time function, it is called a *random process*, symbolized as $z(t)$. Then the PDF may also be time varying and we write $f_z(\zeta, t)$. One can imagine the PDF in Fig. 6.4-3 changing with time. In this situation, the expected value and covariance matrix are also functions of time, so we write $\bar{z}(t)$ and $P_z(t)$.

Many random processes $z(t)$ of interest to us have a time-invariant PDF. These are *stationary* processes and, even though they are random time functions, they have a constant mean and covariance.

To characterize the relation between two random processes $z(t)$ and $x(t)$ we employ the *joint PDF* $f_{zx}(\zeta, \xi, t_1, t_2)$, which represents the probability that $(z(t_1), x(t_2))$ is within the differential area $d\zeta \times d\xi$ centered at (ζ, ξ) . For our purposes, we shall assume that the processes $z(t)$ and $x(t)$ are *jointly stationary*; that is, the joint PDF is not a function of both times t_1 and t_2 , but depends only on the difference $(t_1 - t_2)$.

In the stationary case, the expected value of the function of two variables $g(z, x)$ is defined as

$$E\{g(z(t_1), x(t_2))\} = \int_{-\infty}^{\infty} g(\zeta, \xi) f_{z,x}(\zeta, \xi, t_1 - t_2) d\zeta d\xi. \quad (6.4-27)$$

In particular, the *cross-correlation matrix* is defined by

$$R_{zx}(\tau) = E\{z(t + \tau) x^T(t)\}. \quad (6.4-28)$$

In the sequel, we shall briefly require the cross-correlation matrix of two nonstationary processes, which is defined as

$$R_{zx}(t, \tau) = E\{z(t) x^T(\tau)\}. \quad (6.4-29)$$

Considering $z(t_1)$ and $z(t_2)$ as two jointly distributed random stationary processes, we may define the *autocorrelation function* of $z(t)$ as

$$R_z(\tau) = E\{z(t + \tau)z^T(t)\}. \quad (6.4-30)$$

The autocorrelation function gives us some important information about the random process $z(t)$. For instance,

$$\text{tr}[R_z(0)] = \text{tr}[E\{z(t)z^T(t)\}] = E\{\|z(t)\|^2\}$$

is equal to the total energy in the process $z(t)$. [In writing this equation recall that for any compatible matrices M and N , $\text{tr}(MN) = \text{tr}(NM)$.]

If

$$R_{zx}(\tau) = 0, \quad (6.4-31)$$

we call $z(t)$ and $x(t)$ *orthogonal*. If

$$R_z(\tau) = P\delta(\tau) \quad (6.4-32)$$

where P is a constant matrix and $\delta(\tau)$ is the Dirac delta, then $z(t)$ is orthogonal to $z(t + \tau)$ for any $\tau \neq 0$. What this means is that the value of the process $z(t)$ at one time t is unrelated to its value at another time $\tau \neq t$. Such a process is called *white noise*. An example is the thermal noise in an electric circuit, which is due to the thermal agitation of the electrons in the resistors.

Note that $P\delta(0)$ is the covariance of $z(t)$, which is unbounded. We call P a *spectral density matrix*. It is sometimes loosely referred to as a covariance matrix.

Derivation of the Kalman Filter. We may now return to system (6.4-20)/(6.4-21). Neither the initial state $x(0)$, the process noise $w(t)$, nor the measurement noise $v(t)$ is exactly known. However, in practice we may have some feeling for their general characteristics. Using the concepts we have just discussed, we may formalize this general knowledge so that it may be used in controls design.

The process noise is due to some sort of system disturbance, such as wind gusts; the measurement noise is due to sensor inaccuracies; and the initial state is uncertain because of our ignorance. Since these are all unrelated, it is reasonable to assume that $x(0)$, $w(t)$, and $v(t)$ are mutually orthogonal. Some feeling for $x(0)$ may be present in that we may know its mean \bar{x}_0 and covariance P_0 . We symbolize this as

$$x(0) \sim (\bar{x}_0, P_0). \quad (6.4-33)$$

It is not unreasonable to assume that $w(t)$ and $v(t)$ have means of zero, since, for instance, there should be no bias on the measuring instruments. We shall also assume that the process noise and measurement noise are white noise processes, so that

$$R_w(\tau) = E\{w(t + \tau)w^T(t)\} = Q\delta(\tau) \quad (6.4-34)$$

$$R_v(\tau) = E\{v(t + \tau)v^T(t)\} = R\delta(\tau). \quad (6.4-35)$$

Spectral density matrices Q and R will be assumed known. [Often, we have a good feeling for the standard deviations of $w(t)$ and $v(t)$.] According to (6.4-30), Q and R are positive semidefinite. We shall assume in addition that R is nonsingular. In summary, we shall assume that

$$w(t) \sim (0, Q), \quad Q \geq 0 \quad (6.4-36)$$

$$v(t) \sim (0, R), \quad R > 0. \quad (6.4-37)$$

The assumption that $w(t)$ and $v(t)$ are white may in some applications be a bad one. For instance, wind-gust noise is generally of low frequency. However, suppose that $w(t)$ is not white. Then we can determine a system description

$$\dot{x}_w = A_w x_w + B_w n \quad (6.4-38)$$

$$w = C_w x_w + D_w n \quad (6.4-39)$$

which has a white noise input $n(t)$ and output $w(t)$. This is called a *noise-shaping filter*. These dynamics may be combined with the plant (6.4-20)–(6.4-21) to obtain the augmented dynamics

$$\begin{bmatrix} x \\ x_w \end{bmatrix} = \begin{bmatrix} A & GC_w \\ 0 & A_w \end{bmatrix} \begin{bmatrix} x \\ x_w \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} CD_w \\ B_w \end{bmatrix} n \quad (6.4-40)$$

$$y = [C \quad 0] \begin{bmatrix} x \\ x_w \end{bmatrix} + v. \quad (6.4-41)$$

This augmented system does have a white process noise $n(t)$. A similar procedure may be followed if $v(t)$ is nonwhite. Thus we can generally describe a plant with nonwhite noises in terms of an augmented system with white process and measurement noises.

The determination of a system (6.4-38)/(6.4-39) that describes nonwhite noise $w(t)$ [or $v(t)$] is based on factoring the spectral density of the noise $w(t)$. For details, see Lewis [1986b]. We shall illustrate the procedure in Example 6.4-2.

We should now like to design an estimator for the stochastic system

(6.4-20)/(6.4-21) under the assumptions just listed. We shall propose the output-injection observer, which has the form

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \quad (6.4-42)$$

or

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly. \quad (6.4-43)$$

The time function $\hat{x}(t)$ is the state estimate and

$$\hat{y} = E\{Cx + v\} = C\hat{x} \quad (6.4-44)$$

is the estimate of the output $y(t)$. (This expected value is actually the *conditional* mean given the previous measurements; see Lewis [1986b].)

The estimator gain L must be selected to provide an optimal estimate in the presence of the noises $w(t)$ and $v(t)$. To select L , we shall need to define the estimation error

$$\tilde{x}(t) = x(t) - \hat{x}(t). \quad (6.4-45)$$

Using (6.4-20) and (6.4-42), we may derive the error dynamics to be

$$\begin{aligned} \dot{\tilde{x}} &= (A - LC)\tilde{x} + Gw - Lv \\ &\equiv A_0\tilde{x} + Gw - Lv. \end{aligned} \quad (6.4-46)$$

Note that the error system is driven by both the process and measurement noise. The output of the error system may be taken as $\tilde{y} = y - \hat{y}$ so that

$$\tilde{y} = C\tilde{x}. \quad (6.4-47)$$

The *error covariance* is given by

$$P(t) = E\{\tilde{x}\tilde{x}^T\}, \quad (6.4-48)$$

which is time-varying. Thus $\tilde{x}(t)$ is a nonstationary random process. The error covariance is a measure of the *uncertainty* in the estimate, and smaller values for $P(t)$ mean that the estimate is better, since the error is more closely distributed about its mean value of zero if $P(t)$ is smaller.

If the observer is asymptotically stable and $w(t)$ and $v(t)$ are stationary processes, the error $\tilde{x}(t)$ will eventually reach a *steady state* in which it is also stationary with constant mean and covariance. The gain L will be chosen to minimize the *steady-state error covariance* P . Thus the optimal gain L will be a constant matrix of observer gains.

Before determining the optimal gain L , let us compute the mean and covariance of the estimation error $\tilde{x}(t)$. Using (6.4-46) and the linearity of the expectation operator,

$$E\{\dot{\tilde{x}}\} = A_0 E\{\tilde{x}\} + GE\{w\} - LE\{v\}, \quad (6.4-49)$$

so that

$$\frac{d}{dt} E\{\tilde{x}\} = A_0 E\{\tilde{x}\}. \quad (6.4-50)$$

Thus $E\{\tilde{x}\}$ is a deterministic time-varying quantity that obeys a differential equation with system matrix A_0 . If $A_0 = A - LC$ is stable, then $E\{\tilde{x}\}$ eventually stabilizes at a steady-state value of zero, since the process and measurement noise are of zero mean. Since

$$E\{\tilde{x}\} = E\{x\} - E\{\hat{x}\} = E\{x\} - \hat{x}, \quad (6.4-51)$$

it follows that in this case the estimate $\hat{x}(t)$ approaches $E\{x(t)\}$. Then the estimate is said to be *unbiased*. According also to (6.4-51), the mean of the initial error $\tilde{x}(0)$ is equal to zero if the observer (6.4-43) is initialized to $\hat{x}(0) = \bar{x}_0$, with \bar{x}_0 the mean of $x(0)$.

If the process noise $w(t)$ and/or measurement noise $v(t)$ have means that are not zero, then according to (6.4-49), the steady-state value of $E\{\tilde{x}\}$ is not equal to zero. In this case, $\hat{x}(t)$ does not tend asymptotically to the true state $x(t)$, but is offset from it by the constant value $-E\{\tilde{x}\}$. Then the estimates are said to be biased (see the Problems).

To determine the error covariance, note that the solution of (6.4-46) is given by

$$\tilde{x}(t) = e^{A_0 t} \tilde{x}(0) - \int_0^t e^{A_0(t-\tau)} L v(\tau) d\tau + \int_0^t e^{A_0(t-\tau)} G w(\tau) d\tau. \quad (6.4-52)$$

We shall soon require the cross-correlation matrices $R_{v\tilde{x}}(t, t)$ and $R_{w\tilde{x}}(t, t)$. To find them, use (6.4-52) and the assumption that $x(0)$ [and hence $\tilde{x}(0)$], $w(t)$, and $v(t)$ are orthogonal. Thus

$$\begin{aligned} R_{v\tilde{x}}(t, t) &= E\{v(t) \tilde{x}^T(t)\} \\ &= - \int_0^t E\{v(t) v^T(\tau)\} L^T e^{A_0^T(t-\tau)} d\tau. \end{aligned} \quad (6.4-53)$$

Note that

$$R_v(t, \tau) = R \delta(t - \tau) \quad (6.4-54)$$

but the integral in (6.4-53) has an upper limit of t . Recall that the unit

impulse can be expressed as

$$\delta(t) = \lim_{T \rightarrow 0} \frac{1}{T} \prod\left(\frac{t}{T}\right) \quad (6.4-55)$$

where the rectangle function

$$\frac{1}{T} \prod\left(\frac{t}{T}\right) = \begin{cases} \frac{1}{T}, & |t| < \frac{T}{2} \\ 0 & \text{otherwise} \end{cases} \quad (6.4-56)$$

is centered at $t = 0$. Therefore, only half the area of $\delta(t - \tau)$ should be considered as being to the left of $\tau = t$. Hence (6.4-53) is

$$R_{v\tilde{x}}(t, t) = -\frac{1}{2}RL^T. \quad (6.4-57)$$

Similarly,

$$\begin{aligned} R_{w\tilde{x}}(t, t) &= E\{w(t)\tilde{x}^T(t)\} \\ &= \int_0^t E\{w(t)w^T(\tau)\}G^T e^{A^T_0}(t - \tau) d\tau \end{aligned} \quad (6.4-58)$$

or

$$R_{w\tilde{x}}(t, t) = \frac{1}{2}QG^T. \quad (6.4-59)$$

To find a differential equation for $P(t) = E\{\tilde{x}\tilde{x}^T\}$, write

$$\dot{P}(t) = E\left\{\frac{d\tilde{x}}{dt}\tilde{x}^T\right\} + E\left\{\tilde{x}\frac{d\tilde{x}^T}{dt}\right\} \quad (6.4-60)$$

According to the error dynamics (6.4-46) the first term is equal to

$$E\left\{\frac{d\tilde{x}}{dt}\tilde{x}^T\right\} = (A - LC)P + \frac{1}{2}LRL^T + \frac{1}{2}GQG^T, \quad (6.4-61)$$

where we have used (6.4-57) and (6.4-59). To this equation add its transpose to obtain

$$\dot{P} = A_0 P + P A_0^T + LRL^T + GQG^T. \quad (6.4-62)$$

What we have derived in (6.4-62) is an expression for the error covariance when the observer (6.4-43) is used with a specific gain L . Given any L such that $(A - LC)$ is stable, we may solve (6.4-62) for $P(t)$, using as initial condition $P(0) = P_0$, with P_0 the covariance of the initial state, which represents the uncertainty in the initial estimate $\hat{x}(0) = \bar{x}_0$.

Clearly, gains that result in smaller error covariances $P(t)$ are better, for then the error $\tilde{x}(t)$ is generally closer to its mean of zero. That is, the error covariance is a measure of the performance of the observer, and smaller covariance matrices are indicative of better observers. We say that P is a measure of the uncertainty in the estimate. [Given symmetric positive semidefinite matrices P_1 and P_2 , P_1 is less than P_2 if $(P_2 - P_1) \geq 0$.]

The error covariance $P(t)$ reaches a bounded steady-state value P as $t \rightarrow \infty$ as long as A_0 is asymptotically stable. At steady state, $\dot{P} = 0$ so that (6.4-62) becomes the algebraic equation

$$0 = A_0 P + P A_0^T + L R L^T + G Q G^T. \quad (6.4-63)$$

The steady-state error covariance is the positive (semi)definite solution to (6.4-63). To obtain a constant observer gain, we may select L to minimize the steady-state error covariance P . Necessary conditions for L are now easily obtained after the same fashion that the output feedback gain K was obtained in Section 5.3.

Thus define a performance index (PI)

$$J = \frac{1}{2} \operatorname{tr}(P). \quad (6.4-64)$$

[Note that $\operatorname{tr}(P)$ is the sum of the eigenvalues of P . Thus a small J corresponds to a small P .] To select L so that J is minimized subject to the constraint (6.4-63), define the Hamiltonian

$$H = \frac{1}{2} \operatorname{tr}(P) + \frac{1}{2} \operatorname{tr}(gS), \quad (6.4-65)$$

where

$$g = A_0 P + P A_0^T + L R L^T + G Q G^T \quad (6.4-66)$$

and S is an $n \times n$ undetermined (Lagrange) multiplier.

To minimize J subject to the constraint $g = 0$, we may equivalently minimize H with no constraints. Necessary conditions for a minimum are therefore given by

$$\frac{\partial H}{\partial S} = A_0 P + P A_0^T + L R L^T + G Q G^T = 0 \quad (6.4-67)$$

$$\frac{\partial H}{\partial P} = A_0^T S + S A_0 + I = 0 \quad (6.4-68)$$

$$\frac{1}{2} \frac{\partial H}{\partial L} = S L R - S P C^T = 0. \quad (6.4-69)$$

If A_0 is stable, the solution S to (6.4-68) is positive definite. Then, according

to (6.4-69),

$$L = PC^T R^{-1}. \quad (6.4-70)$$

Substituting this value for L into (6.4-67) yields

$$(A - PC^T R^{-1} C)P + P(A - PC^T R^{-1} C)^T + PC^T R^{-1} CP + GQG^T = 0 \quad (6.4-71)$$

or

$$AP + PA^T + GQG^T - PC^T R^{-1} CP = 0. \quad (6.4-72)$$

To determine the optimal observer gain L , we may therefore proceed by solving (6.4-72) for the error covariance P , and then using (6.4-70) to compute L . The matrix quadratic equation (6.4-72) is called the *algebraic (filter) Riccati equation (ARE)*. There are several efficient techniques for solving the ARE for P (e.g., [Armstrong, 1980; IMSL, 1980], MATRIX_x [1989], MATLAB [Moler et al., 1987]).

The optimal gain L determined using (6.4-70) is called the (*steady-state Kalman gain*, and the observer so constructed is called the (*steady-state Kalman filter*). The term “steady state” refers to the fact that although the optimal gain that minimizes $P(t)$ is generally time varying, we have selected the optimal gain that minimizes the *steady-state* error covariance in order to obtain a constant observer gain. Since the gain must eventually be gain-scheduled in actual flight controls applications, we require a constant gain to keep the number of parameters to be scheduled within reason.

The design equations for the Kalman filter are collected in Table 6.4-1. A block diagram appears in Fig. 6.4-1. The steady-state Kalman filter is the best estimator with constant gains that has the dynamics of the form in the table. Such a filter is said to be *linear*. It can be shown [Lewis, 1986b] that if the process noise $w(t)$ and measurement noise $v(t)$ are *gaussian*, this is also *the optimal* steady-state estimator of any form.

The quantity

$$\tilde{y}(t) = y(t) - \hat{y}(t) = y(t) - C\hat{x}(t) \quad (6.4-73)$$

that drives the filter dynamics in the table is called the residual. For more information on the Kalman filter, see Bryson and Ho [1975], Kwakernaak and Sivan [1972], and Lewis [1986b].

The filter ARE should be compared to the ARE we discussed at the beginning of this section in connection with output-injection design. There, no particular meaning was given to the auxiliary matrix P . In this stochastic setting, we have discovered that it is nothing but the error covariance. Small values of P generally indicate a filter with good estimation performance.

TABLE 6.4-1. The Kalman Filter

System Model

$$\begin{aligned}\dot{x} &= Ax + Bu + Gw \\ y &= Cx + v \\ x(0) &\sim (\bar{x}_0, P_0), \quad w(t) \sim (0, Q), \quad v(t) \sim (0, R)\end{aligned}$$

Assumptions

$w(t)$ and $v(t)$ are white noise processes orthogonal to each other and to $x(0)$.

Initialization

$$\hat{x}(0) = \bar{x}_0$$

Error Covariance ARE

$$AP + PA^T + GQG^T - PC^TR^{-1}CP = 0$$

Kalman Gain

$$L = PC^TR^{-1}$$

Estimate Dynamics (Filter Dynamics)

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

The theorem offered in connection with output-injection observer design also holds here. Thus suppose that (C, A) is observable and $(A, G\sqrt{Q})$ is reachable. Then the ARE has a unique positive definite solution P . Moreover, error system (6.4-46) using the gain Kalman gain L given by (6.4-70), with P the unique positive definite solution to the ARE, is asymptotically stable.

One might be inclined to believe that the less noise in the system, the better. However, the actual situation is quite surprising. For the existence of the Kalman filter it was necessary to assume that $R > 0$; that is, that *the measurement noise corrupts all the measurements*. If there are some noise-free measurements, a more complicated filter known as the *Deyst filter* must be used. Moreover, the assumption that $(A, G\sqrt{Q})$ is reachable means that *the process noise should excite all the states*.

Example 6.4-2: Kalman Filter Estimation of Angle of Attack in Gust Noise. The short-period approximation to the F-16 longitudinal dynamics is

$$\dot{x} = Ax + B\delta_e + Gw_g \quad (1)$$

with $x = [\alpha \quad q]^T$, α the angle of attack, q the pitch rate, control input δ_e the elevator deflection, and w_g the vertical wind gust disturbance velocity.

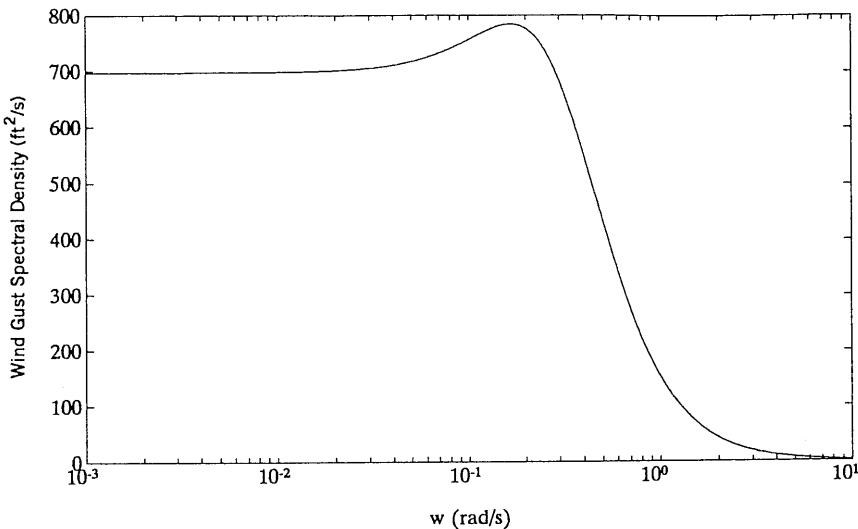


Fig. 6.4-4 Vertical wind gust spectral density.

Using the software described in Chapter 3 to linearize the F-16 dynamics about the nominal flight condition in Table 3.4-1 (true airspeed of 502 ft/s, dynamic pressure of 300 psf, and cg at $0.35\bar{c}$), the plant matrices are found to be

$$A = \begin{bmatrix} -1.01887 & 0.90506 \\ 0.82225 & -1.07741 \end{bmatrix}, \quad B = \begin{bmatrix} -0.00215 \\ -0.17555 \end{bmatrix}, \quad G = \begin{bmatrix} 0.00203 \\ -0.00164 \end{bmatrix}. \quad (2)$$

The vertical wind gust noise is not white, but according to [Mil. Spec. 1797, 1987] has a spectral density given in Dryden form as

$$\Phi_w(\omega) = 2L\sigma^2 \frac{1 + 3L^2\omega^2}{(1 + L^2\omega^2)^2}, \quad (3)$$

with ω the frequency in rad/s, σ the turbulence intensity, and L the turbulence scale length divided by true airspeed. Taking $\sigma = 10$ ft/s and $L = (1750 \text{ ft})/(502 \text{ ft/s}) = 3.49 \text{ s}$ (see Mil. Spec. 1797 [1987]) the gust spectral density is shown in Fig. 6.4-4.

a. *Determination of Gust-Shaping Filter.* Since w_g is not white, a noise shaping filter of the form of (6.4-38), (6.4-39) must be determined by factoring $\Phi_w(s)$ [Lewis, 1986b]. Note that

$$\Phi_w(\omega) = 2L\sigma^2 \frac{(1 + \sqrt{3}Lj\omega)(1 - \sqrt{3}Lj\omega)}{(1 + Lj\omega)^2(1 - Lj\omega)^2}, \quad (4)$$

so that

$$\Phi_w(s) = H_w(s)H_w(-s) \quad (5)$$

with

$$H_w(s) = \sigma \sqrt{\frac{6}{L}} \frac{s + 1/L\sqrt{3}}{L(s + 1/L)^2} \quad (6)$$

$$H_w(s) = \sigma \sqrt{\frac{6}{L}} \frac{s + 1/L\sqrt{3}}{s^2 + 2s/L + 1/L^2}. \quad (7)$$

Now a reachable canonical form realization of $H_w(s)$ [Kailath, 1980] is given by

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{L^2} & -\frac{2}{L} \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \quad (8)$$

$$w_g = \gamma \begin{bmatrix} \frac{1}{L\sqrt{3}} & 1 \end{bmatrix} z, \quad (9)$$

where the gain is $\gamma = \sigma\sqrt{6/L}$. Using $\sigma = 10$, $L = 3.49$ yields

$$\dot{z} = \begin{bmatrix} 0 & 1 \\ -0.0823 & -0.5737 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \equiv A_w z + B_w w \quad (10)$$

$$w_g = [2.1728 \quad 13.1192] z \equiv C_w z. \quad (11)$$

The shaping filter (10)/(11) is a system driven by the *white* noise input $w(t) \sim (0, 1)$ that generates the gust noise $w_g(t)$ with spectral density given by (3).

b. Augmented Plant Dynamics. The overall system, driven by the white noise input $w(t) \sim (0, 1)$ and including an elevator actuator with transfer function $20.2/(s + 20.2)$, is given by [see (6.4-40)]

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \alpha \\ q \\ \hline z_1 \\ z_2 \\ \hline \delta_e \end{bmatrix} &= \left[\begin{array}{cc|cc|c} -1.01887 & 0.90506 & 0.00441 & 0.02663 & -0.00215 \\ 0.82225 & -1.07741 & -0.00356 & -0.02152 & -0.17555 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -0.0823 & -0.5737 & 0 \\ \hline 0 & 0 & 0 & 0 & -20.2 \end{array} \right] \begin{bmatrix} \alpha \\ q \\ \hline z_1 \\ z_2 \\ \hline \delta_e \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ 0 \\ \hline 0 \\ 0 \\ \hline 20.2 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ \hline 0 \\ 0 \\ \hline 1 \end{bmatrix} w \end{aligned} \quad (12)$$

with $u(t)$ the elevator actuator input. To economize on notation, let us symbolize this augmented system as

$$\dot{x} = Ax + Bu + Gw. \quad (13)$$

c. *Estimating Angle of Attack.* Direct measurements of angle of attack α are noisy and biased. However, pitch rate q and normal acceleration n_z are convenient to measure. Using the software in Chapter 3 it is determined that

$$n_z = 15.87875\alpha + 1.48113q. \quad (14)$$

Therefore, let us select the measured output as

$$y = \begin{bmatrix} n_z \\ q \end{bmatrix} = \begin{bmatrix} 15.87875 & 1.48113 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} x + v \equiv Cx + v, \quad (15)$$

where $v(t)$ is measurement noise. A reasonable measurement noise covariance is

$$R = \begin{bmatrix} \frac{1}{20} & 0 \\ 0 & \frac{1}{60} \end{bmatrix}. \quad (16)$$

Now the algebraic Riccati equation in Table 6.4-1 may be solved using standard available software (e.g., ORACLS [Armstrong, 1980; IMSL, 1980], PC-MATLAB [Moler et al., 1987], MATRIX_x [1989]) to obtain the Kalman gain

$$L = \begin{bmatrix} 0.0375 & -0.0041 \\ -0.0202 & 0.0029 \\ 3.5981 & -0.2426 \\ 1.9061 & -0.2872 \\ 0 & 0 \end{bmatrix}, \quad (17)$$

whence the Kalman filter is given by

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly. \quad (18)$$

Note that the Kalman gain corresponding to the fifth state δ_e is zero. This is due to the fact that according to (12), the gust noise $w(t)$ does not excite the actuator motor.

To implement the estimator we could use the state formulation (18) in a subroutine, or we could compute the transfer function to the angle-of-attack estimate given by

$$H_\alpha(s) = [1 \ 0 \ \cdots \ 0][sI - (A - LC)]^{-1}[B \ L]. \quad (19)$$

(Note that α is the first component of x .) Then the angle of attack estimate is given by

$$\hat{\alpha}(s) = H(s) \begin{bmatrix} U(s) \\ Y(s) \end{bmatrix}, \quad (20)$$

so that $\alpha(t)$ may be estimated using $u(t)$ and $y(t)$, both of which are known. Similarly, the estimate of the wind gust velocity $w_g(t)$ may be recovered. ■

Dynamic Regulator Design Using the Separation Principle

The fundamental approach to regulator and compensator design in this book involves selecting the compensator dynamics using the intuition of classical control and traditional aircraft design. Then the adjustable compensator gains are computed using the output feedback design equations in Table 5.3-1, 5.4-1, or 5.5-1. The advantages of this approach include:

1. Good software for solving the design equations is available (e.g., the Davidon–Fletcher–Powell algorithm [Press et al., 1986]). See Appendix B.
3. General multi-input/multi-output controls design is straightforward.
2. If the design is sensible, the closed-loop system is generally stable for any choice of the weighting matrices Q and R .
3. All the intuition in classical controls design in the aircraft industry can be used to select the compensator structure.
4. Complicated compensator structures are avoided, which is important from the point of view of the pilot's feel for the aircraft and also simplifies the gain-scheduling problem.

However, in complicated modern systems (e.g., aircraft engines) there may be no a priori guidelines for selecting the compensator structure. In this case, a combination of LQ state-feedback and observer/filter design proves very useful for controller design. This combination is known as linear quadratic Gaussian (LQG) design, and is explored next. In Section 6.5 we discuss the LQG/LTR technique for robust design, which has become popular in some aspects of aircraft control.

Linear Quadratic Gaussian Design. The linear quadratic regulator (LQR) and the Kalman filter can be used together to design a dynamic regulator. This procedure is called linear quadratic Gaussian (LQG) design, and will now be described. An important advantage of LQG design is that the compensator structure is given by the procedure, so that it need not be known beforehand. This makes LQG design useful in the control of compli-

cated modern-day systems (e.g., space structures, aircraft engines), where an appropriate compensator structure may not be known.

Suppose that the plant and measured output are given by

$$\dot{x} = Ax + Bu + Gw \quad (6.4-74)$$

$$y = Cx + v \quad (6.4-75)$$

with $x(t) \in \mathbf{R}^n$, $u(t)$ the control input, $w(t)$ the process noise, and $v(t)$ the measurement noise. Suppose that the full-state-feedback control

$$u = -Kx + r \quad (6.4-76)$$

has been designed, with $r(t)$ the pilot's input command. That is, the state feedback gain K has been selected by some technique, such as the LQR technique in Section 5.7. If the control (6.4-76) is substituted into (6.4-74) the closed-loop system is found to be

$$\dot{x} = (A - BK)x + Br + Gw. \quad (6.4-77)$$

Full-state-feedback design is attractive because if the conditions in Section 5.7 hold, the closed-loop system is guaranteed stable. Such a strong result has not yet been shown for output feedback. Moreover, using full state feedback all the poles of $(A - BK)$ may be placed arbitrarily as desired. Finally, the state-feedback design equations are simpler than those for output feedback and may be solved using standard available routines (e.g., ORACLS [Armstrong, 1980; IMSL, 1980], PC-MATLAB [Moler et al., 1987], MATTRIX_x [1989]). However, the control law (6.4-76) cannot be implemented since all the states are usually not available as measurements.

Now, suppose that an observer or Kalman filter

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly \quad (6.4-78)$$

has been designed. That is, the filter gain L has been selected by any of the techniques discussed in this section to provide state estimates. Then, since all the states are not measurable and the control (6.4-76) cannot be implemented in practice, we propose to feed back the *estimate* $\hat{x}(t)$ instead of the actual state $x(t)$. That is, let us examine the feedback law

$$u = -K\hat{x} + r. \quad (6.4-79)$$

The closed-loop structure using this controller is shown in Fig. 6.4-5. Due to the fact that the observer is a dynamical system, the proposed controller is nothing but a dynamical regulator of the sort seen in classical control theory. However, in contrast to classical design, the theory makes it easy to design *multivariable regulators with guaranteed stability even for complicated MIMO systems*.

If K is selected using the LQR Riccati equation in Section 5.7 and L is selected using the Kalman filter Riccati equation in Table 6.4-1, this procedure is called LQG design.

We propose to show that using this control:

1. The closed-loop poles are the same as if the full state feedback (6.4-76) had been used.
2. The transfer function from $r(t)$ to $y(t)$ is the same as if (6.4-76) had been used.

The importance of these results is that the state feedback K and the observer gain L may be designed *separately* to yield desired closed-loop plant behavior and observer behavior. This is the *separation principle*, which is at the heart of modern controls design. Two important ramifications of the separation principle are that closed-loop stability is guaranteed and good software is available to solve the matrix design equations that yield K and L .

The Separation Principle. To show the two important results just mentioned, define the estimation error (6.4-45) and examine the error dynamics (6.4-46). In terms of $\tilde{x}(t)$, we may write (6.4-79) as

$$u = -Kx + K\tilde{x} + r \quad (6.4-80)$$

which, when used in (6.4-74), yields

$$\dot{x} = (A - BK)x + BK\tilde{x} + Br + Gw. \quad (6.4-81)$$

Now, write (6.4-81) and (6.4-46) as the augmented system

$$\frac{d}{dt} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} r + \begin{bmatrix} G \\ G \end{bmatrix} w - \begin{bmatrix} 0 \\ L \end{bmatrix} v \quad (6.4-82)$$

$$y = [C \quad 0] \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + v. \quad (6.4-83)$$

This represents the complete dynamics of Fig. 6.4-5.

Since the augmented system is block triangular, the closed-loop characteristic equation is

$$\Delta(s) = |sI - (A - BK)| \cdot |sI - (A - LC)| = 0. \quad (6.4-84)$$

That is, the closed-loop poles are nothing but the plant poles that result by choosing K and the desired observer poles that result by choosing L . Thus the state feedback gain K and observer gain L may be selected separately for desirable closed-loop behavior.

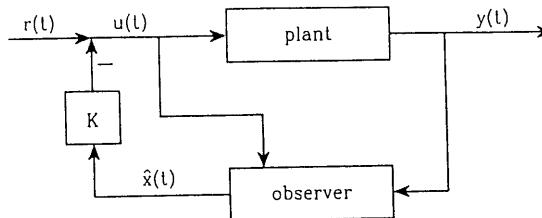


Fig. 6.4-5 Regulator design using observer and full state feedback.

The closed-loop transfer function from $r(t)$ to $y(t)$ is given by

$$H_c(s) = [C \quad 0] \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}^{-1} \begin{bmatrix} B \\ 0 \end{bmatrix},$$

and the triangular form of the system matrix makes it easy to see that

$$H_c(s) = C[sI - (A - BK)]^{-1}B. \quad (6.4-85)$$

This, however, is exactly what results if the full state feedback (6.4-76) is used.

Of course, the initial conditions also affect the output $y(t)$. However, since the observer is stable, the effects of the initial error $\hat{x}(0)$ will vanish with time. The observer poles [i.e., those of $(A - LC)$] should be chosen 5–10 times faster than the desired closed-loop plant poles [i.e., those of $(A - BK)$] for good closed-loop behavior.

Discussion. From our point of view, when possible it is usually better to design compensators using output feedback as we have demonstrated in the past chapters than to use separation principle design. To see why, let us examine the structure of the dynamic compensator in Fig. 6.4-5 in more detail.

The control input $u(t)$ may be expressed as

$$U(s) = H_y(s)Y(s) + H_u(s)U(s) + R(s) \quad (6.4-86)$$

where, according to (6.4-79) and (6.4-78), the transfer function from $y(t)$ to $u(t)$ is

$$H_y(s) = -K[sI - (A - LC)]^{-1}L \quad (6.4-87)$$

and the transfer function from $u(t)$ to $u(t)$ is

$$H_u(s) = -K[sI - (A - LC)]^{-1}B. \quad (6.4-88)$$

Now, note that the compensator designed by this technique has order equal to the order n of the plant. This means that it has too many parameters to be conveniently gain scheduled. Moreover, it has no special structure. This means that none of the classical controls intuition available in the aircraft industry has been used in its design.

It is possible to design *reduced-order* compensators using the separation principle. Three possible approaches are:

1. First find a reduced-order model of the plant, then design a compensator for this reduced-order model.
2. First design a compensator for the full plant, then reduce the order of the compensator.
3. Design the reduced-order compensator directly from the full-order plant.

One technique for order reduction is the partial-fraction-expansion technique in Example 6.2-3. Other techniques include principal component analysis [Moore, 1982] and the frequency-weighted technique in Anderson and Liu [1989]. A very convenient approach is given in Ly et al. [1985].

It is important to realize that although the plant is minimal (i.e., reachable and observable), the LQ regulator may not be. That is, it may have unreachable or unobservable states. A technique for reducing the regulator to minimal form is given in [Yousuff and Skelton, 1984].

In Section 6.5 we illustrate the design of a LQ regulator in robust design using the LQG/LTR approach.

6.5 LQG / LOOP-TRANSFER RECOVERY

We saw in Sections 6.2 and 6.3 how to use the multivariable Bode plot to design controllers guaranteeing performance robustness and stability robustness using output feedback. In Section 6.4 we discussed the Kalman filter. In this section we propose to cover the linear quadratic gaussian/loop-transfer recovery (LQG/LTR) design technique for robust controllers. This approach is quite popular in the current literature and has been used extensively by Honeywell and others to design multivariable aircraft flight control systems [Doyle and Stein, 1981; Athans, 1986]. It is based on the fact that the linear quadratic regulator (LQR) using state-variable feedback has certain *guaranteed robustness properties*.

Thus suppose that a state feedback gain K has been computed using the ARE as in Section 5.7. This state feedback cannot be implemented since all of the states are not available as measurements; however, it can be used as the basis for the design of a dynamic LQ regulator by using a Kalman filter to provide state estimates for feedback purposes. We would like to discuss two

issues. First, we shall show that in contrast to output feedback, state feedback has certain guaranteed robustness properties in terms of gain and phase margins. Then we shall see that the Kalman filter may be designed so that the dynamic regulator recovers the desirable robustness properties of full state feedback.

Guaranteed Robustness of the Linear-Quadratic Regulator

We have discussed conditions for performance robustness and stability robustness for the general feedback configuration of the form shown in Fig. 6.2-1, where $G(s)$ is the plant and $K(s)$ is the compensator. The linear quadratic regulator using *state feedback* has many important properties, as we have seen Section 5.7. In this subsection we should like to return to the LQR to show that it has certain *guaranteed robustness properties* that make it even more useful [Safonov and Athans, 1977].

Thus, suppose that in Fig. 6.2-1, $K(s) = K$, the constant optimal LQ state feedback gain determined using the algebraic Riccati equation (ARE) as in Table 5.7-1. Suppose, moreover, that

$$G(s) = (sI - A)^{-1}B \quad (6.5-1)$$

is a plant in state-variable formulation.

For this subsection, it will be necessary to consider the loop gain *referred to the control input $u(t)$* in Fig. 6.2-1. This is in contrast to the work in Section 6.2, where we referred the loop gain to the output $z(t)$, or equivalently to the signal $s(t)$ in the figure. Breaking the loop at $u(t)$ yields the loop gain

$$KG(s) = K(sI - A)^{-1}B. \quad (6.5-2)$$

Our discussion will be based on the *optimal return difference relation* that holds for the LQR with state feedback [Lewis, 1986a; Grimble and Johnson, 1988; Kwakernaak and Sivan, 1972], namely,

$$\begin{aligned} & [I + K(-sI - A)^{-1}B]^T [I + K(sI - A)^{-1}B] \\ &= I + \frac{1}{\rho} B^T (-sI - A)^{-T} Q (sI - A)^{-1} B, \end{aligned} \quad (6.5-3)$$

where “ $-T$ ” means the inverse transposal. We have selected $R = I$.

Denoting the i th singular value of a matrix M as $\sigma_i(M)$, we note that by definition

$$\sigma_i(M) = \sqrt{\lambda_i(M^*M)}, \quad (6.5-4)$$

with $\lambda_i(M^*M)$ the i th eigenvalue of matrix M^*M and M^* the complex

conjugate transpose of M . Therefore, according to (6.5-3), there results [Doyle and Stein, 1981]

$$\begin{aligned}\sigma_i[I + KG(j\omega)] &= \left[\lambda_i \left[I + \frac{1}{\rho} B^T (-j\omega I - A)^{-T} Q (j\omega I - A)^{-1} B \right] \right]^{1/2} \\ &= \left[1 + \frac{1}{\rho} \lambda_i \left[B^T (-j\omega I - A)^{-T} Q (j\omega I - A)^{-1} B \right] \right]^{1/2}\end{aligned}$$

or

$$\sigma_i[I + KG(j\omega)] = \left[1 + \frac{1}{\rho} \sigma_i^2[H(j\omega)] \right]^{1/2} \quad (6.5-5)$$

with

$$H(s) = H(sI - A)^{-1} B \quad (6.5-6)$$

and $Q = H^T H$.

We could call (6.5-5) the *optimal singular-value relation* of the LQR. It is important due to the fact that the right-hand side is known in terms of open-loop quantities *before the optimal feedback gain is found* by solution of the ARE, while the left-hand side is the closed-loop return difference. Thus, exactly as in classical control, we are able to derive properties of the closed-loop system in terms of properties of the open-loop system.

According to this relation, for all ω the minimum singular value satisfies the *LQ optimal singular-value constraint*

$$\underline{\sigma}[I + KG(j\omega)] \geq 1. \quad (6.5-7)$$

Thus the LQ regulator always results in a *decreased sensitivity*.

Some important conclusions on the guaranteed robustness of the LQR may now be discovered using the *multivariable Nyquist criterion* [Postlethwaite et al., 1981], which we shall refer to the polar plot of the return difference $I + KG(s)$, where the origin is the critical point [Grimble and Johnson, 1988]. [Usual usage is to refer the criterion to the polar plot of the loop gain $KG(s)$, where -1 is the critical point.]

A typical polar plot of $\underline{\sigma}[I + KG(j\omega)]$ is shown in Fig. 6.5-1, where the optimal singular-value constraint appears as the condition that *all the singular values remain outside the unit disc*. To see how the endpoints of the plots were discovered, note that since $K(sI - A)^{-1} B$ has relative degree of at least 1, its limiting value for $s = j\omega$ as $\omega \rightarrow \infty$ is zero. Thus, in this limit, $I + KG(j\omega)$ tends to I . On the other hand, as $\omega \rightarrow 0$, the limiting value of $I + KG(j\omega)$ is determined by the dc loop gain, which should be large.

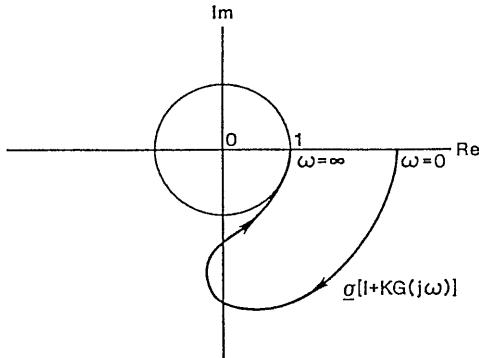


Fig. 6.5-1 Typical polar plot for optimal LQ return difference (referred to the plant input).

The multivariable Nyquist criterion says that the closed-loop system is stable if none of the singular-value plots of $I + KG(j\omega)$ encircle the origin in the figure. Clearly, due to the optimal singular-value constraint, no encirclements are possible. This constitutes a proof of the *guaranteed stability* of the LQR.

Multiplying the optimal feedback K by any positive scalar gain k results in a loop gain of $kKG(s)$, which has a minimum singular value plot identical to the one in Fig. 6.5-1 except that it is scaled outward. That is, the $\omega \rightarrow 0$ limit (i.e., the dc gain) will be larger, but the $\omega \rightarrow \infty$ limit will still be 1. Thus the closed-loop system will still be stable. In classical terms, the LQ regulator has *an infinite gain margin*.

The *phase margin* may be defined for multivariable systems as the angle marked "PM" in Fig. 6.5-2. As in the classical case, it is the angle through

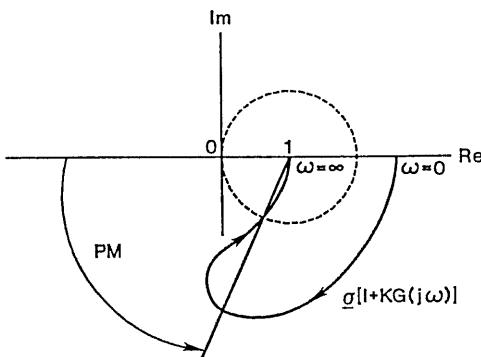


Fig. 6.5-2 Definition of multivariable phase margin.

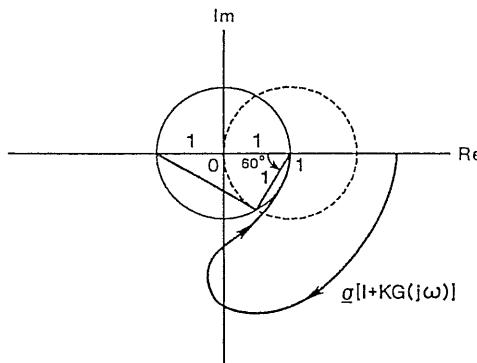


Fig. 6.5-3 Guaranteed phase margin of the LQR.

which the polar plot of $\sigma[I + KG(j\omega)]$ must be rotated (about the point 1) clockwise to make the plot go through the critical point.

Figure 6.5-3 combines Figs. 6.5-1 and 6.5-2. By using some simple geometry, we may find the value of the angle indicated as 60° . Therefore, due to the LQ singular value constraint, the plot of $\sigma[I + KG(j\omega)]$ must be rotated through at least 60° to make it pass through the origin. The LQR thus has a *guaranteed phase margin of at least 60°* .

This means that a phase shift of up to 60° may be introduced in any of the m paths in Fig. 6.2-1, or in all paths simultaneously as long as the paths are not coupled to each other in the process.

This phase margin is excessive; it is higher than that normally required in classical control system design. This overdesign means that in other performance aspects, the LQ regulator may have some deficiencies. One of these turns out to be that at the crossover frequency (loop gain = 1), the slope of the multivariable Bode plot is -20 dB/decade, which is a relatively slow attenuation rate [Doyle and Stein, 1981]. By allowing a Q weighting matrix in the PI that is not positive semidefinite, it is possible to obtain better LQ designs that have higher roll-off rates at high frequencies [Shin and Chen, 1974; Ohta et al., 1990; Al-Sunni et al. 1991].

A stability robustness bound like (6.2-49) may be obtained for the loop gain referred to the input $u(t)$. It is

$$\bar{\sigma}[KG(I + KG)^{-1}] < \frac{1}{m(\omega)}. \quad (6.5-8)$$

The inverse of this is

$$m(\omega) < \frac{1}{\bar{\sigma}[KG(I + KG)^{-1}]} = \underline{\sigma}[I + (KG)^{-1}]. \quad (6.5-9)$$

It can be shown (see the Problems) that (6.5-7) implies that

$$\underline{\sigma} \left[I + (KG(j\omega))^{-1} \right] \geq \frac{1}{2}. \quad (6.5-10)$$

Therefore, the LQR remains stable for all multiplicative uncertainties in the plant transfer function which satisfy $m(\omega) < \frac{1}{2}$.

Loop-Transfer Recovery

The controls design techniques we have discussed in Chapter 5 involve selecting a desirable compensator structure using classical aircraft controls intuition. Then the compensator gains are adjusted using output-feedback design for suitable performance. Robustness may be guaranteed using the multivariable Bode plot as shown in Sections 6.2 and 6.3.

However, in some cases, the plant may be so complex that there is little intuition available for selecting the compensator structure. This can be the case, for instance, for a jet engine [Athans et al., 1986]. In this event, the technique to be presented in this section may be useful for controller design, since it yields a suitable compensator structure automatically.

Let us examine here the plant

$$\dot{x} = Ax + Bu + Gw \quad (6.5-11)$$

$$y = Cx + v, \quad (6.5-12)$$

with process noise $w(t) \sim (0, M)$ and measurement $n(t) \sim (0, \nu^2 N)$ both white, $M > 0$, $N > 0$, and ν a scalar parameter.

We have seen that the full-state-feedback control

$$u = -Kx \quad (6.5-13)$$

has some extremely attractive features, including simplified design equations (Section 5.7) and some important guaranteed robustness properties. Unfortunately, these are not shared by an output-feedback control law, where the robustness must be checked independently. However, state feedback is usually impossible to use since all the states are seldom available for feedback in any practical application.

According to Fig. 6.5-4a, where the plant transfer function is

$$\Phi(s)B = (sI - A)^{-1}B, \quad (6.5-14)$$

the loop gain, breaking the loop at the input $u(t)$, is

$$L_s(s) = K\Phi B. \quad (6.5-15)$$

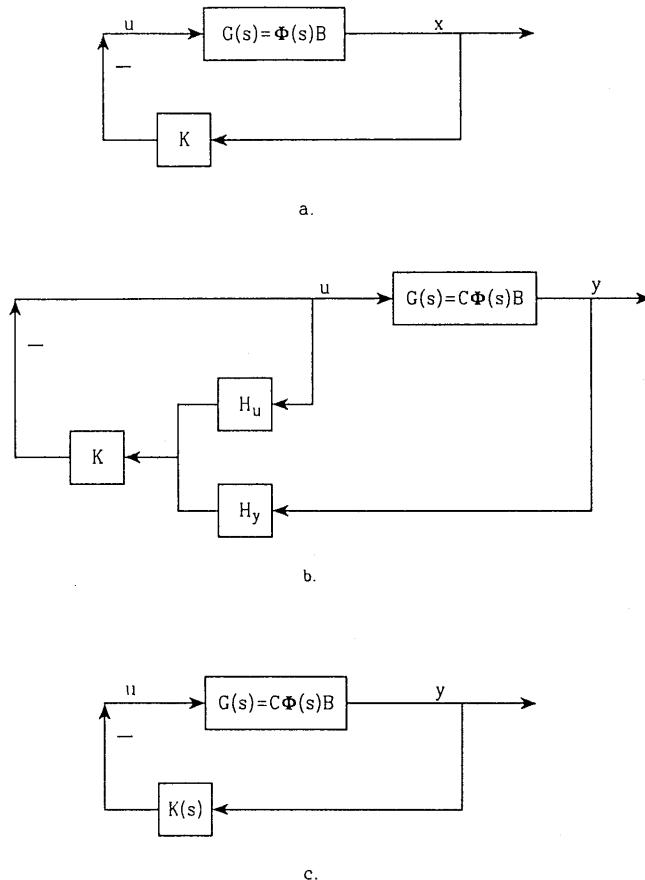


Fig. 6.5-4 (a) Loop gain with full state feedback; (b) regulator using observer and estimate feedback; (c) regulator loop gain.

According to Section 6.4, if an observer or Kalman filter is used to produce a state estimate $\hat{x}(t)$, which is then used in the control law

$$u = -K\hat{x}, \quad (6.5-16)$$

the result is a regulator which, due to the separation principle, has the same transfer function as the state-feedback controller. However, it is known that the guaranteed robustness properties of the full-state-feedback controller are generally lost [Doyle, 1978].

In this section we shall assume that a state-feedback gain K has already been determined using, for instance, the algebraic Riccati equation design technique in Section 5.7. This K yields suitable robustness properties of

$K\Phi B$. We should like to present a technique for designing a Kalman filter that results in a regulator that *recovers* the guaranteed robustness properties of the full-state-feedback control law as the design parameter ν goes to zero. The technique is called LQG/loop-transfer recovery (LQG/LTR), since the loop gain (i.e., loop transfer function) $K\Phi B$ of full state feedback is recovered in the regulator as $\nu \rightarrow 0$. As we shall see, the key to robustness using a stochastic regulator is in the selection of the noise spectral densities M and N .

Regulator Loop Gain. Using an observer or Kalman filter, the closed-loop system appears in Fig. 6.5-4b, where the regulator is given by (Section 6.4)

$$\begin{aligned} U(s) &= -K[sI - (A - LC)]^{-1}BU(s) - K[sI - (A - LC)]^{-1}LY(s) \\ &\equiv -H_u(s)U(s) - H_y(s)Y(s) \end{aligned} \quad (6.5-17)$$

and L is the observer or Kalman gain. Denoting the observer resolvent matrix as

$$\Phi_0(s) = [sI - (A - LC)]^{-1} \quad (6.5-18)$$

we write

$$H_u = K\Phi_0 B, \quad H_y = K\Phi_0 L. \quad (6.5-19)$$

To find an expression for $K(s)$ in Fig. 6.5-4c using the regulator, note that $(I + H_u)U = -H_y Y$, so that

$$U = -(I + H_u)^{-1}H_y Y = -K(s)Y. \quad (6.5-20)$$

However,

$$\begin{aligned} (I + H_u)^{-1}K &= \left[I + K(sI - (A - LC))^{-1}B \right]^{-1}K \\ &= \left[I - K(sI - (A - BK - LC))^{-1}B \right]K \\ &= K(sI - (A - BK - LC))^{-1}[(sI - (A - BK - LC)) - BK] \\ &= K(sI - (A - BK - LC))^{-1}\Phi_0^{-1}, \end{aligned}$$

where the matrix inversion lemma was used in the second step. Therefore,

$$\begin{aligned} K(s) &= (I + H_u)^{-1}H_y \\ &= K[sI - (A - BK - LC)]^{-1}\Phi_0^{-1}\Phi_0 L \end{aligned}$$

or

$$K(s) = K[sI - (A - BK - LC)]^{-1}L \equiv K\Phi_r L, \quad (6.5-21)$$

with $\Phi_r(s)$ the regulator resolvent matrix.

We shall now show how to make the loop gain (at the input) using the regulator

$$L_r(s) = K(s)G(s) = K\Phi_r L C \Phi B \quad (6.5-22)$$

approach the loop gain $L_s(s) = K\Phi B$ using full state feedback, which is guaranteed to be robust.

Recovery of State-Feedback Loop Gain at the Input. To design the Kalman filter so that the regulator loop gain at the input $L_r(s)$ is the same as the state feedback loop gain $L_s(s)$, we shall need to assume that the plant $C\Phi B$ is *minimum phase* (i.e., with stable zeros), with B and C of full rank and $\dim(u) = \dim(y)$. The references for this subsection are Doyle and Stein [1979, 1981]; Athans [1986]; Stein and Athans [1987]; and Birdwell [1989].

Let us propose $G = I$ and the process noise spectral density matrix

$$M = \nu^2 M_0 + BB^T, \quad (6.5-23)$$

with $M_0 > 0$. Then, according to Table 6.4-1,

$$L = PC^T(\nu^2 N)^{-1} \quad (6.5-24)$$

and the Kalman filter ARE becomes

$$0 = AP + PA^T + (\nu^2 M_0 + BB^T) - PC^T(\nu^2 N)^{-1}CP. \quad (6.5-25)$$

According to [Kwakernaak and Sivan, 1972], if the aforementioned assumptions hold, then $P \rightarrow 0$ as $\nu \rightarrow 0$, so that

$$L(\nu^2 N)L^T = PC^T(\nu^2 N)^{-1}CP \rightarrow BB^T.$$

The general solution of this equation is

$$L \rightarrow \frac{1}{\nu} B U N^{-1/2}, \quad (6.5-26)$$

with U any unitary matrix.

We claim that in this situation $L_r(s) \rightarrow L_s(s)$ as $\nu \rightarrow 0$. Indeed, defining the full-state-feedback resolvent as

$$\Phi_c(s) = (sI - (A - BK))^{-1} \quad (6.5-27)$$

we may write

$$\begin{aligned}
 L_r(s) &= K(s)G(s) = K[sI - (A - BK - LC)]^{-1}LC\Phi B \\
 &= K[\Phi_c^{-1} + LC]^{-1}LC\Phi B \\
 &= K[\Phi_c - \Phi_c L(I + C\Phi_c L)^{-1}C\Phi_c]LC\Phi B \\
 &= K\Phi_c L[I - (I + C\Phi_c L)^{-1}C\Phi_c L]C\Phi B \\
 &= K\Phi_c L[(I + C\Phi_c L) - C\Phi_c L](I + C\Phi_c L)^{-1}C\Phi B \\
 &= K\Phi_c L(I + C\Phi_c L)^{-1}C\Phi B \\
 &\rightarrow K\Phi_c B(C\Phi_c B)^{-1}C\Phi B \\
 &= K\Phi B(I + K\Phi B)^{-1}[C\Phi B(I + K\Phi B)^{-1}]^{-1}C\Phi B \\
 &= [K\Phi B(C\Phi B)^{-1}]C\Phi B = K\Phi B. \tag{6.5-28}
 \end{aligned}$$

The matrix inversion lemma was used in going from line 2 to line 3 and from line 7 to 8. The limiting value (6.5-26) for L was used at the arrow.

What we have shown is that using $G = I$ and the process noise given by (6.5-23), as $\nu \rightarrow 0$ the regulator loop gain using a Kalman filter approaches the loop gain using full state feedback. This means that as $\nu \rightarrow 0$, all the robustness properties of the full-state-feedback control law are recovered in the stochastic regulator.

The *LQG/LTR design procedure* is thus as follows:

1. Use the control ARE in Table 5.7-1 to design a state feedback gain K with desirable properties. This may involve iterative design varying the PI weighting matrices Q and R .
2. Select $G = I$, process noise spectral density $M = \nu^2 M_0 + BB^T$, and noise spectral density $\nu^2 N$ for some $M_0 > 0$ and $N > 0$. Fix the design parameter ν and use the Kalman filter ARE to solve for the Kalman gain L .
3. Plot the maximum and minimum singular values of the regulator loop gain $L_r(s)$ and verify that the robustness bounds are satisfied. If they are not, decrease ν and return to 2.

A *reduced-order* regulator with suitable robustness properties may be designed by the LQG/LTR approach using the notions at the end of Section 6.4. That is, either a regulator may be designed for a reduced-order model of the plant, or the regulator designed for the full-order plant may then have its order reduced. In using the first approach, a high-frequency bound characterizing the unmodeled dynamics should be used to guarantee stability robustness.

An interesting aspect of the LQR/LTR approach is that the recovery process may be viewed as a *frequency-domain linear quadratic* technique that trades off the smallness of the sensitivity $S(j\omega)$ and the cosensitivity $T(j\omega)$ at various frequencies. These notions are explored in Stein and Athans [1987] and Safonov et al. [1981].

Non-Minimum-Phase Plants and Parameter Variations. The limiting value of $K(s)$ is given by the bracketed term in (6.5-28). Clearly, as $\nu \rightarrow 0$ the regulator *inverts the plant transfer function* $C\phi B$. If the plant is of minimum-phase, with very stable zeros, the LQG/LTR approach generally gives good results. On the other hand, if the plant is non-minimum-phase or has stable zeros with large time constants, the approach can be unsuitable.

In some applications, however, even if the plant is non-minimum-phase, the LQG/LTR technique can produce satisfactory results [Athans, 1986]. In this situation, better performance may result if the design parameter ν is not nearly zero. If the right-half plane zeros occur at high frequencies where the loop gain is small, the LQG/LTR approach works quite well.

An additional defect of the LQG/LTR approach appears when there are plant parameter variations. As seen in Section 6.2, stability in the presence of parameter variations requires that the loop-gain singular values be below some upper bound at low frequencies. However, this bound is not taken into account in the LQG/LTR derivation. Thus LQG/LTR can yield problems for aircraft controls design, where gain scheduling is required. The H -infinity design approach [Francis et al., 1984; Doyle et al., 1989] has been used with success to overcome this problem.

Recovery of Robust Loop Gain at the Output. We have shown that by designing the state feedback first and then computing the Kalman filter gain using a specific choice of noise spectral densities, the stochastic regulator recovers the robustness of the loop gain $K(s)G(s)$ referred to the input $u(t)$ in Fig. 6.5-4. However, in Section 6.2 we saw that for a small tracking error the robustness should be studied in terms of the loop gain $G(s)K(s)$ referred to the error, or equivalently to the system *output*.

Here we should like to show how to design a stochastic regulator that recovers a robust loop gain $G(s)K(s)$. This yields a second LQG/LTR design algorithm.

Thus, suppose that we *first design a Kalman filter* with gain L using Table 6.4-1. By duality theory, one may see that the Kalman filter loop gain

$$L_k(s) = C\Phi L \quad (6.5-29)$$

enjoys exactly the same guaranteed robustness properties as the state-feedback loop gain $K\Phi B$ that were described earlier in this section.

The regulator loop gain referred to the output is

$$L_r^o(s) = G(s)K(s) = C\Phi BK\Phi_r L. \quad (6.5-30)$$

Thus we should like to determine how to design a state-feedback gain K so that $L_r^o(s)$ approaches $C\Phi L$. As we shall see, the key to this is in the selection of the PI weighting matrices Q and R in Table 5.7-1.

To determine K , let us propose the PI

$$J = \frac{1}{2} \int_0^\infty (x^T Q x + \rho^2 u^T R u) dt \quad (6.5-31)$$

with

$$Q = \rho^2 Q_0 + C^T C, \quad (6.5-32)$$

with $Q_0 > 0$. By using techniques dual to those above, we may demonstrate that as $\rho \rightarrow 0$ the state feedback gain determined using Table 5.7-1 approaches

$$K \rightarrow \frac{1}{\rho} R^{-1/2} W C, \quad (6.5-33)$$

with W a unitary matrix. Using this fact, it may be shown that

$$L_r^o(s) = G(s)K(s) \rightarrow C\Phi L. \quad (6.5-34)$$

The LQG/LTR design technique for loop gain recovery at the output is therefore exactly dual to that for recovery at the input. Specifically, the Kalman gain L is first determined using Table 6.4-1 for desired robustness properties. Then Q and R are selected, with Q of the special form (6.5-32). For a small value of ρ , the state-feedback gain K is determined using Table 5.7-1. If the singular-value Bode plots of $L_r^o(s)$ do not show acceptable robustness, ρ is decreased and a new K is determined.

If the plant $C\Phi B$ is minimum-phase, all is well as ρ is decreased. However, if there are zeros in the right-half plane, there could be problems as ρ becomes too small, although with care the LQG/LTR technique often still produces good results for suitable ρ .

Example 6.5.1: LQG / LTR Design of Aircraft Lateral Control System. We shall illustrate the loop-transfer recovery technique on a lateral aircraft CAS design. This example should be compared with Examples 4.4-3, 4.5-4, 4.6-7, and 5.5-4. All computations, including solving for the state feedback gains and Kalman filter gains, were carried out very easily using MATLAB [Moler et al., 1987].

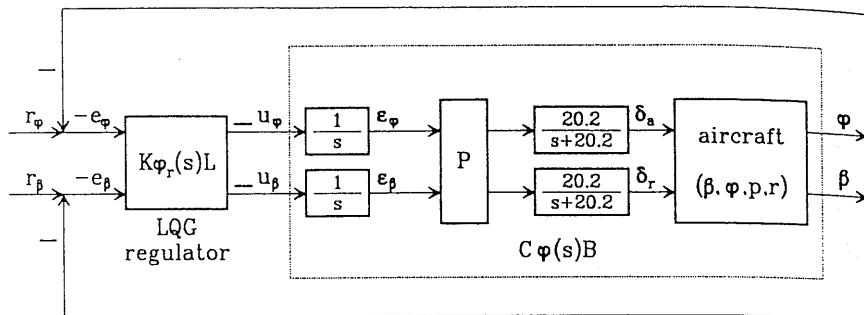


Fig. 6.5-5 Aircraft turn coordinator control system.

a. *Control Objective.* The tracking control system shown in Fig. 6.5-5 is meant to provide coordinated turns by causing the bank angle $\phi(t)$ to follow a desired command while maintaining the sideslip angle $\beta(t)$ at zero. It is a two-channel system with control input $u = [u_\phi \ u_\beta]^T$.

The reference command is $r = [r_\phi \ r_\beta]^T$. The control system should hold ϕ at the commanded value of r_ϕ and $\beta(t)$ at the commanded value of r_β , which is equal to zero. The tracking error is $e = [e_\phi \ e_\beta]^T$ with

$$\begin{aligned} e_\phi &= r_\phi - \phi \\ e_\beta &= r_\beta - \beta. \end{aligned} \tag{1}$$

The negatives of the errors appear in the figure since a minus sign appears in $u = -K\hat{x}$ as is standard for LQG design.

b. *State Equations of Aircraft and Basic Compensator Dynamics.* To obtain the basic aircraft dynamics, the nonlinear F-16 model was linearized at the nominal flight condition in Table 3.4-3 ($V_T = 502$ ft/s, 0 ft altitude, 300 psf dynamic pressure, cg at 0.35c) retaining the states sideslip β , bank angle ϕ , roll rate p , and yaw rate r . Additional states δ_a and δ_r are introduced by the aileron and rudder actuators, both of which are modeled as having approximate transfer functions of $20.2/(s + 20.2)$. The aileron deflection is δ_a and the rudder deflection is δ_r .

The singular values versus frequency of the basic aircraft with actuators are shown in Fig. 6.5-6. Clearly, the steady-state error will be large in closed-loop since the loop gain has neither integrator behavior nor large singular values at dc. Moreover, the singular values are widely separated at dc, so that they are not balanced.

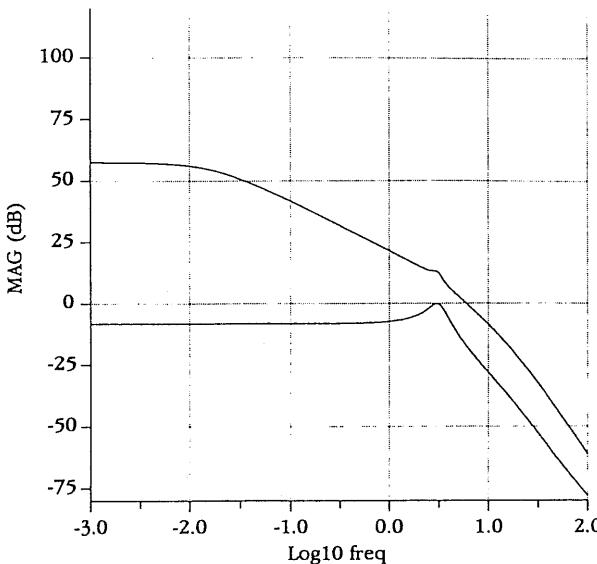


Fig. 6.5-6 Singular values of the basic aircraft dynamics.

To correct these deficiencies we may use the techniques of Example 6.2-3. The dc gain of the system is given by

$$H(0) = \begin{bmatrix} -727.37 & -76.94 \\ -2.36 & 0.14 \end{bmatrix}. \quad (2)$$

First, the dynamics are augmented by integrators in each control channel. We denote the integrator outputs by ϵ_ϕ and ϵ_β . The singular value plots including the integrators are shown in Fig. 6.5-7. The dc slope is now $-20, so that the closed-loop steady-state error will be zero. Next, the system was augmented by $P = H^{-1}(0)$ to balance the singular values at dc. The net result is shown in Fig. 6.5-8, which is very suitable.$

The entire state vector, including aircraft states and integrator states, is

$$x = [\beta \ \phi \ p \ r \ \delta_a \ \delta_r \ \epsilon_\phi \ \epsilon_\beta]^T. \quad (3)$$

The full-state-variable model of the aircraft plus actuators and integrators is of the form

$$\dot{x} = Ax + Bu \quad (4)$$

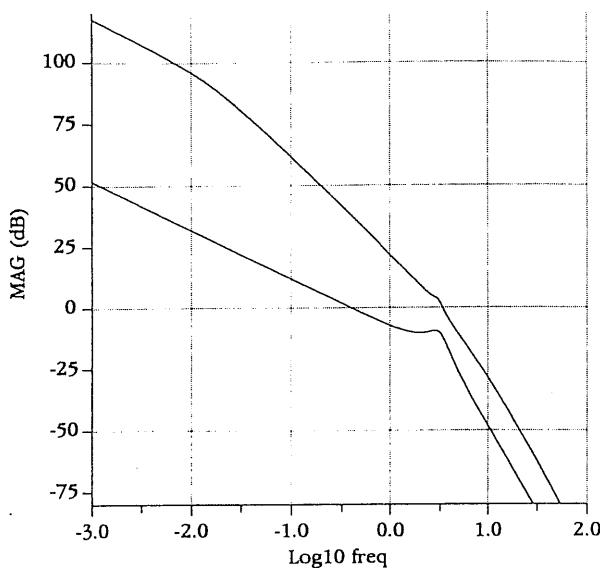


Fig. 6.5-7 Singular values of aircraft augmented by integrators.

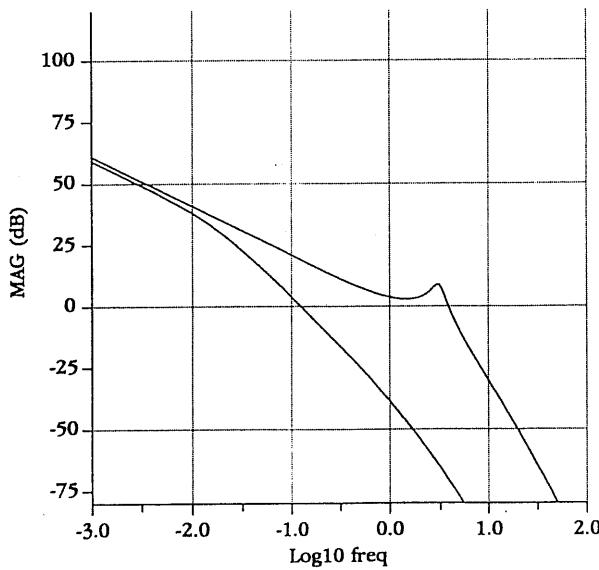


Fig. 6.5-8 Singular values of aircraft augmented by integrators and inverse dc gain matrix P .

with

$$A = \begin{bmatrix} -0.3220 & 0.0640 & 0.0364 & -0.9917 & 0.0003 & 0.0008 & 0 & 0 \\ 0 & 0 & 1 & 0.0037 & 0 & 0 & 0 & 0 \\ -30.6492 & 0 & -3.6784 & 0.6646 & -0.7333 & 0.1315 & 0 & 0 \\ 8.5395 & 0 & -0.0254 & -0.4764 & -0.0319 & -0.0620 & 0 & 0 \\ 0 & 0 & 0 & 0 & -20.2 & 0 & -0.01 & -5.47 \\ 0 & 0 & 0 & 0 & 0 & -20.2 & -0.168 & 51.71 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5)$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (6)$$

The output is given by $y = [\phi \quad \beta]^T$, or

$$y = \begin{bmatrix} 0 & 57.2958 & 0 & 0 & 0 & 0 & 0 & 0 \\ 57.2958 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x = Cx, \quad (7)$$

where the factor of 57.2958 converts radians to degrees. Then

$$e = r - y. \quad (8)$$

c. Frequency-Domain Robustness Bounds. We now derive the bounds on the loop-gain MIMO Bode magnitude plot that guarantee robustness of the closed-loop system. Consider first the high-frequency bound. Let us assume that the aircraft model is accurate to within 10% up to a frequency of 2 rad/s, after which the uncertainty grows without bound at the rate of 20 dB/decade. The uncertainty could be due to actuator modeling inaccuracies, aircraft flexible modes, and so on. This behavior is modeled by

$$m(\omega) = \frac{s+2}{20}. \quad (9)$$

We assume $m(\omega)$ to be a bound on the multiplicative uncertainty in the aircraft transfer function (Section 6.2).

For stability robustness, despite the modeling errors, we saw in Section 6.2 that the loop gain referred to the output should satisfy

$$\bar{\sigma}(GK(j\omega)) < \frac{1}{m(\omega)} = \left| \frac{20}{s+2} \right| \quad (10)$$

when $1/m(\omega) \ll 1$. The function $1/m(\omega)$ is plotted in Fig. 6.5-9.

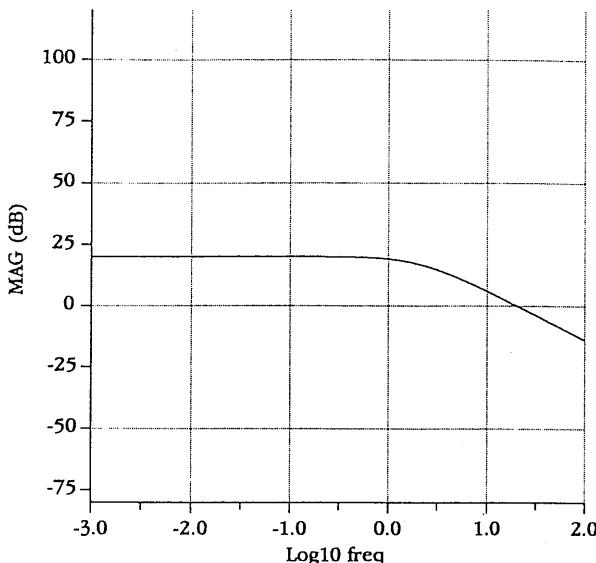


Fig. 6.5-9 Multiplicative uncertainty bound $1/m(\omega)$ for the aircraft dynamical model.

Turning to the low-frequency bound on the closed-loop loop gain, the closed-loop system should be robust to wind gust disturbances. Using techniques like those in Example 6.3-1, the gust magnitude plot shown in Fig. 6.5-10a may be obtained. According to Section 6.2, for robust performance despite wind gusts, the minimum loop-gain singular value $\underline{\sigma}(GK(j\omega))$ should be above this bound.

d. Target Feedback Loop Design. The robustness bounds just derived are expressed in terms of the singular-value plots referred to $e(t)$. To recover the loop gain $GK(j\omega)$ at $e(t)$, or equivalently at the output, the Kalman filter should be designed first, so that we should employ LQG/LTR algorithm 2. Then $(C\Phi(s)L$ is the target feedback loop which should be recovered in the state-feedback design phase.

In standard applications of the LQG/LTR technique, the regulator is designed for robustness, but the time responses are not even examined until the design has been completed. It is difficult to obtain decent time responses using this approach. In this example we should like to emphasize the fact that *it is not difficult to obtain good time responses as well as robustness using LQG/LTR*. It is only necessary to select the Kalman gain L in Table 6.4-1 for good robustness properties as well as *suitable step responses* of the target feedback loop $C\Phi(s)L$, where $\Phi(s) = (sI - A)^{-1}$.

Using MATLAB, the Kalman filter design equations in Table 6.4-1 were solved using

$$Q = \text{diag}\{0.01, 0.01, 0.01, 0.01, 0, 0, 1, 1\}, \quad (11)$$

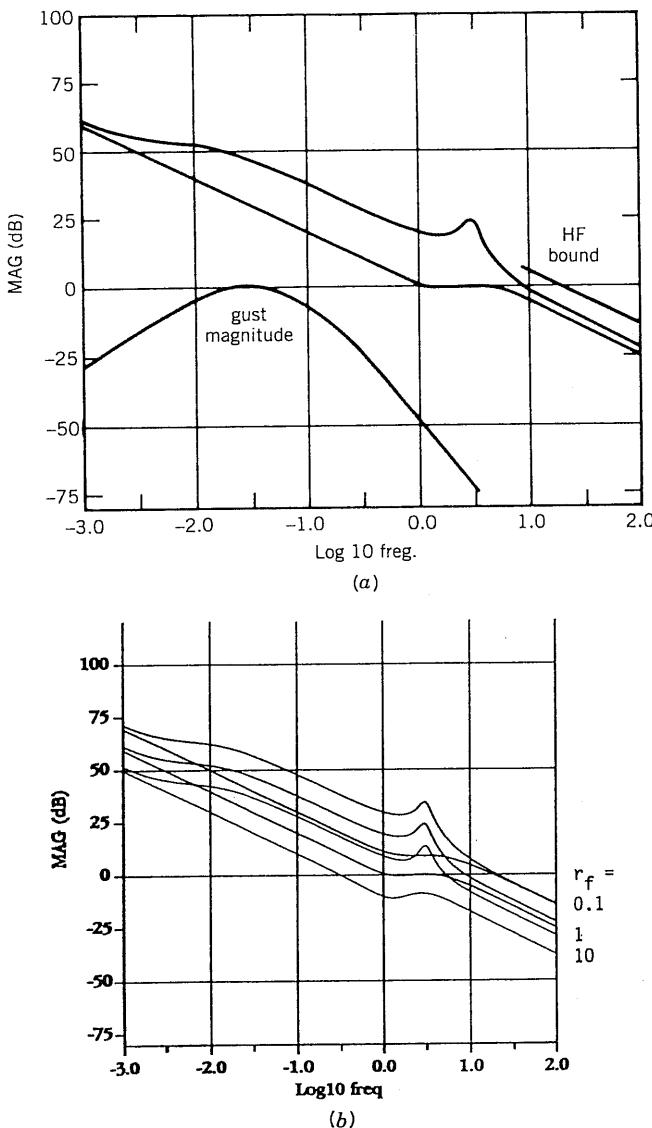


Fig. 6.5-10 Singular values of Kalman filter open-loop gain $C\Phi(s)L$: (a) for $r_f = 1$, including robustness bounds; (b) for various values of r_f .

$R = r_f I$, and various values of r_f . The maximum and minimum singular values of the filter open-loop gain $C\Phi(s)L$ for $r_f = 1$ are shown in Fig. 6.5-10a, which also depicts the robustness bounds. The singular values for several values of r_f are shown in Fig. 6.5-10b. Note how the singular-value magnitudes increase as r_f decreases, reflecting improved rejection of low-frequency disturbances. The figures show that the robustness bounds are

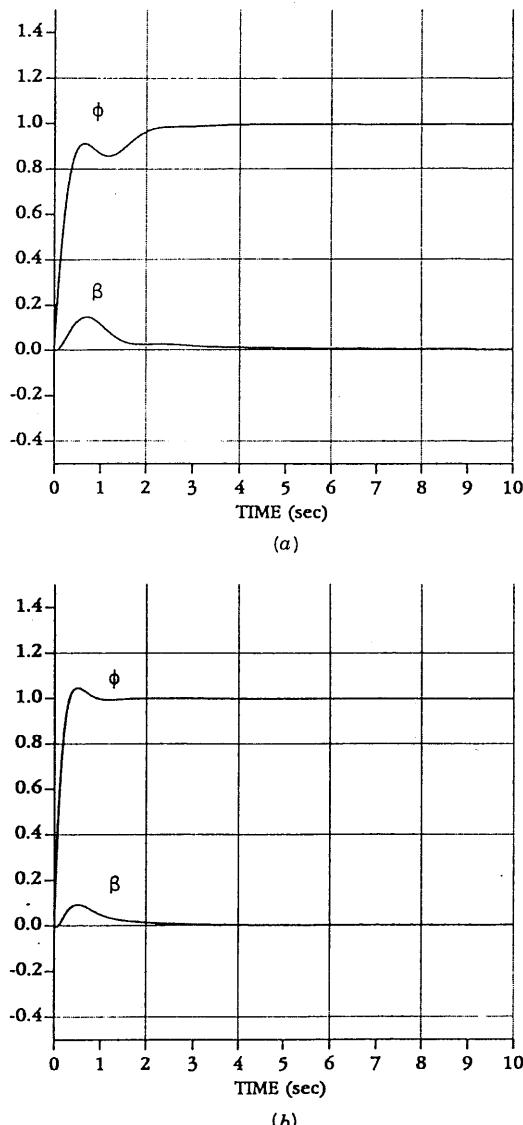


Fig. 6.5-11 Step responses of target feedback loop $C\Phi(s)L$: (a) $r_f = 10$; (b) $r_f = 1$.

satisfied for $r_f = 1$ and $r_f = 10$, but that the high-frequency bound is violated for $r_f = 0.1$.

The associated step responses of $C\Phi(s)L$ with reference commands of $r_\phi = 1$, $r_\beta = 0$ are shown in Fig. 6.5-11. The response for $r_f = 10$ is unsatisfable, while the response for $r_f = 0.1$ is too fast and would not be appreciated by the pilot. On the other hand, the response for $r_f = 1$ shows suitable time

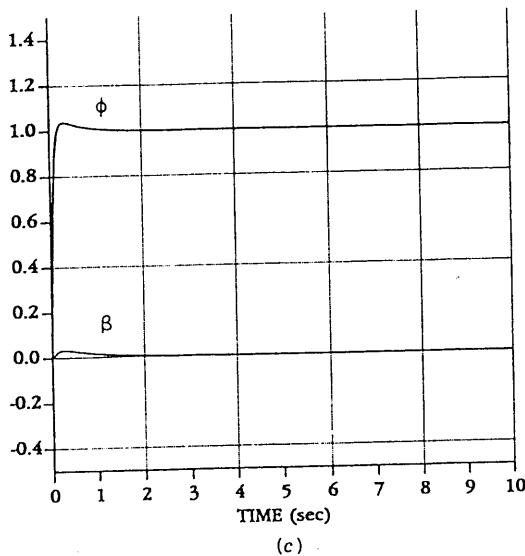


Fig. 6.5-11 (Continued) (c) $r_f = 0.1$.

of response and overshoot characteristics, as well as good decoupling between the bank angle $\phi(t)$ and the sideslip $\beta(t)$.

Therefore, the target feedback loop was selected as $C\Phi(s)L$ with $r_f = 1$, since this results in a design that has suitable robustness properties and step responses. The corresponding Kalman gain is given by

$$L = \begin{bmatrix} -0.007 & 0.097 \\ 0.130 & -0.007 \\ 0.199 & -0.198 \\ -0.093 & -0.020 \\ -0.197 & -0.185 \\ 1.858 & 1.757 \\ 0.685 & -0.729 \\ 0.729 & 0.684 \end{bmatrix}. \quad (12)$$

The Kalman filter poles (e.g., those of $A - LC$) are given by

$$\begin{aligned} s = & -0.002, -0.879, -1.470, \\ & -3.952 \pm j3.589, \\ & -7.205, -20.2, -20.2. \end{aligned} \quad (13)$$

Although there is a slow pole, the step response is good, so this pole evidently has a small residue.

It is of interest to discuss how the frequency and time responses were plotted. For the frequency response, we used the open-loop system

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Le \\ \hat{y} &= C\hat{x},\end{aligned}\tag{14}$$

which has a transfer function of $C\Phi(s)L = C(sI - A)^{-1}L$. A program was written which plots the singular values versus frequency for a system given in state-space form. This yielded Fig. 6.5-10.

For the step response, it is necessary to examine the closed-loop system. In this case, the loop is closed by using $e = r - \hat{y}$ in (14), obtaining

$$\begin{aligned}\dot{\hat{x}} &= (A - LC)\hat{x} + Lr \\ \hat{y} &= C\hat{x}.\end{aligned}\tag{15}$$

Using these dynamics in program TRESP (Appendix B) with $r = [1 \ 0]^T$ produces the step response plot.

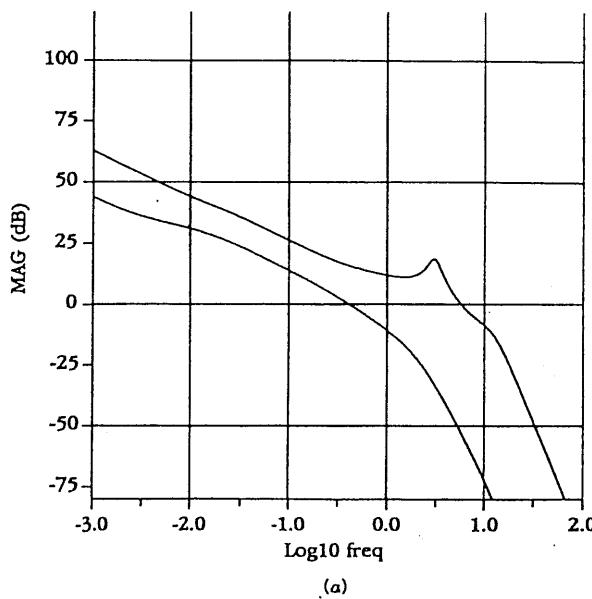
A word on the choice for Q is in order. The design parameters Q and R should be selected so that the target feedback loop $C\Phi(s)L$ has good robustness and time-response properties. It is traditional to select $Q = BB^T$, which accounts for the last two diagonal entries of (11). However, in this example it was impossible to obtain good step responses using this selection for Q . Motivated by the fact that the process noise in the aircraft excites the first four states as well, we experimented with different values for Q , plotting in each case the singular values and step responses. After a few iterations, the final choice (11) was made.

e. Loop Transfer Recovery at the Output. The target feedback loop $C\Phi(s)L$ using $r_f = 1$ has good properties in both the frequency and time domains. Unfortunately, the closed-loop system with LQG regulator has a loop gain referred to the output of $C\Phi(s)BK\Phi_r(s)L$, with the regulator resolvent given by

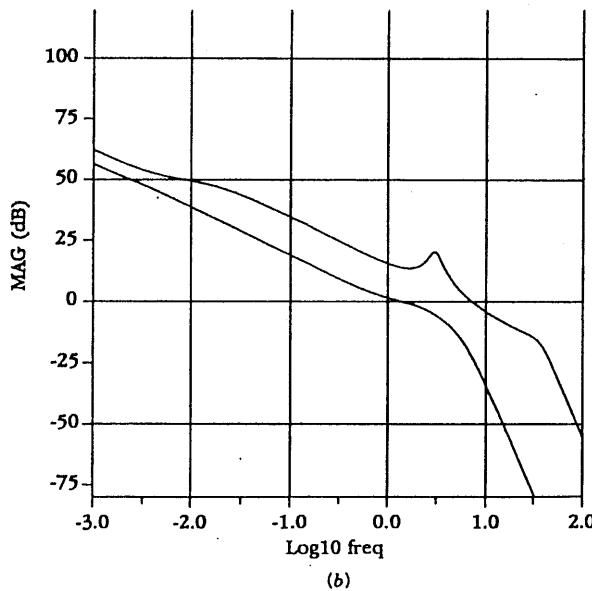
$$\Phi_r(s) = [sI - (A - LC - BK)]^{-1}.\tag{16}$$

On the other hand, LQG/LTR algorithm 2 shows how to select a state-feedback gain K so that the LQG regulator loop gain approaches the ideal loop gain $C\Phi(s)L$. Let us now select such a feedback gain matrix.

Using MATLAB, the LQR design problem in Table 5.7-1 was solved with $Q = C^TC$, $R = \rho^2 I$, and various values of $r_c \equiv \rho^2$ to obtain different feedback gains K . Some representative singular values of the LQG loop gain $C\Phi(s)BK\Phi_r(s)L$ are plotted in Fig. 6.5-12, where L is the target-loop Kalman gain (12). Note how the actual singular values approach the target



(a)



(b)

Fig. 6.5-12 Singular value plots for the LQG regulator: (a) LQG with $r_c = 10^{-3}$; (b) LQG with $r_c = 10^{-7}$.

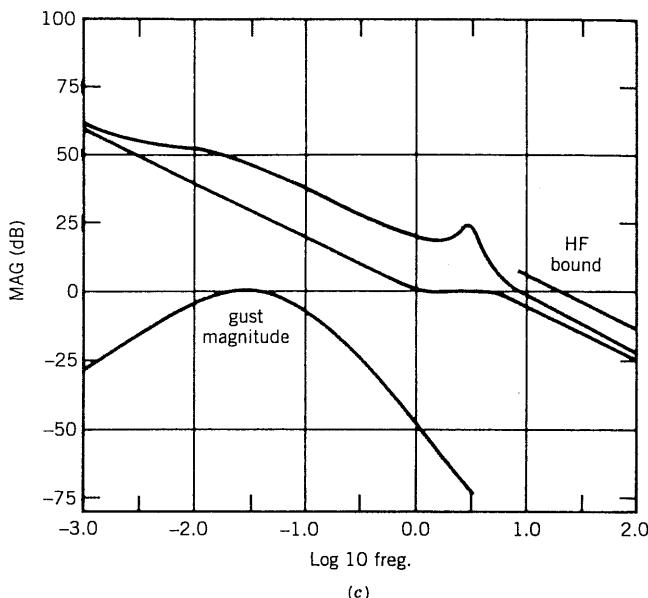


Fig. 6.5-12 (Continued) (c) LQG with $r_c = 10^{-11}$, including robustness bounds.

singular values in Fig. 6.5-10a as r_c decreases. A good match is obtained for $r_c = 10^{-11}$.

Figure 6.5-12c also depicts the robustness bounds, which are satisfied for this choice of $r_c = 10^{-11}$. The corresponding step responses are given in Fig. 6.5-13. A suitable step response that matches well the target response of Fig. 6.5-11b results when $r_c = 10^{-11}$.

It is of interest to discuss how these plots were obtained. For the LQG singular value plots, the complete dynamics are given by

$$\begin{aligned}\dot{x} &= Ax + Bu \\ \dot{\hat{x}} &= (A - LC)\hat{x} + Bu + Lw \\ u &= -K\hat{x},\end{aligned}\tag{17}$$

where $w(t) = -e(t)$. These may be combined into the augmented system

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & -BK \\ 0 & A - LC - BK \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} 0 \\ L \end{bmatrix} w\tag{18}$$

$$y = [C \quad 0] \begin{bmatrix} x \\ \hat{x} \end{bmatrix}\tag{19}$$

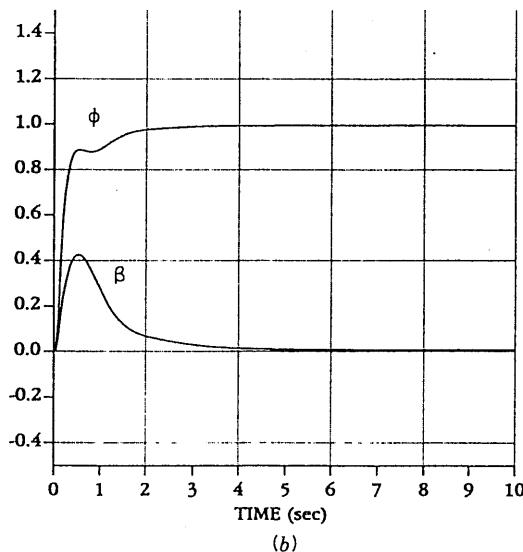
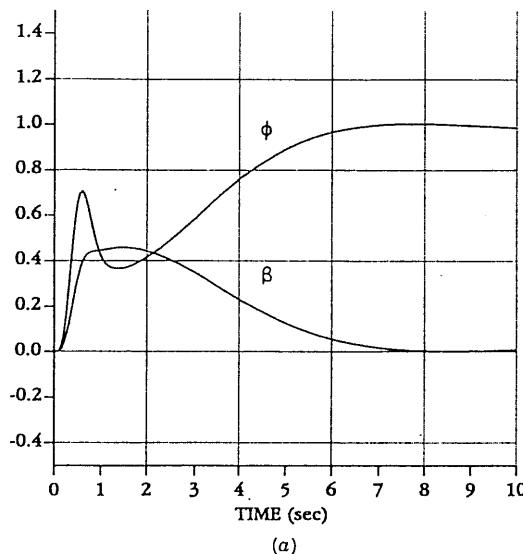


Fig. 6.5-13 Closed-loop step responses of the LQG regulator: (a) LQG with $r_c = 10^{-3}$; (b) LQG with $r_c = 10^{-7}$.

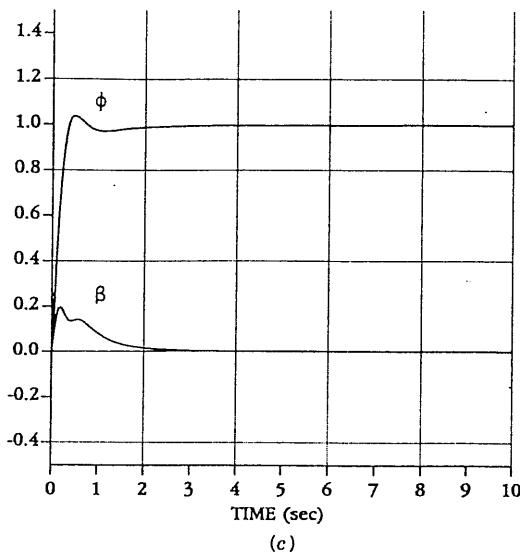


Fig. 6.5-13 (Continued) (c) LQG with $r_c = 10^{-11}$.

which has transfer function $C\Phi(s)BK\Phi_r(s)L$. The singular values are now easily plotted.

For the step responses, the closed-loop system must be studied. To close the loop, set $w = y - r$ in (18) to obtain the closed-loop dynamics

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & -BK \\ LC & A - LC - BK \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} 0 \\ -L \end{bmatrix} r \quad (20)$$

$$y = [C \quad 0] \begin{bmatrix} x \\ \hat{x} \end{bmatrix}. \quad (21)$$

These are used with program TRESP in Appendix B to obtain Fig. 6.5-13.

The final LQG regulator is given by the Kalman gain L in (12) and the feedback gain K corresponding to $r_c = 10^{-11}$.

f. Reduced-Order Regulator. The LQG regulator just designed has order $n = 8$, the same as the plant. This is excessive for an aircraft lateral control system. A reduced-order regulator that produces very good results may easily be determined using the partial-fraction-expansion approach in Example 6.2-4, principal component analysis [Moore, 1982], or other techniques. This is easily accomplished using MATLAB. The singular value plots and step response using the reduced-order regulator should be examined to verify robustness and suitable performance. ■

6.6 SUMMARY

In Section 6.2 we extended some classical frequency-domain analysis techniques to multivariable systems using the notion of the singular value. We defined the multivariable loop gain, return difference, and sensitivity, and showed that the multivariable Bode magnitude plot is just the plot of the maximum and minimum singular values of the loop gain versus frequency. To guarantee stability robustness to unmodeled high-frequency dynamics and plant parameter variations, as well as performance robustness in the presence of disturbances, we derived various frequency-domain bounds that the singular values of the loop gain must satisfy.

A convenient robust output-feedback design approach was presented in Section 6.3 that results in acceptable time-domain performance with guaranteed robustness.

In Section 6.4 we covered the design of multivariable observers for estimating the full state of the aircraft model from the measured outputs. We discussed the Kalman filter, showing an example of its use by reconstructing the angle of attack from normal acceleration and pitch-rate measurements in the presence of wind-gust noise. We showed how to use full state feedback and an observer to design a dynamic regulator.

Finally, in Section 6.5 we covered the popular LQG/LTR robust design technique, illustrating with the design of a multivariable lateral control system.

REFERENCES

- Al-Sunni, F. M., B. L. Stevens, and F. L. Lewis, "Negative state weighting in the linear quadratic regulator for aircraft control," *J. Guid. Control Dynam.*, 1991 (to appear).
- Anderson, B. D. O., and Y. Liu, "Controller reduction: concepts and approaches," *IEEE Trans. Autom. Control*, vol. AC-34, no. 8, Aug. 1989, pp. 802-812.
- Armstrong, E. S., *ORACLS: A Design System for Linear Multivariable Control*, New York: Marcel Dekker, 1980.
- Athans, M., "A tutorial on the LQG/LTR method," *Proc. Am. Control Conf.*, June 1986, pp. 1289-1296.
- Athans, M., P. Kapsouris, E. Kappos, and H. A. Spang III, "Linear-quadratic gaussian with loop-transfer recovery methodology for the F-100 engine," *J. Guid.*, vol. 9, no. 1, Jan.-Feb. 1986, pp. 45-52.
- Birdwell, J. D., "Evolution of a design methodology for LQG/LTR," *IEEE Control Syst.*, Apr. 1989, pp. 73-77.
- Blakelock, J. H., *Automatic Control of Aircraft and Missiles*. New York: Wiley, 1965.
- Bryson, A. E., Jr., and Y.-C. Ho, *Applied Optimal Control*. New York: Hemisphere, 1975.
- Doyle, J. C., "Guaranteed margins for LQG regulators," *IEEE Trans. Autom. Control*, Aug. 1978, pp. 756-757.

- Doyle, J. C., and G. Stein, "Robustness with observers," *IEEE Trans. Autom. Control*, vol. AC-24, no. 4, Aug. 1979, pp. 607-611.
- Doyle, J. C., and G. Stein, "Multivariable feedback design: concepts for a classical/modern synthesis," *IEEE Trans. Autom. Control*, vol. AC-26, no. 1, Feb. 1981, pp. 4-16.
- Doyle, J. C., K. Glover, P. P. Khargonekar, and B. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Trans. Autom. Control*, vol. AC-34, no. 8, Aug. 1989, pp. 831-847.
- Francis, B., J. W. Helton, and G. Zames, " H_∞ optimal feedback controllers for linear multivariable systems," *IEEE Trans. Autom. Control*, vol. AC-29, no. 10, Oct. 1984, pp. 888-900.
- Franklin, G. F., J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Reading, Mass.: Addison-Wesley, 1986.
- Gelb, A., ed., *Applied Optimal Estimation*. Cambridge, Mass.: MIT Press, 1974.
- Grimble, M. J., and M. A. Johnson, *Optimal Control and Stochastic Estimation: Theory and Applications*, Vol. 1. New York: Wiley, 1988.
- IMSL, *Library Contents Document*, 8th ed., International Mathematical and Statistical Libraries, Inc., 7500 Bellaire Blvd., Houston, TX 77036, 1980.
- Kailath, T., *Linear Systems*. Englewood Cliffs, N.J.: Prentice-Hall, 1980.
- Kaminer, I., P. P. Khargonekar, and G. Robel, "Design of localizer capture and track modes for a lateral autopilot using H -infinity synthesis," *IEEE Control Syst.*, vol. 10, no. 4, June 1990, pp. 13-21.
- Kwakernaak, H., and R. Sivan, *Linear Optimal Control Systems*. New York: Wiley, 1972.
- Laub, A. J., "An inequality and some computations related to the robust stability of linear dynamic systems," *IEEE Trans. Autom. Control*, vol. AC-24, no. 2, Apr. 1979, pp. 318-320.
- Laub, A. J., "Efficient multivariable frequency response computations," *IEEE Trans. Automat. Control*, vol. AC-26, no. 2, Apr. 1981, pp. 407-408.
- Lewis, F. L., *Optimal Control*. New York: Wiley, 1986a.
- Lewis, F. L., *Optimal Estimation*. New York: Wiley, 1986b.
- Ly, U.-L., A. E. Bryson, and R. H. Cannon, "Design of low-order compensators using parameter optimization," *Automatica*, vol. 21, no. 3, 1985, pp. 315-318.
- MacFarlane, A. G. J., "Return-difference and return-ratio matrices and their use in the analysis and design of multivariable feedback control systems," *Proc. IEE*, vol. 117, no. 10, Oct. 1970, pp. 2037-2049.
- MacFarlane, A. G. J., and B. Kouvaritakis, "A design technique for linear multivariable feedback systems," *Int. J. Control.*, vol. 25, 1977, pp. 837-874.
- MATRIX_x, Integrated Systems, Inc., 2500 Mission College Blvd., Santa Clara, CA 95054, 1989.
- Mil. Spec. 1797*, "Flying qualities of piloted vehicles," 1987.
- Minto, K. D., J. H. Chow, and J. W. Beseler, "An explicit model-matching approach to lateral-axis autopilot design," *IEEE Control Syst.*, vol. 10, no. 4, June 1990, pp. 22-28.

- Moerder, D. D., and A. J. Calise, "Convergence of a numerical algorithm for calculating optimal output feedback gains," *IEEE Trans. Autom. Control*, vol. AC-30, no. 9, Sept. 1985, pp. 900-903.
- Moler, C., J. Little, and S. Bangert, *PC-Matlab*, The Mathworks, Inc., 20 North Main St., Suite 250, Sherborn, MA 01770, 1987.
- Moore, B. C., "Principal component analysis in linear systems: controllability, observability, and model reduction," *IEEE Trans. Autom. Control*, vol. AC-26, no. 1, 1982, pp. 17-32.
- Morari, M., and E. Zafiriou, *Robust Process Control*. Englewood Cliffs, N.J.: Prentice Hall, 1989.
- Ohta, H., M. Nakinuma, and P. N. Nikiforuk, "Use of negative weights in linear quadratic regulator synthesis," preprint, 1990.
- Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, 2nd ed. New York: McGraw-Hill, 1984.
- Postlethwaite, I., J. M. Edmunds, and A. G. J. MacFarlane, "Principal gains and principal phases in the analysis of linear multivariable systems," *IEEE Trans. Autom. Control*, vol. AC-26, no. 1, Feb. 1981, pp. 32-46.
- Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*. New York: Cambridge University Press, 1986.
- Rosenbrock, H. H., *Computer-Aided Control System Design*. New York: Academic Press, 1974.
- Safonov, M. G., and M. Athans, "Gain and phase margin for multiloop LQG regulators," *IEEE Trans. Autom. Control*, vol. AC-22, no. 2, Apr. 1977, pp. 173-178.
- Safonov, M. G., A. J. Laub, and G. L. Hartmann, "Feedback properties of multivariable systems: the role and use of the return difference matrix," *IEEE Trans. Autom. Control*, vol. AC-26, no. 1, Feb. 1981, pp. 47-65.
- Shin, V., and C. Chen, "On the weighting factors of the quadratic criterion in optimal control," *Int. J. Control*, vol. 19, May 1974, pp. 947-955.
- Stein, G., and M. Athans, "The LQG/LTR procedure for multivariable feedback control design," *IEEE Trans. Autom. Control*, vol. AC-32, no. 2, Feb. 1987, pp. 105-114.
- Stevens, B. L., P. Vesty, B. S. Heck, and F. L. Lewis, "Loop shaping with output feedback," *Proc. Am. Control Conf.*, June 1987, pp. 146-149.
- Strang, G., *Linear Algebra and Its Applications*, 2nd ed. New York: Academic Press, 1980.
- Yousuff, A., and R. E. Skelton, "A note on balanced controller reduction," *IEEE Trans. Autom. Control*, vol. AC-29, no. 3, Mar. 1984, pp. 254-257.

PROBLEMS

Section 6.2

- 6.2-1** Derive in detail the multivariable expressions (6.2-16) and (6.2-17) for the performance output and the tracking error.

- 6.2-2** Prove (6.2-54). You will need to neglect any terms that contain second-order terms in the parameter variation matrices and use the fact that for small X , $(I - X)^{-1} \approx (I + X)$.

- 6.2-3** **Multivariable Closed-Loop Transfer Relations.** In Fig. 6.2-1, let the plant $G(s)$ be described by

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \end{bmatrix}x + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}u, \quad z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}x$$

and the compensator is $K(s) = 2I_2$.

- (a) Find the multivariable loop gain and return difference.
 - (b) Find the sensitivity and cosensitivity.
 - (c) Find the closed-loop transfer function from $r(t)$ to $z(t)$, and hence the closed-loop poles.
- 6.2-4** For the continuous-time system in Example 6.2-1, plot the individual SISO Bode magnitude plots from input 1 to outputs 1 and 2, and from input 2 to outputs 1 and 2. Compare them to the MIMO Bode plot to see that there is no obvious relation. Thus the robustness bounds cannot be given in terms of the individual SISO Bode plots.

- 6.2-5** **Software for MIMO Bode Magnitude Plot.** Write a computer program to plot the Bode magnitude plot for a multivariable system given in state-space form $\dot{x} = Ax + Bu$, $y = Cx + Du$. Your program should read in A, B, C, D . You may use a SVD routine (e.g., IMSL [1980] and Press et al. [1986]) or the technique in Laub [1981]. Use the software to verify Examples 6.2-1 and 6.2-2.

- 6.2-6** **Multivariable Bode Plot.** For the system in Problem 6.2-3, plot the multivariable Bode magnitude plots for:
- (a) The loop gain GK
 - (b) The sensitivity S and cosensitivity T . For which frequency ranges do the plots for $GK(j\omega)$ match those for $S(j\omega)$? For $T(j\omega)$?

- 6.2-7** **Bode Plots for F-16 Lateral Regulator.** Plot the loop gain multivariable Bode magnitude plot for the F-16 lateral regulator designed in Example 5.3-1.

- 6.2-8** **Balancing and Zero Steady-State Error.** Find a precompensator for balancing the SVs at low frequency and ensuring zero steady-state error for the system

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ -2 & -3 & 0 \\ 0 & 0 & -3 \end{bmatrix}x + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}u, \quad z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}x.$$

Plot the SVs of the original and precompensated system.

Section 6.3

6.3-1 Model Reduction and Neglected High-Frequency Modes. An unstable system influenced by high-frequency parasitics is given by

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & -10 \end{bmatrix}x + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}u, \quad z = [1 \ 0 \ 0]x.$$

- (a) Use the technique of Example 6.2-4 to find a reduced-order model that neglects the high-frequency mode at $s = 10$ rad/s. Find the bound $m(j\omega)$ on the magnitude of the neglected portion.
- (b) Using techniques like those in Sections 5.4 and 5.5, design a servo control system for the reduced-order model. Try a lead compensator whose gains are varied by the LQ algorithm, as used in Example 5.5-5. Verify the step response of the closed-loop system by performing a simulation on the reduced-order system.
- (c) Find the loop gain of the closed-loop system and plot its singular values. Do they fall below the bound $1/m(j\omega)$, thus guaranteeing robustness to the neglected mode? If not, return to part (b) and find other gains that do guarantee stability robustness.
- (d) Simulate your controller on the full system including the high-frequency mode. How does the step response look?
- (e) A better controller results if high-frequency dynamics are not neglected in the design stage. Design a servo control system for the full third-order system. It may be necessary to use a more complicated controller. Verify the step response of the closed-loop system by performing a simulation. Compare to the results of part (d).

6.3-2 Gain Scheduling Robustness. In the problems for Section 5.4 a gain-scheduled normal acceleration CAS was designed for a transport aircraft using three equilibrium points. Using the results at the end of Section 6.2, we want to check the design for robustness to plant parameter variations. Call the systems at the three equilibrium points (A_i, B_i, C_i) , $i = 1, 2, 3$.

- (a) In Problem 6.2-5 you wrote a program to plot the MIMO Bode magnitude plots for a state-variable system. Note that a state-space realization of $\Delta G(s)$ in (6.2-54) is given by

$$\dot{x} = \begin{bmatrix} A & -\Delta A \\ 0 & A \end{bmatrix}x + \begin{bmatrix} \Delta B \\ B \end{bmatrix}u, \quad y = [C \ \Delta C]x.$$

That is, this system has transfer function of ΔG . Define $\Delta G_{ij}(s)$ as being computed using $\Delta A = A_i - A_j$, $\Delta B = B_i - B_j$, $\Delta C = C_i - C_j$. Use these facts combined with (6.2-48) to obtain low-frequency

bounds for robustness to the gain-scheduling plant parameter variations.

- (b) Find the loop-gain singular values of your design for the gain-scheduled CAS. Do they fall below the robustness bounds? If not, select new PI weights and try to improve the design. If this fails, you will need to select more closely spaced equilibrium points for the gain-scheduled design.

Section 6.4

- 6.4-1 Nonzero-Mean Noise.** Use (6.4-49) to write down the best estimate for $x(t)$ in terms of the filter state $\hat{x}(t)$ if the process noise $w(t)$ and measurement noise $v(t)$ have nonzero means of \bar{w} and \bar{v} , respectively.
- 6.4-2 Observer for Angle of Attack.** In Example 5.5-3 a low-pass filter of $10/(s + 10)$ was used to smooth out the angle-of-attack measurements to design a pitch rate CAS. An alternative is to use an observer to reconstruct α . This completely avoids measurements of the angle of attack.
- (a) Considering only the 2×2 short period approximation, design an observer that uses measurements of $q(t)$ to provide estimates of $\alpha(t)$. The observer should have $\zeta = 1/\sqrt{2}$ and $\omega_n = 10$ rad/s. Use Ackermann's formula to find the output-injection matrix L .
 - (b) Delete the α filter in Example 5.5-3, replacing it by the dynamics of the second-order observer just designed. With the new augmented dynamics, perform the LQ design of Example 5.5-3. Compare the performance of this pitch-rate CAS to the one using the α filter.
- 6.4-3 Dynamic LQ Regulator for Pitch-Rate CAS.** In Example 5.5-3 and Problem 6.4-2, output-feedback design was used to build a pitch-rate CAS. In this problem we would like to use LQG theory to perform the design.
- (a) Design an observer for α using q measurements, as described in the previous problem.
 - (b) Neglect the elevator actuator, considering only the 2×2 short period approximation in Example 5.5-3 plus the feedforward-path integrator. Find the state-feedback gain K to place the poles at $\zeta = 1/\sqrt{2}$, $\omega_n = 3.5$ rad/s; this yields good flying qualities for the short-period mode. Use Ackermann's formula, or the design software for Table 5.3-1 with $C = I$.
 - (c) Using the 2×2 observer and the state feedback K , construct a dynamic pitch-rate CAS. Verify its performance by plotting the step response.

- 6.4-4 Kalman Filter.** Software for solving the Kalman filter ARE is available in Armstrong [1980] and IMSL [1980]; also MATRIX_x [1989] and MATLAB [Moler et al., 1987]. Alternatively, the Kalman filter gain L can be found using the software for Table 5.3-1 on the dual plant (A^T, C^T, B^T) with $B = I$. Repeat Example 6.4-2 if the wind gusts have a turbulence intensity of 20 ft/s.

Section 6.5

- 6.5-1** Show that (6.5-7) implies (6.5-10) (see [Laub, 1979]).
- 6.5-2 LQG / LTR Design.** Note that the state-feedback gain K can be found using the software for Table 5.3-1 with $C = I$. Likewise, the Kalman filter gain L can be found using the software for Table 5.3-1 on the dual plant (A^T, C^T, B^T) , with $B = I$.
- In Problem 6.4-3(b), plot the loop-gain singular values assuming full state feedback.
 - Now angle-of-attack measurements are not allowed. Design a Kalman filter for various values of the design parameter ν . In each case, plot the closed-loop step response as well as the loop-gain singular values. Compare the step response and the SVs to the case for full state feedback as ν becomes small.

CHAPTER 7

DIGITAL CONTROL

7.1 INTRODUCTION

In Chapters 4 through 6 we have shown how to design continuous-time controllers for aircraft. However, with microprocessors so fast, light, and economical, control laws are usually implemented on modern aircraft in digital form. In view of the requirement for gain scheduling of aircraft controllers, digital control schemes are especially useful, for gain scheduling is very easy on a digital computer.

To provide reliability in the event of failures, modern aircraft control schemes are redundant, with two or three control laws for each application. The actual control to be applied is selected by "voting"; that is, there should be good agreement between two out of three controllers. Such schemes are more conveniently implemented on a microprocessor, where the comparison and voting logic resides.

In this chapter we address the design of digital, or discrete-time, controllers, since the design of such controllers involves some extra considerations of which one should be aware. In Section 7.2 we discuss the *simulation* of digital controllers on a digital computer. Then in Sections 7.3 and 7.4 two approaches to digital controls *design* are examined. Finally, some aspects of the actual *implementation* are mentioned in Section 7.5.

In the first approach to digital controls design, covered in Section 7.3, we show how to convert an already designed continuous-time controller to a discrete-time controller using, for instance, the bilinear transform (BLT). An advantage of this *continuous controller redesign* approach is that the sample period T does not have to be selected until after the continuous controller has been designed.

Unfortunately, controller discretization schemes based on transformations such as the BLT are approximations. Consequently, the sampling period T must be small to ensure that the digital controller performs like the continuous version from which it was designed. Therefore, in Section 7.4 we show how the design of the continuous-time controller may be *modified* to take into account some properties of the sampling process, as well as computation delays. Discretization of such a modified continuous controller yields a digital control system with improved performance.

In Section 7.5 we discuss some implementation considerations, such as actuator saturation and controller structure.

There are many excellent references on digital control; some of them are listed at the end of the chapter. We shall draw most heavily on Franklin and Powell [1980], Åström and Wittenmark [1984], and Lewis [1992].

7.2 SIMULATION OF DIGITAL CONTROLLERS

A digital control scheme is shown in Fig. 7.2-1. The plant $G(s)$ is a continuous-time system, and $K(z)$ is the dynamic digital controller, where s and z are, respectively, the Laplace and Z -transform variables (i.e., $1/s$ represents integration and z^{-1} represents a unit time delay). The digital controller $K(z)$ is implemented using software code in a microprocessor.

The hold device in the figure is a digital-to-analog (D/A) converter that converts the discrete control samples u_k computed by the software controller $K(z)$ into the continuous-time control $u(t)$ required by the plant. It is a *data reconstruction* device. The input u_k and output $u(t)$ for a *zero-order hold* (*ZOH*) are shown in Fig. 7.2-2. Note that $u(kT) = u_k$, so that $u(t)$ is continuous from the right. That is, $u(t)$ is updated at times kT . The sampler with sample period T is an analog-to-digital (A/D) converter that takes the samples $y_k = y(kT)$ of the output $y(t)$ that are required by the software controller $K(z)$.

In this chapter we discuss the design of the digital controller $K(z)$. Once the controller has been designed, it is important to *simulate* it before it is implemented to determine if the closed-loop response is suitable. The simulation should provide the response at all times, including times between the samples.

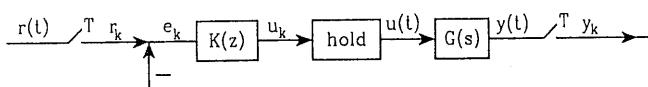


Fig. 7.2-1 Digital controller.

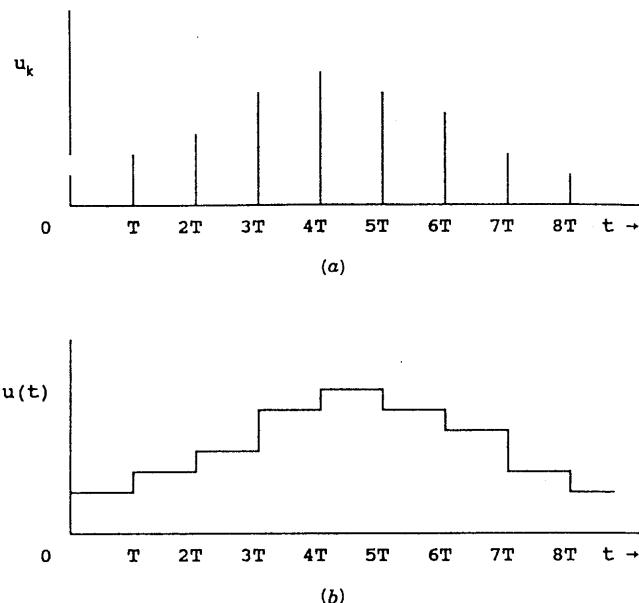


Fig. 7.2-2 Data reconstruction using a ZOH: (a) discrete control sequence u_k ; (b) reconstructed continuous signal $u(t)$.

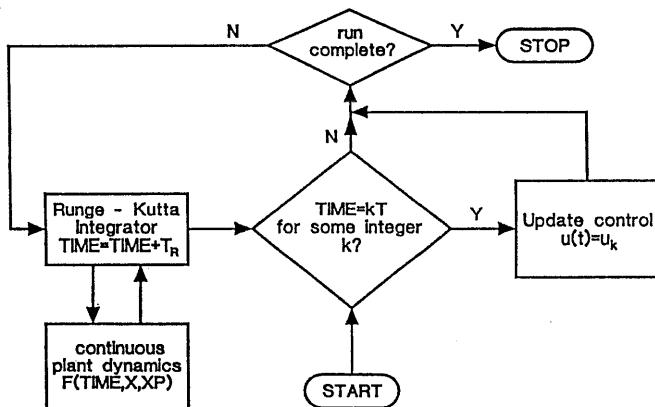


Fig. 7.2-3 Digital control simulation scheme.

To simulate a digital controller we may use the scheme shown in Fig. 7.2-3. There the continuous dynamics $G(s)$ are contained in the subroutine $F(t, x, \dot{x})$; they are integrated using a Runge-Kutta integrator. Note that two time intervals are involved: the sampling period T and the *Runge-Kutta integration period* $T_R \ll T$. T_R should be selected as an integral divisor of T .

Several numerical integration schemes were discussed in Section 3.5. We have found that the Runge-Kutta routines are very suitable, while Adams-Basforth routines do not give enough accuracy for digital controls purposes. This is especially true when advanced adaptive and parameter estimation techniques are used. For most purposes, the fixed step size Runge-Kutta algorithm in Appendix B.3 is suitable if T_R is selected small enough. In rare instances it may be necessary to use an adaptive step size integrator such as Runge-Kutta-Fehlburg. In all the examples in this book, the fixed step size version was used.

A driver program that realizes Fig. 7.2-3 is given in Fig. 7.2-4. It is written in a modular fashion to apply to a wide variety of situations, and calls a Runge-Kutta integration routine such as the one in Appendix B.3. The Runge-Kutta integrator in turn calls subroutine $F(\text{TIME}, X, XP)$ containing the continuous-time dynamics.

The digital controller is contained in subroutine $\text{DIG}(T, X)$. Figure 7.2-3 assumes a ZOH; thus, the control input $u(t)$ is updated to u_k at each time kT and then held constant until time $(k + 1)T$. The driver program in Fig. 7.2-4 performs this.

It is important to realize that this simulation technique provides $x(t)$ as a continuous function of time, even at values *between* the sampling instants (in fact, it provides $x(t)$ at multiples of T_R). This is essential in verifying acceptable *intersample behavior* of the closed-loop system prior to implementing the digital controller on the actual plant. Even though the closed-loop behavior is acceptable at the sample points, with improper digital control system design there can be serious problems between the samples. The basic problem is that a badly designed controller can destroy observability, so that poor intersample behavior is not apparent at the sample points [Lewis, 1992]. This simulation scheme allows the intersample behavior to be checked prior to actual implementation.

We shall soon present several examples that demonstrate the simulation of digital controllers. First, it is necessary to discuss the design of digital controllers.

7.3 DISCRETIZATION OF CONTINUOUS CONTROLLERS

A digital controls design approach that could directly use all of the continuous-time techniques of the previous chapters would be extremely appealing. Therefore, in this section we discuss the design of digital controllers by the

```

C DRIVER PROGRAM TO COMPUTE AND SIMULATE DIGITAL CONTROL SCHEME
C REQUIRES SUBROUTINES:
C   DIG(T,X) FOR DIGITAL CONTROL UPDATE AT SAMPLING INSTANTS
C   RUNKUT(TIME,TR,X,NSTATES) TO INTEGRATE CONTINUOUS DYNAMICS
C   F(TIME,X,XP) TO PROVIDE CONTINUOUS PLANT DYNAMICS

      PROGRAM DIGICON
      REAL X(1)
      COMMON/CONTROL/U(1)
      COMMON/OUTPUT/Y(1)

C SET RUN TIME, SAMPLING PERIOD, RUNGE KUTTA STEP SIZE
      DATA TRUN,T,TR/5.,0.5,0.01/
C SET INITIAL PLANT STATE
      DATA X(1)/0./

      TIME= 0.
      N= NINT(TRUN/T)
      NT= NINT(T/TR)

* DIGITAL CONTROL SIMULATION RUN

      DO 10 K= 0,N-1

C UPDATE DIGITAL CONTROL INPUT
      CALL DIG(T,X)

C INTEGRATE CONTINUOUS DYNAMICS BETWEEN SAMPLES
      DO 10 I= 1,NT

C WRITE TO FILE FOR PLOT
      WRITE(7,*) TIME,X(1),U
      10  CALL RUNKUT(TIME,TR,X,1)
          WRITE(7,*) TIME,X(1),U

      STOP
      END

```

Fig. 7.2-4 Digital control simulation driver program.

redesign of existing continuous controllers. In this approach, the continuous controller is first designed using any desired technique. Then the controller is discretized using, for instance, the bilinear transform, to obtain the digital control law, which is finally programmed on the microprocessor.

An alternative approach for digital controls design is given in [Lewis, 1992]. In that approach, it is not necessary to design a continuous-time controller first, but a discrete-time controller is designed *directly* using a sampled version of the aircraft dynamics.

We now show how to discretize a continuous controller to obtain a digital controller. The idea is illustrated by designing a digital PID controller in Example 7.3-1 and a digital pitch rate control system in Example 7.3-2.

Suppose that a continuous-time controller $K^c(s)$ has been designed for the plant $G(s)$ by some means, such as root locus or LQ design. We shall discuss two approximate schemes for converting $K^c(s)$ into a discrete-time controller $K(z)$ that can be implemented on a microprocessor. First we discuss the bilinear transformation (BLT) and then the matched pole-zero (MPZ) technique.

The sample period is T seconds, so that the *sampling frequency* is

$$f_s = \frac{1}{T}, \quad \omega_s = \frac{2\pi}{T}. \quad (7.3-1)$$

Bilinear Transformation

A popular way to convert a continuous transfer function to a discrete one is the *bilinear transformation (BLT)* or *Tustin's approximation*. On sampling [Franklin and Powell, 1980] the continuous poles are mapped to discrete poles according to $z = e^{sT}$. As may be seen by series expansion

$$z = e^{sT} \approx \frac{1 + sT/2}{1 - sT/2}. \quad (7.3-2)$$

Therefore, to obtain an approximate sampling technique for continuous transfer functions, we may propose inverting this transformation and defining

$$s' = \frac{2}{T} \frac{z - 1}{z + 1}. \quad (7.3-3)$$

An approximate discrete equivalent of the continuous transfer function is then given by

$$K(z) = K^c(s'). \quad (7.3-4)$$

We call (7.3-3) the *bilinear transformation* or BLT.

The BLT corresponds to approximating integration using the trapezoid rule, since if

$$\frac{Y(z)}{U(z)} = \frac{2}{T} \frac{z - 1}{z + 1} = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

then (recall that z^{-1} is the unit delay in the time domain so that $z^{-1}u_k = u_{k-1}$)

$$u_k = u_{k-1} + \frac{T}{2}(y_k + y_{k-1}). \quad (7.3-5)$$

If the continuous transfer function is

$$K^c(s) = \frac{\prod_{i=1}^m(s + t_i)}{\prod_{i=1}^n(s + s_i)} \quad (7.3-6)$$

with the relative degree $r = n - m > 0$, then the BLT yields the approximate discrete equivalent transfer function given by

$$\begin{aligned} K(z) &= \frac{\prod_{i=1}^m \left[\frac{2(z-1)}{T(z+1)} + t_i \right]}{\prod_{i=1}^n \left[\frac{2(z-1)}{T(z+1)} + s_i \right]} \\ K(z) &= \left[\frac{T}{2}(z+1) \right]^r \frac{\prod_{i=1}^m [(z-1) + (z+1)t_i T/2]}{\prod_{i=1}^n [(z-1) + (z+1)s_i T/2]} \\ K(z) &= \left[\frac{T}{2}(z+1) \right]^r \frac{\prod_{i=1}^m [(1+t_i T/2)z - (1-t_i T/2)]}{\prod_{i=1}^n [(1+s_i T/2)z - (1-s_i T/2)]}. \end{aligned} \quad (7.3-7)$$

It can be seen that the poles and finite zeros map to the z -plane according to

$$z = \frac{1 + sT/2}{1 - sT/2}; \quad (7.3-8)$$

however, the r zeros at infinity in the s -plane map into zeros at $z = -1$. This is sensible, since $z = -1$ corresponds to the *Nyquist frequency* ω_N , where $z = e^{j\omega_N T} = -1$, so that $\omega_N T = \pi$ or

$$\omega_N = \frac{\pi}{T} = \frac{\omega_s}{2}. \quad (7.3-9)$$

This is the highest frequency before folding of $|K(e^{j\omega T})|$ occurs (see Fig. 7.4-1). Since the BLT maps the left-half of the s -plane into the unit circle, it maps stable continuous systems $K^c(s)$ into stable discrete $K(z)$.

According to (7.3-7), the BLT gives discretized transfer functions that have a relative degree of zero; that is, the degrees of the numerator and denominator are the same. If

$$K(z) = \frac{b_0 z^n + b_1 z^{n-1} + \cdots + b_n}{z^n + a_1 z^{n-1} + \cdots + a_n} \quad (7.3-10)$$

and $Y(z) = K(z)U(z)$, then the difference equation relating y_k and u_k is

$$y_k = -a_1 y_{k-1} - \cdots - a_n y_{k-n} + b_0 u_k + b_1 u_{k-1} + \cdots + b_n u_{k-n} \quad (7.3-11)$$

and the current output y_k depends on the current input u_k . This is usually an undesirable state of affairs, since it takes some computation time for the microprocessor to compute y_k . Techniques for including the computation time will be discussed later.

If the continuous-time controller is given in the state-space form

$$\begin{aligned} \dot{x} &= A^c x + B^c u \\ y &= Cx + Du, \end{aligned} \quad (7.3-12)$$

one may use the Laplace transform and (7.3-3) to show that the discretized system using the BLT is given by [Hanselmann, 1987]

$$\begin{aligned} x_{k+1} &= Ax_k + B_1 u_{k+1} + B_0 u_k \\ y_k &= Cx_k + Du_k \end{aligned} \quad (7.3-13)$$

with

$$\begin{aligned} A &= \left[I - A^c \frac{T}{2} \right]^{-1} \left[I + A^c \frac{T}{2} \right] \\ B_1 = B_0 &= \left[I - A^c \frac{T}{2} \right]^{-1} \frac{T}{2} B^c. \end{aligned} \quad (7.3-14)$$

Note that the discretized system is not a traditional state-space system since x_{k+1} depends on u_{k+1} . Aside from computation time delays, this is not a problem in our applications, since all we require of (7.3-13) is to implement it on a microprocessor. Since (7.3-13) is only a set of difference equations, this is easily accomplished. We illustrate how to discretize a continuous-time controller using the BLT in Examples 7.3-1 and 7.3-2, where we design a digital PID controller and a digital pitch rate controller.

Matched Pole-Zero

The second popular approximation technique for converting a continuous transfer function to a discrete one is the *matched pole-zero (MPZ)* method. Here, both the poles and finite zeros are mapped into the z -plane using the transformation e^{sT} , as follows:

1. If $K^c(s)$ has a pole (or finite zero) at $s = s_i$, then $K(z)$ will have a pole (or finite zero) at

$$z_i = e^{s_i T}. \quad (7.3-15)$$

2. If the relative degree of $K^c(s)$ is r , so that it has r zeros at infinity, r zeros of $K(z)$ are taken at $z = -1$ by multiplying by the factor $(1 + z)^r$.
3. The gain of $K(z)$ is selected so that the dc gains of $K^c(s)$ and $K(z)$ are the same, that is, so that

$$K(1) = K^c(0). \quad (7.3-16)$$

An alternative to step 2 is to map only $r - 1$ of the infinite s -plane zeros into $z = -1$. This leaves the relative degree of $K(z)$ equal to 1, which allows one sample period for control computation time. We shall call this the *modified MPZ* method.

Thus if

$$K^c(s) = \frac{\prod_{i=1}^m (s + t_i)}{\prod_{i=1}^n (s + s_i)} \quad (7.3-17)$$

and the relative degree is $r = n - m$, the MPZ discretized transfer function is

$$K(z) = k(z + 1)^{r-1} \frac{\prod_{i=1}^m (z - e^{-t_i T})}{\prod_{i=1}^n (z - e^{-s_i T})}, \quad (7.3-18)$$

where the gain k is chosen to ensure (7.3-16). Note that if $K^c(s)$ is stable, so is the $K(z)$ obtained by the MPZ, since $z = e^{sT}$ maps the left-half s -plane into the unit circle in the z -plane. Although the MPZ requires simpler algebra than the BLT, the latter is more popular in industry.

Digital Design Examples

Now let us show some examples of digital controller design using the BLT and MPZ to discretize continuous controllers.

Example 7.3-1: Discrete PID Controller. Since the continuous PID controller is so useful in aircraft controls design, let us demonstrate how to discretize it to obtain a digital PID controller. A standard continuous-time PID controller has the transfer function [Åström and Wittenmark, 1984]

$$K^c(s) = k \left[1 + \frac{1}{T_I s} + \frac{T_D s}{1 + T_D s/N} \right], \quad (1)$$

where k is the proportional gain, T_I is the integration time constant or “reset” time, and T_D is the derivative time constant. Rather than use pure differentiation, a “filtered derivative” is used that has a pole far left in the s -plane at $s = -N/T_D$. A typical value for N is 3 to 10; it is usually fixed by the manufacturer of the controller.

Let us consider a few methods of discretizing (1) with sample period T seconds.

a. **BLT.** Using the BLT, the discretized version of (1) is found to be

$$K(z) = k \left[1 + \frac{1}{T_I \frac{2(z-1)}{T(z+1)}} + \frac{T_D \frac{2(z-1)}{T(z+1)}}{1 + \frac{T_D}{N} \frac{2(z-1)}{T(z+1)}} \right] \quad (2)$$

or, on simplifying,

$$K(z) = k \left[1 + \frac{T}{T_{Id}} \frac{z+1}{z-1} + \frac{T_{Dd}}{T} \frac{z-1}{z-\nu} \right] \quad (3)$$

with the discrete integral and derivative time constants

$$T_{Id} = 2T_I \quad (4)$$

$$T_{Dd} = \frac{NT}{1 + NT/2T_D} \quad (5)$$

and the derivative-filtering pole at

$$\nu = \frac{1 - NT/2T_D}{1 + NT/2T_D}. \quad (6)$$

b. **MPZ.** Using the MPZ approach to discretize the PID controller yields

$$K(z) = k \left[1 + \frac{k_1(z+1)}{T_I(z-1)} + \frac{k_2 N(z-1)}{z - e^{-NT/T_D}} \right], \quad (7)$$

where k_1 and k_2 must be selected to match the dc gains. At dc, the D terms in (1) and (7) are both zero, so we may select $k_2 = 1$. The dc values of the I terms in (1) and (7) are unbounded. Therefore, to select k_1 let us match the low-frequency gains. At low frequencies, $e^{j\omega T} \approx 1 + j\omega T$. Therefore, for small ω , the I terms of (1) and (7) become

$$K^c(j\omega) = \frac{1}{j\omega T_I}$$

$$K(e^{j\omega T}) \approx \frac{2k_1}{T_I(j\omega T)},$$

and to match them, we require that $k_1 = T/2$.

Thus, using the MPZ the discretized PID controller again has the form (3), but now with

$$T_{Id} = 2T_I \quad (8)$$

$$T_{Dd} = NT \quad (9)$$

$$\nu = e^{-NT/T_D}. \quad (10)$$

c. *Modified MPZ.* If we use the modified MPZ method, then in the I term in (7) the factor $(z + 1)$ does not appear. Then the normalizing gain k_1 is computed to be T . In this case, the discretized PID controller takes on the form

$$K(z) = k \left[1 + \frac{T}{T_{Id}} \frac{1}{z - 1} + \frac{T_{Dd}}{T} \frac{z - 1}{z - \nu} \right] \quad (11)$$

with

$$T_{Id} = T_I \quad (12)$$

$$T_{Dd} = NT \quad (13)$$

$$\nu = e^{-NT/T_D}. \quad (14)$$

Now, there is a control delay of 1 sample period (T seconds) in the integral term, which could be advantageous if there is a computation delay.

d. *Difference Equation Implementation.* Let us illustrate how to implement the modified MPZ PID controller (11) using difference equations, which are easily placed into a software computer program. It is best from the point of view of numerical accuracy in the face of computer round-off error to implement digital controllers as several first- or second-order systems in parallel. Such a parallel implementation may be achieved as follows.

First, write $K(z)$ in terms of z^{-1} , which is the unit delay in the time domain (i.e. a delay of T s, so that, for instance, $z^{-1}u_k = u_{k-1}$), as

$$K(z^{-1}) = k \left[1 + \frac{T}{T_{Id}} \frac{z^{-1}}{1 - z^{-1}} + \frac{T_{Dd}}{T} \frac{1 - z^{-1}}{1 - \nu z^{-1}} \right]. \quad (15)$$

(Note: there is some abuse in notation in denoting (15) as $K(z^{-1})$; this, we shall accept.)

Now, suppose that the control input u_k is related to the tracking error as

$$u_k = K(z^{-1})e_k. \quad (16)$$

Then, u_k may be computed from past and present values of e_k using auxiliary variables as follows:

$$v_k^I = v_{k-1}^I + \frac{T}{T_{Id}} e_{k-1} \quad (17)$$

$$v_k^D = \nu v_{k-1}^D + \frac{T_{Dd}}{T} (e_k - e_{k-1}) \quad (18)$$

$$u_k = k(e_k + v_k^I + v_k^D). \quad (19)$$

The variables v_k^I and v_k^D represent the integral and derivative portion of the PID controller, respectively. For more discussion, see Åström and Wittenmark [1984]. ■

Example 7.3-2: Digital Pitch Rate Controller via BLT. In Example 5.5-3 we designed a pitch-rate control system using LQ output-feedback techniques. Here we demonstrate how to convert that continuous control system into a digital control system. The BLT is popular in industry; therefore, we shall use it here.

The continuous controller is illustrated in Fig. 7.3-1, where

$$K_1^c(s) = \frac{k_I}{s} \quad (1)$$

$$K_2^c(s) = \frac{10k_\alpha}{s + 10} \quad (2)$$

$$K_3^c(s) = k_q. \quad (3)$$

The most suitable feedback gains in Example 5.5-3 were found using derivative weighting design to be

$$k_I = 1.361, \quad k_\alpha = -0.0807, \quad k_q = -0.475. \quad (4)$$

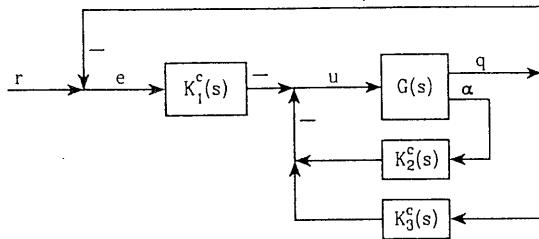


Fig. 7.3-1 Continuous pitch-rate controller.

A digital control scheme with the same structure is shown in Fig. 7.3-2. We have added samplers with period T to produce the samples of pitch rate q and angle of attack α , as well as a hold device to convert the control samples u_k computed by the digital controller back to a continuous-time control input $u(t)$ for the plant. Note that the reference input $r(t)$ must also be sampled.

Since the integrator and alpha smoothing filter are part of the digital controller, the continuous dynamics $G(s)$ in Fig. 7.3-2 are given by $\dot{x} = Ax + Bu$, $y = Cx$ with

$$A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -20.2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 20.2 \end{bmatrix}$$

$$C = \begin{bmatrix} 57.2958 & 0 & 0 \\ 0 & 57.2958 & 0 \end{bmatrix}, \quad (5)$$

where

$$x = \begin{bmatrix} \alpha \\ q \\ \delta_e \end{bmatrix}, \quad y = \begin{bmatrix} \alpha \\ q \end{bmatrix}. \quad (6)$$

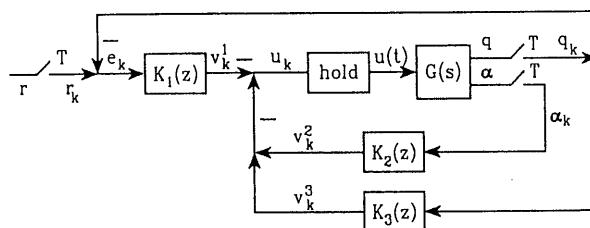


Fig. 7.3-2 Digital pitch-rate controller.

Using the BLT, the discrete equivalents to (1)–(3) are found to be

$$K_1(z) = k_1 \frac{z + 1}{z - 1} \quad \text{with } k_1 = \frac{k_1 T}{2} \quad (7)$$

$$K_2(z) = k_2 \frac{z + 1}{z - \pi} \quad \text{with } k_2 = \frac{10k_\alpha T}{10T + 2}, \quad \pi = \frac{1 - 10T/2}{1 + 10T/2} \quad (8)$$

$$K_3(z) = k_q. \quad (9)$$

Defining the intermediate signals v_k^1, v_k^2, v_k^3 shown in Fig. 7.3-2 and denoting the unit delay in the time domain by z^{-1} , we may express (7)–(9) in terms of difference equations as follows:

$$e_k = r_k - q_k, \quad (10)$$

$$v_k^1 = k_1 \frac{1 + z^{-1}}{1 - z^{-1}} e_k$$

or

$$v_k^1 = v_{k-1}^1 + k_1(e_k + e_{k-1}), \quad (11)$$

$$v_k^2 = k_2 \frac{1 + z^{-1}}{1 - \pi z^{-1}} \alpha_k$$

or

$$v_k^2 = \pi v_{k-1}^2 + k_2(\alpha_k + \alpha_{k-1}), \quad (12)$$

$$v_k^3 = k_q q_k. \quad (13)$$

The control samples u_k are thus given by

$$u_k = -(v_k^1 + v_k^2 + v_k^3). \quad (14)$$

Note the low-pass filtering effects manifested by the averaging of e_k and α_k that occurs in these equations. This will tend to average out any measurement noise.

These difference equations describe the digital controller, and are easily implemented on a microprocessor. First, however, the controller should be simulated. The FORTRAN subroutine in Fig. 7.3-3a may be used with the driver program in Fig. 7.2-4 to simulate the digital control law. The subroutine $F(t, x, \dot{x})$ required by the Runge-Kutta integrator for the continuous plant dynamics (5) is given in Fig. 7.3-3b.

The step response using this digital controller was plotted for several sampling periods T in Fig. 7.3-4. A zero-order hold was used. Note that the step response improves as T becomes small. Indeed, the response for

C DIGITAL PITCH RATE CONTROLLER

```

SUBROUTINE DIG(IK,T,X)
REAL X(*), K(2), KI, KA, KQ
COMMON/CONTROL/U
COMMON/OUTPUT/AL,Q,UPILOT
DATA REF, KI,KA,KQ/1., 1.361,-0.0807,-0.475/

K(1)= KI*T/2
K(2)= 10*KA*T/(10*T + 2)
P= (1 - 10*T/2) / (1 + 10*T/2)

E= REF - Q
V1= V1 + K(1)*(E + EKM1)
V2= P*V2 + K(2)*(AL + ALKM1)
V3= KQ*Q
U= -(V1 + V2 + V3)
UPILOT= U

EKM1= E
ALKM1= AL

RETURN
END

```

(a)

C CONTINUOUS SHORT PERIOD DYNAMICS

```

SUBROUTINE F(TIME,X,XP)
REAL X(*), XP(1)
COMMON/CONTROL/U
COMMON/OUTPUT/AL,Q

XP(1)= -1.01887*X(1) + 0.90506*X(2) - 0.00215*X(3)
XP(2)= 0.82225*X(1) - 1.07741*X(2) - 0.17555*X(3)
XP(3)= -20.2 *X(3) + 20.2*u

AL = 57.2958*X(1)
Q = 57.2958*X(2)

RETURN
END

```

(b)

Fig. 7.3-3 Digital simulation software: (a) FORTRAN subroutine to simulate digital pitch-rate controller; (b) subroutine $F(t, x, \dot{x})$ to simulate continuous plant dynamics.

$T = 0.025$ s is indistinguishable from the response using a continuous controller in Example 5.5-3c.

The motivation for selecting $T = 0.025$ s was as follows. The settling time of the continuous controller step response in Example 5.5-3c was $t_s = 1$ s. The settling time is about four times the slowest time constant, which is thus 0.25 s. The sampling period should be selected about one-tenth of this for good performance.

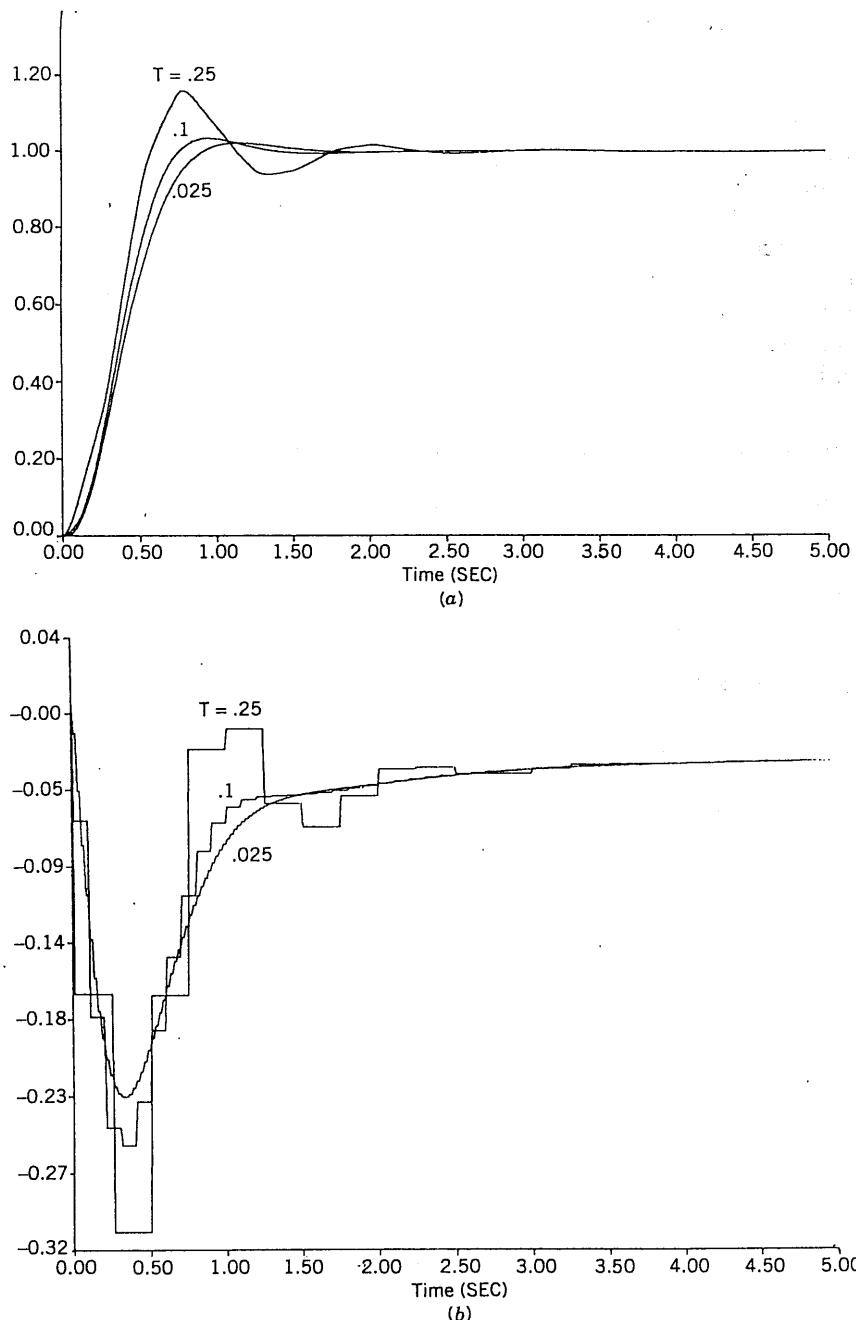


Fig. 7.3-4 Effect of sampling period: (a) step response $q(t)$; (b) control input $u(t)$.

7.4 MODIFIED CONTINUOUS DESIGN

In Section 7.3 we showed how to convert a continuous-time controller to a digital controller using the BLT and MPZ. However, that technique is only an approximate one that gives worse results as the sample period T increases. In this section we show how to *modify the design of the continuous controller* so that it yields a more suitable digital controller. This allows the use of larger sample periods. To do this we shall take into account some properties of the zero-order-hold and sampling processes. Using modified continuous design, we are able to design in Example 7.4-1 a digital pitch-rate control system that works extremely well even for relatively large sample periods.

Sampling, Hold Devices, and Computation Delays

We shall examine some of the properties of the discretization and implementation processes to see how the continuous controller may be designed in a fashion that will yield an improved digital controller. Specifically, in the design of the continuous controller it is desirable to include the effects of sampling, hold devices, and computation delays.

Sampling and Aliasing. We would like to gain some additional insight on the sampling process [Oppenheim and Schafer, 1975; Franklin and Powell, 1980; Åström and Wittenmark, 1984]. To do so, define the Nyquist frequency $\omega_N = \omega_s/2 = \pi/T$, and the sampling frequency $\omega_s = 2\pi/T$ and picture the output $y^*(t)$ of the sampler with input $y(t)$ as the string of impulses

$$y^*(t) = \sum_{k=-\infty}^{\infty} y(t)\delta(t - kT) \quad (7.4-1)$$

where $\delta(t)$ is the unit impulse. Since the impulse train is periodic, it has a Fourier series that may be computed to be

$$\sum_{k=-\infty}^{\infty} \delta(t - kT) = \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{jn\omega_s t}. \quad (7.4-2)$$

Using this in (7.4-1) and taking the Laplace transform yields

$$\begin{aligned} Y^*(s) &= \frac{1}{T} \int_{-\infty}^{\infty} y(t) \left[\sum_{n=-\infty}^{\infty} e^{jn\omega_s t} \right] e^{-st} dt \\ Y^*(s) &= \frac{1}{T} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} y(t) e^{-(s-jn\omega_s)t} dt \\ Y^*(s) &= \frac{1}{T} \sum_{n=-\infty}^{\infty} Y(s - jn\omega_s), \end{aligned} \quad (7.4-3)$$

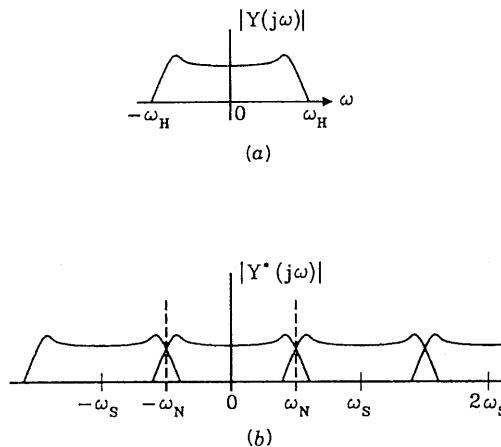


Fig. 7.4-1 Sampling in the frequency domain: (a) spectrum of $y(t)$; (b) spectrum of sampled signal $y^*(t)$.

where $Y(s)$ is the Laplace transform of $y(t)$ and $Y^*(s)$ is the Laplace transform of the sampled signal $y^*(t)$. Due to the factor $1/T$ appearing in (7.4-3), the sampler is said to have a *gain of $1/T$* .

Sketches of a typical $Y(j\omega)$ and $Y^*(j\omega)$ are shown in Fig. 7.4-1, where ω_H is the highest frequency contained in $y(t)$. Notice that the digital frequency response is symmetric with respect to ω_N and periodic with respect to ω_s . At frequencies less than ω_N , the spectrum of $Y^*(j\omega)$ has two parts: one part comes from $Y(j\omega)$ and is the portion that should appear. However, there is an additional portion from $Y(j(\omega - \omega_s))$; the “tail” of $Y(j(\omega - \omega_s))$, which contains high-frequency information about $y(t)$, is “folded” back or *aliased* into the lower frequencies of $Y^*(j\omega)$. Thus the high-frequency content of $y(t)$ appears at low frequencies and can lead to problems in reconstructing $y(t)$ from its samples.

If $\omega_H < \omega_N$, the tail of $Y^*(j(\omega - \omega_s))$ does not appear to the right of $\omega = \omega_N$ and $y(t)$ can be uniquely reconstructed from its samples by low-pass filtering. This condition is equivalent to

$$\omega_s > 2\omega_H, \quad (7.4-4)$$

which is the sampling theorem of Shannon that guarantees aliasing does not occur.

It is interesting to see what the sampling theorem means in the time domain. Examine Fig. 7.4-2, where we show two continuous signals that have the same samples. If the original signal was the higher-frequency signal, the D/A reconstruction process will produce the lower-frequency signal from the samples of the higher-frequency signal. Thus aliasing can result in *high-*

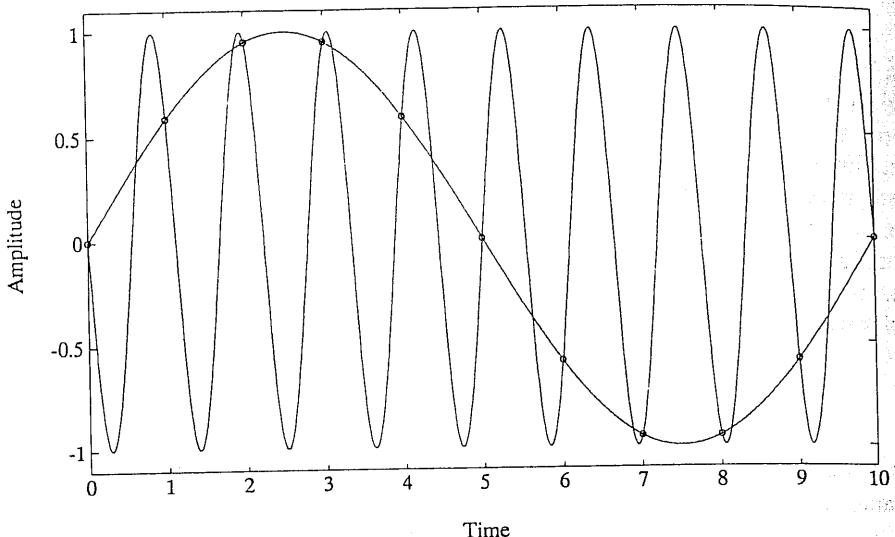


Fig. 7.4-2 Example of aliasing in the time domain.

frequency signals being misinterpreted as low-frequency signals. If the sampling frequency ω_s is greater than twice the highest frequency ω_H appearing in the continuous signal, the problem depicted in the figure does not occur and the signal can be accurately reconstructed from its samples.

Selecting the Sampling Period. For controls design, the sampling frequency ω_s must generally be significantly greater than twice the highest frequency of any signal appearing in the system. That is, in controls applications the sampling theorem does not usually provide much insight in selecting ω_s . Some guides for selecting the sampling period T are now discussed.

If the continuous-time system has a single dominant complex pole pair with natural frequency of ω , the rise time is given approximately by

$$t_r = \frac{1.8}{\omega}. \quad (7.4-5)$$

It is reasonable to have at least two to four samples per rise time so that the error induced by ZOH reconstruction is not too great during the fastest variations of the continuous-time signal [Åström and Wittenmark, 1984]. Then we have $t_r = 1.8/\omega \geq 4T$, or approximately

$$T \leq \frac{1}{2\omega}. \quad (7.4-6)$$

However, if high-frequency components are present up to a frequency of ω_H radians and it is desired to retain them in the sampled system, a rule of thumb is to select

$$T \leq \frac{1}{4\omega_H}. \quad (7.4-7)$$

These formulae should be used with care, and to select a suitable T it may be necessary to perform digital control designs for several values of T , for each case carrying out a computer simulation of the behavior of the plant under the influence of the proposed controller. Note particularly that using continuous redesign of digital controllers with the BLT or MPZ, even smaller sample periods may be required since the controller discretization technique is only an approximate one.

Zero-Order Hold. The D/A hold device in Fig. 7.2-1 is required to reconstruct the plant control input $u(t)$ from the samples u_k provided by the digital control scheme. The zero-order hold (ZOH) is usually used. There, we take

$$u(t) = u(kT) = u_k, \quad kT \leq t < (k+1)T, \quad (7.4-8)$$

with u_k the k th sample of $u(t)$. The ZOH yields the sort of behavior in Fig. 7.2-2 and has the impulse response shown in Fig. 7.4-3. This impulse response may be written as

$$h(t) = u_{-1}(t) - u_{-1}(t-T),$$

with $u_{-1}(t)$ the unit step. Thus the transfer function of the ZOH is

$$G_0(s) = \frac{1 - e^{-sT}}{s}. \quad (7.4-9)$$

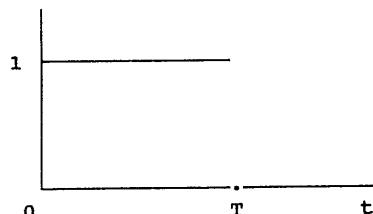


Fig. 7.4-3 ZOH impulse response.

To determine the Bode magnitude and phase of $G_0(s)$, write

$$\begin{aligned} G_0(j\omega) &= \frac{1 - e^{-j\omega T}}{j\omega} = e^{-j\omega T/2} \frac{e^{j\omega T/2} - e^{-j\omega T/2}}{j\omega} \\ G_0(j\omega) &= Te^{-j\omega T/2} \frac{\sin(\omega T/2)}{\omega T/2} = Te^{-j\omega T/2} \operatorname{sinc} \frac{\omega}{\omega_s}, \end{aligned} \quad (7.4-10)$$

where $\operatorname{sinc} x \equiv (\sin \pi x)/\pi x$. The magnitude and phase of the ZOH are shown in Fig. 7.4-4. Note that the ZOH is a low-pass filter of magnitude $T|\operatorname{sinc}(\omega/\omega_s)|$ with a phase of

$$\angle \text{ZOH} = -\frac{\omega T}{2} + \theta = -\frac{\pi \omega}{\omega_s} + \theta, \quad \theta = \begin{cases} 0, & \sin \frac{\omega T}{2} > 0 \\ \pi, & \sin \frac{\omega T}{2} < 0. \end{cases} \quad (7.4-11)$$

According to (7.4-10), for frequencies ω much smaller than ω_s , the ZOH may be approximated by

$$G_0(s) \approx Te^{-sT/2}, \quad (7.4-12)$$

that is, by a pure delay of half the sampling period and a scale factor of T .

As we saw in the digital pitch-rate controller in Example 7.3-1, the performance of the digital controller deteriorates with increasing T , so that sample periods are required which may be too small. (We note that smaller values of T require faster computation to compute u_k ; thus a faster, and more expensive, microprocessor may be required for small T .) This deterioration is partly due to the delay introduced by the hold device. We shall soon see how to take this delay into account *while designing the continuous controller*, so that discretization yields a digital controller that gives suitable performance for larger values of T .

Computation Delay. If the microprocessor is fast so that the time Δ required to compute the digital control law is negligible, Δ will have little effect when a digital controller is implemented. However, if Δ is appreciable, it can have a deleterious effect on the closed-loop response. Then it may be necessary to account for it.

If $\Delta \leq T$, the computation delay may be accounted for by ensuring that u_k depends only on *previous* values of the outputs. This may often be achieved by using the modified MPZ approach for digital controller design. However, the BLT is more popular and it always yields a u_k that depends on *current* values of the outputs [see (7.3-11)]. Moreover, the discrete PID controller

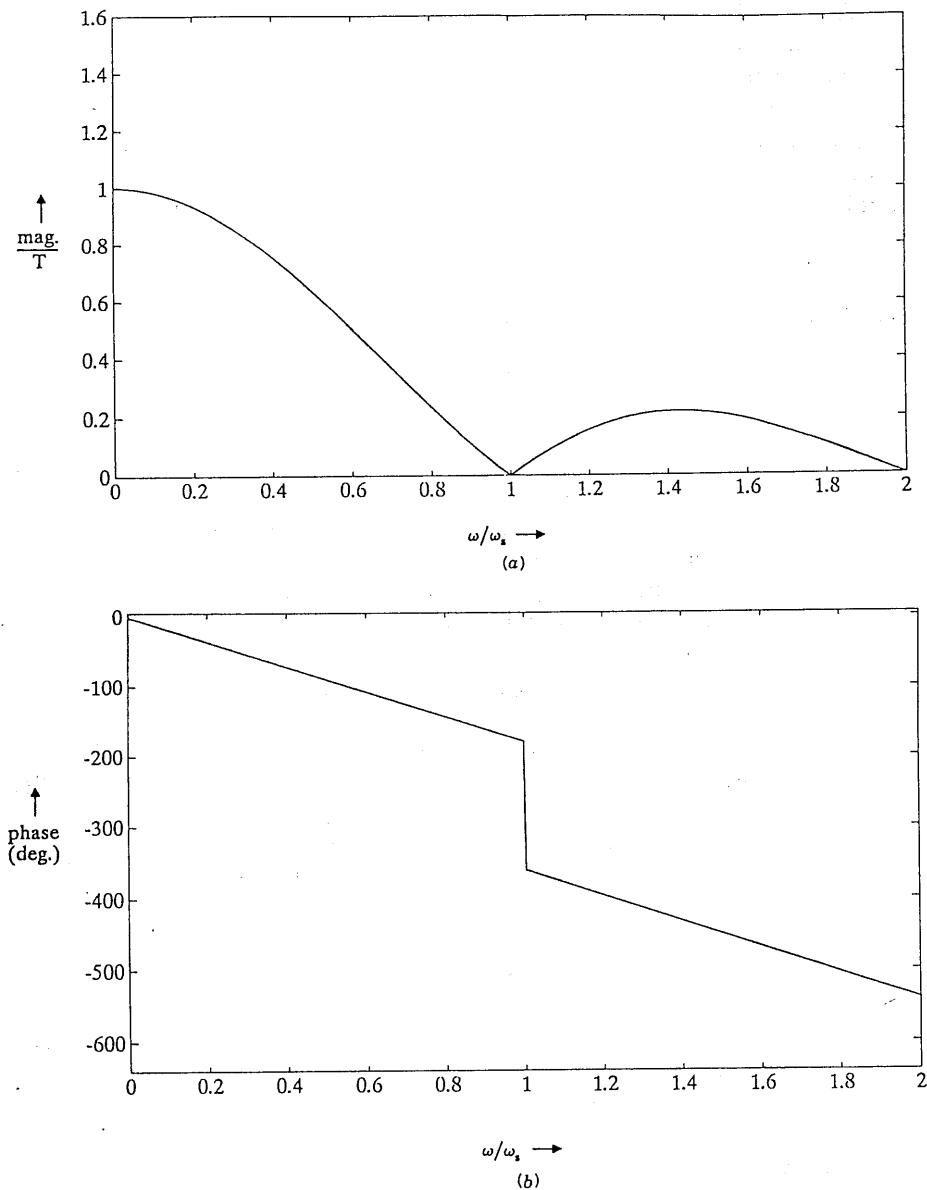


Fig. 7.4-4 ZOH Bode plots: (a) magnitude; (b) phase.

(see Example 7.3-1) always has a dependence on the current outputs through the derivative term, even if the modified MPZ is used.

If the computation delay is not negligible but is only a fraction of T , it seems inefficient to allow it to cause a delay of a full T seconds in applying the control to the plant. If there is noise present in the system, then using outputs delayed by an entire sample period to compute u_k can, for large sample periods, lead to significant deterioration over using more recent outputs to compute u_k .

Modified Continuous Design Procedures

We shall now show how to account for the hold delay and computation delay while designing the continuous controller $K^c(s)$ for discretization. Then, when the BLT or MPZ is used to discretize $K^c(s)$, a digital controller $K(z)$ with improved performance will be obtained. We call this approach *modified continuous controller design for discretization*.

A disadvantage of modified continuous design techniques is that the sample period T must be selected prior to the continuous controller design. However, good software makes it easy to redesign the continuous controller with a different value of T . The advantage of the approach is that the effects of the sampling and hold operations, computation delay, and aliasing are apparent while the continuous design is being performed. Thus they may be to some extent compensated for.

Modified continuous design can often allow significantly larger sample periods than direct application of the BLT or MPZ to a continuous controller designed with no consideration that the next step will be conversion to a digital control law. This will be illustrated in Example 7.4-1, where we design a pitch-rate control system by modified continuous design.

Let us discuss aliasing, computation delays, and then the ZOH.

Aliasing. The plant $G(s)$ is generally a low-pass filter. We have seen in Fig. 7.4-1 that as long as the sampling frequency ω_s is selected at least twice as large as the plant cutoff frequency ω_H , the effects of aliasing will be small.

However, one type of signal appearing in the closed-loop system that may not be bandlimited is *measurement noise*. High-frequency measurement noise may be aliased down to lower frequencies that are within the plant bandwidth and thus have a detrimental effect on system performance. To avoid this, low-pass *anti-aliasing filters* of the form

$$H_a(s) = \frac{a}{s + a} \quad (7.4-13)$$

may be inserted after the measuring devices and before the samplers. The cutoff frequency a should be selected less than $\omega_N = \omega_s/2$, so that there is good attenuation beyond ω_N rad/s.

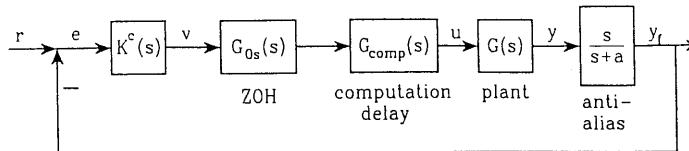


Fig. 7.4-5 Modified continuous plant with antialiasing filter and compensation to model hold device and computation delays.

If the cutoff frequency of the antialiasing filter is not much higher than the plant cutoff frequency, the filter will affect the closed-loop performance, and it should be appended to the plant *at the design stage* so that the continuous controller is designed taking it into account. See Fig. 7.4-5, which represents the actual plant $G(s)$ augmented by various filters, some still to be discussed, that should be taken into account in the design stage.

Computation Delay. The delay associated with a computation time of Δ has a transfer function of

$$G_{\text{comp}}(s) = e^{-s\Delta}, \quad (7.4-14)$$

which has a magnitude of 1 and a phase of $-\omega\Delta$ radians. To account for this delay, we may perform the continuous controller design not on the plant $G(s)$, but on $G(s)e^{-s\Delta}$. However, it is awkward to design a controller for a plant whose transfer function is not rational [Franklin et al., 1986]. It is more convenient to approximate the delay with a rational transfer function.

For this purpose, we may use Padé approximants to $e^{-s\Delta}$, which match the first few terms of the Taylor's series expansion [Su, 1971; Franklin et al., 1986]. In Table 7.4-1 are given several Padé approximants to $e^{-s\Delta}$. These approximants match the first $n + m + 1$ terms of the Taylor series expansion, where n is the denominator degree and m the numerator degree.

To perform a modified continuous design that takes into account the computation delay Δ , it is only necessary to incorporate a Padé approximant $G_{\text{comp}}(s)$ to $e^{-s\Delta}$ of suitable order into the plant as shown in Fig. 7.4-5. The continuous controller $K^c(s)$ designed for this modified plant is then discretized using the BLT or MPZ to produce a digital controller $K(z)$.

Notice that the Padé approximants in Table 7.4-1 having finite zeros are *non-minimum-phase*. This is a property of a pure time delay. The advantage of the modified continuous design approach is that the non-minimum-phase nature of the delayed plant manifests itself at the continuous controller design stage, so that the digital controller that results after using the BLT compensates for this problem automatically.

TABLE 7.4-1. Padé Approximants to $e^{-s\Delta}$ for Approximation of Computation Delay

$\frac{1}{1+s\Delta}$	$\frac{1-s\Delta/2}{1+s\Delta/2}$	$\frac{1}{1+s\Delta+(s\Delta)^2/2}$	$\frac{1-s\Delta/3}{1+2s\Delta/3+(s\Delta)^2/6}$	$\frac{1}{1+s\Delta+(s\Delta)^2/4}$	$\frac{1-s\Delta/4}{1+3s\Delta/4+(s\Delta)^2/4+(s\Delta)^3/24}$	$\frac{1}{1+s\Delta+(s\Delta)^2/2+(s\Delta)^3/6}$	$\frac{1-s\Delta/2+(s\Delta)^2/12}{1+s\Delta/2+(s\Delta)^2/12}$	$\frac{1-2s\Delta/5+(s\Delta)^2/20}{1+3s\Delta/5+3(s\Delta)^2/20+(s\Delta)^3/60}$
-----------------------	-----------------------------------	-------------------------------------	--	-------------------------------------	---	---	---	---

TABLE 7.4-2. Approximants to $(1 - e^{-sT}) / sT$ for Approximation of Hold Delay

$\frac{1}{1 + sT/2}$
$\frac{1 - sT/6}{1 + sT/3}$
$\frac{1 - sT/10 + (sT)^2/60}{1 + 2sT/5 + (sT)^2/20}$
$\frac{1 - sT/14 + 23(sT)^2/840 - (sT)^3/840}{1 + 3sT/7 + (sT)^2/14 + (sT)^3/120}$

Zero-Order Hold. Finally, let us discuss modified continuous design taking into account the ZOH. Since the sampler has a gain of $1/T$, the sampler plus ZOH has a transfer function of

$$G_{0s}(s) = \frac{1 - e^{-sT}}{sT}. \quad (7.4-15)$$

Some useful approximants to $G_{0s}(s)$ are given in Table 7.4-2. These have been computed using Padé approximants of e^{-sT} , so they are not strictly speaking Padé approximants, since they only match the first $n + m$ terms of the Taylor series. They are, however, sufficiently accurate for our purposes. Note that the approximants to $G_{0s}(s)$ have unstable zeros. Modified continuous controller design taking into account $G_{0s}(s)$ involves designing a controller for $G(s)G_{0s}(s)$ (see Fig. 7.4-5).

Implementation. It is important to realize that the antialiasing filter should be implemented using analog circuitry as part of the plant $G(s)$. It should immediately precede the sampler. $G_{\text{comp}}(s)$, on the other hand, is not implemented since it is a model of the computation delay. $G_{0s}(s)$ is implemented by the ZOH and the sampler. $K^c(s)$ is discretized using the BLT or MPZ and becomes the digital controller $K(z)$.

The next example illustrates modified continuous design for discretization.

Example 7.4-1: Digital Pitch Rate Controller via Modified Continuous Design. In Example 5.5-3 we designed a continuous-time pitch rate controller. In Example 7.3-2 we showed how to use the BLT to convert that controller into digital form. It was seen that the response was good for $T = 0.025$ s, slightly worse for $T = 0.1$ s, and unacceptable for $T = 0.25$ s.

In this example let us design a modified continuous controller which, on discretization, will yield a better digital controller using large sample periods than the one of Example 7.3-2. We will select the sampling period in this example to be $T = 0.25$ s.

a. Modified Continuous-Time Plant. To account for the effects of the hold delay we shall incorporate a model of the sampling and hold processes into the continuous-time dynamical model of the aircraft as shown in Fig. 7.4-5. Let us use a Padé approximant to (7.4-15). Specifically, examining Table 7.4-2, select

$$G_{0s}(s) = \frac{1 - sT/6}{1 + sT/3} = -\frac{1}{2} + \frac{9/2T}{s + 3/T}. \quad (1)$$

According to Fig. 7.4-5, the ZOH/sampler approximant should act as a filter on the plant control input $u(t)$. Thus a state-variable representation of $G_{0s}(s)$ is given by

$$\begin{aligned} \dot{x}_z &= -\frac{3}{T}x_z + \frac{9}{2T}v \\ u &= x_z - \frac{1}{2}v, \end{aligned} \quad (2)$$

where $v(t)$ is the new input shown in Fig. 7.4-5. With $T = 0.25$ s this becomes

$$\begin{aligned} \dot{x}_z &= -12x_z + 18v \\ u &= x_z - 0.5v. \end{aligned} \quad (3)$$

We should like to propose the same control structure used in Example 5.5-3. There, an angle-of-attack filter and an integrator in the feedforward channel were used. The ZOH/sampler dynamics (3) may be augmented into the system-plus-compensator state equations by defining the augmented state

$$x = [\alpha \quad q \quad \delta_e \quad \alpha_F \quad \epsilon \quad x_z]^T. \quad (4)$$

Then

$$\dot{x} = Ax + Bu + Er \quad (5)$$

$$y = Cx + Fr \quad (6)$$

$$z = Hx \quad (7)$$

with

$$A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 & 0 & 0 & 0 \\ 0.82225 & -1.07741 & -0.17555 & 0 & 0 & 0 \\ 0 & 0 & -20.2 & 0 & 0 & 20.2 \\ 10.0 & 0 & 0 & -10 & 0 & 0 \\ 0 & -57.2958 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -12 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 0 \\ -10.1 \\ 0 \\ 0 \\ 18 \end{bmatrix}, \quad E = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 57.2958 & 0 & 0 \\ 0 & 57.2958 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$H = [0 \quad 57.2958 \quad 0 \quad 0 \quad 0 \quad 0].$$

Then, according to Example 5.5-3 the control input $v(t)$ is given by

$$v = -Ky = -[k_\alpha \quad k_q \quad k_I]y = -k_\alpha\alpha_F - k_qq - k_I\epsilon. \quad (8)$$

We are now in a position to perform the controls design to select the control gains.

b. PI and Continuous Controls Design. To design the continuous-time controller, let us select the PI

$$J = \frac{1}{2} \int_0^{\infty} (q_5 t^2 e^2 + \dot{\delta}_e^2) dt \quad (9)$$

that weights elevator rate of change, since this is closely related to actuator energy. Since $e(t) = \dot{\epsilon}(t)$, this may be written

$$J = \frac{1}{2} \int_0^{\infty} (q_5 t^2 \dot{\epsilon}^2 + \dot{\delta}_e^2) dt, \quad (10)$$

with $\epsilon(t)$ and $\dot{\delta}_e(t)$ the deviations in the integrator output and elevator deflection. Thus this is the PI with derivative weighting discussed in Section 5.5.

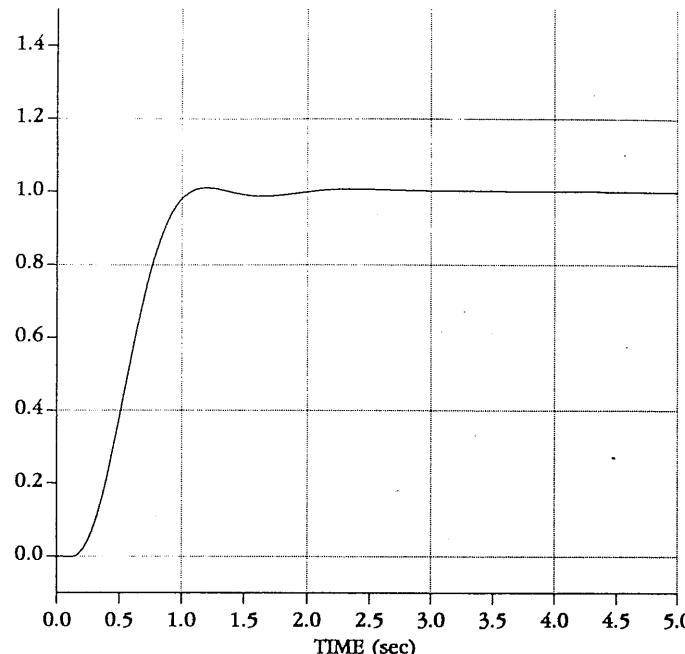


Fig. 7.4-6 Step response $q(t)$ using modified continuous-time pitch rate controller.

Using $q_5 = 5$ and the software described in Appendix B, we computed the optimal gain matrix

$$K = \begin{bmatrix} -0.04238 & -0.4098 & 0.8426 \end{bmatrix} \quad (11)$$

which gave the closed-loop poles

$$\begin{aligned} s = & -2.40 \pm j4.71 \\ & -1.08, -2.76 \\ & -9.86, -25.80. \end{aligned} \quad (12)$$

The closed-loop step response of the continuous-time controller is shown in Fig. 7.4-6. Note that it is comparable to the responses shown in Example 5.5-3.

Let us note that the transfer function from $v(t)$ to $q(t)$ contains the approximate ZOH/sampler dynamics described by (1), (3). These include a pole at $s = -12$ which has no significant effect. However, they also include a non-minimum-phase zero at $s = 24$. This zero significantly changes the root locus, and the control gains (11) selected automatically by the LQ approach take this non-minimum-phase zero into account. Indeed, note the delay of approximately $T = 0.25$ s in Fig. 7.4-6.

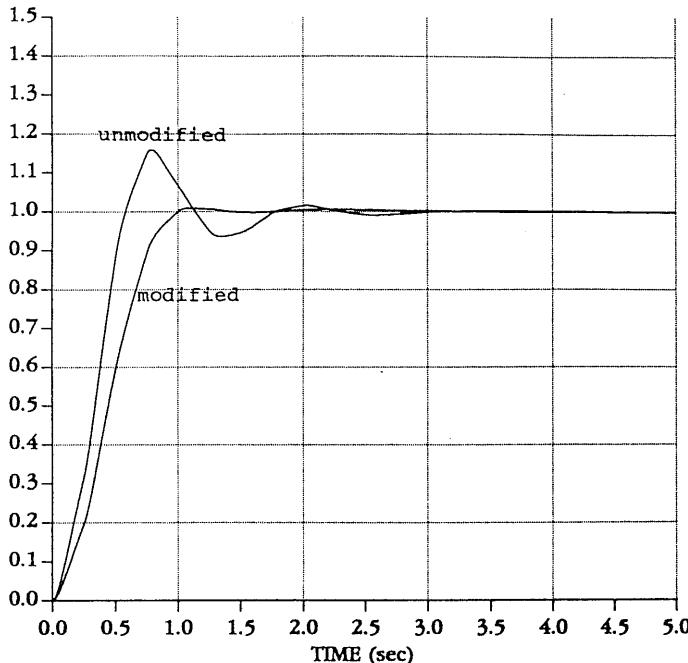


Fig. 7.4-7 Response of digital controller using modified and unmodified continuous-time design.

It should also be realized that in contrast to the situation in Example 7.3-2, which relied on the continuous design from Example 5.5-3, the sampling period is now needed to write the continuous dynamics (5), and hence to design the continuous-time controller.

c. *Digital Controller.* The modified continuous controller just designed is described by exactly the same equations as in Example 7.3-2, with, however, the modified gain K given in (11). Thus the new digital controller is exactly the same as the one described in that example, though using the modified gains.

To examine the performance of the modified digital controller, we may use the driver program described in Section 7.2, along with the continuous-time aircraft dynamics and the subroutine DIG(IK, T, X) from Example 7.3-2 with the gains in (11). The response for $T = 0.25$ s is shown in Fig. 7.4-7. Note that at this design sample period of $T = 0.25$ s, the digital control response is much like the response using the continuous-time controller shown in Fig. 7.4-6. It is important to note, however, that, using the digital controller, the delay noted in Fig. 7.4-6 does not appear.

For comparison we have also shown in Fig. 7.4-7 the unacceptable response from Example 7.3-2 for $T = 0.25$ s. This was the result of using a

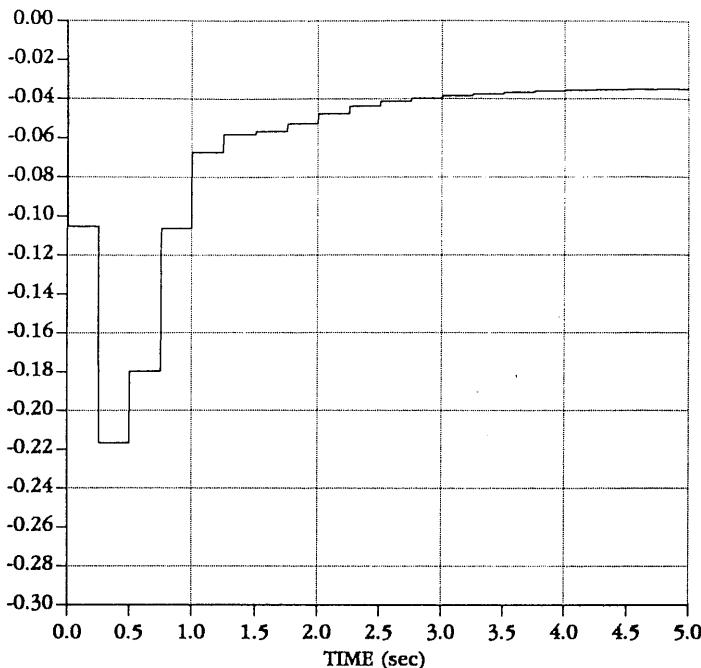


Fig. 7.4-8 Control input $u(t)$ required for modified digital pitch rate controller.

digital controller, obtained simply by applying the BLT to the unmodified continuous-time controller, which did not take into account the effects of the hold delay.

The control input $u(t)$ required in the modified digital controller with $T = 0.25$ s is shown in Fig. 7.4-8. It may be compared to the control signals in Example 7.3-2.

Clearly, the response shown in Fig. 7.4-7 obtained using modified continuous design is excellent. It far surpasses the digital control response in Example 7.3-2 for $T = 0.25$ s. Thus we have demonstrated that a sensible technique for taking into account some of the properties of the sample-and-hold process in the design stage of the continuous controller results in improved digital controllers that may be used with larger sample periods T . ■

7.5 IMPLEMENTATION CONSIDERATIONS

In this chapter we have discussed a design approach for digital controllers that is based on discretizing a continuous-time controller using the BLT or MPZ. It now behooves us to consider some practical considerations involved

with implementing the digital controller. Our discussion will necessarily be brief, giving only an indication of some of the issues. More detail may be discovered in [Aström and Wittenmark, 1984; Franklin and Powell, 1980; Franklin et al., 1986; Phillips and Nagle, 1984; Hanselmann, 1987; Lewis, 1992; Slivinsky and Borninski, 1987]. We shall mention actuator saturation and windup and controller realization structures.

Actuator Saturation and Windup

Actuator saturation is a problem that occurs in both continuous-time and digital control systems. Since it is easy to protect against by using a digital controller, we have placed it in this section.

A digital controller may be represented in the dynamic state-space form

$$x_{k+1} = Fx_k + Gw_k \quad (7.5-1)$$

$$u_k = Cx_k + Dw_k, \quad (7.5-2)$$

where $x_k \in \mathbb{R}^n$ is the controller state and w_k the controller input, composed generally of the tracking error and the plant measured output.

We have assumed thus far that the plant control input $u_k \in \mathbb{R}^m$ which is computed by the controller can actually be applied to the plant. However, in flight controls the plant inputs (such as elevator deflection δ_e , throttle, and so on) are limited by *maximum* and *minimum* allowable values. Thus the relation between the *desired plant input* v_k and the *actual plant input* u_k is given by the sort of behavior shown in Fig. 7.5-1, where u_H and u_L represent, respectively, the maximum and minimum control effort allowed by the mechanical actuator. Thus, to describe the actual case in an aircraft flight control system, we are forced to include *nonlinear saturation functions* in the control channels as shown in Fig. 7.5-2.

Consider the simple case where the controller is an integrator with input w_k and output v_k . Then all is well as long as v_k is between u_L and u_H , for in this region the aircraft input u_k equals v_k . However, if v_k exceeds u_H , then u_k is limited to its maximum value u_H . This in itself may not be a problem. The problem arises if w_k remains positive, for then the integrator continues

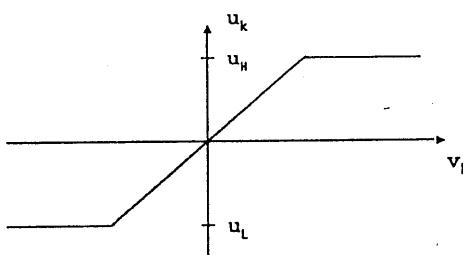


Fig. 7.5-1 Actuator saturation function.

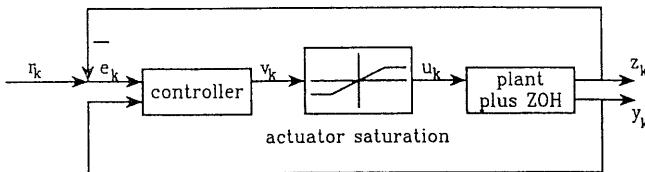


Fig. 7.5-2 Flight control system including actuator saturation.

to integrate and v_k may increase well beyond u_H . Then, when w_k becomes negative, it may take considerable time for v_k to decrease below u_H . In the meantime, u_k is held at u_H , giving an incorrect control input to the aircraft. This effect of integrator saturation is called *windup*. It arises because the controllers we design are generally dynamical in nature, which means that they store information or energy.

To correct integrator windup, it is necessary to *limit the state of the controller* so that it is consistent with the saturation effects being experienced by the plant input u_k . This is not difficult to achieve [Åström and Wittenmark, 1984]. Indeed, write (7.5-2) in the form

$$0 = u_k - Cx_k - Dw_k,$$

multiply it by L , which shall soon be selected, and add it to (7.5-1) to obtain

$$x_{k+1} = (F - LC)x_k + (G - LD)w_k + Lu_k. \quad (7.5-3)$$

This is the form in which the digital controller should be implemented to avoid windup, as we now argue. A little thought shows that actuator windup occurs in the form (7.5-1) when F is not asymptotically stable. For then, as long as w_k in (7.5-1) is nonzero, x_k will continue to increase. However, by selecting L so that

$$F_0 = F - LC \quad (7.5-4)$$

is asymptotically stable, this problem is averted.

A special case occurs when L is selected so that F_0 has all poles at the origin. Then x_k displays *deadbeat behavior*; after n time steps it remains limited to an easily computed value dependent on the values of w_k and u_k (see the Problems).

The *antiwindup gain* L may be selected to place the poles of F_0 arbitrarily if (C, F) is observable. However, as long as (C, F) is *detectable* (i.e., has all its unstable poles observable), windup may be eliminated using this technique.

To complete the design for antiwindup protection, the digital controller should be implemented in the form (7.5-3) and the aircraft control input then

selected according to

$$u_k = \text{sat}(Cx_k + Dw_k), \quad (7.5-5)$$

where the *saturation function* (shown in Fig. 7.5-1) is defined for scalars as

$$\text{sat}(v) = \begin{cases} u_H, & v \geq u_H \\ v, & u_L < v < u_H \\ u_L, & v \leq u_L, \end{cases} \quad (7.5-6)$$

with u_H and u_L the maximum and minimum allowable values, respectively. For vectors, the saturation function is defined as

$$\text{sat}(v) = \begin{bmatrix} \text{sat}(v_1) \\ \text{sat}(v_2) \\ \vdots \\ \text{sat}(v_m) \end{bmatrix}. \quad (7.5-7)$$

The values of u_H and u_L for each component v_i should be selected to correspond to the actual limits on the components of the plant input u_k .

Note that the limited signal u_k is used in (7.5-3), providing a feedback arrangement in the controller with antiwindup protection. What we have in effect done is include an *observer* with dynamics F_0 in the digital controller. Since F_0 is asymptotically stable, the observer will provide reasonable “estimates” even in the event of saturation.

Where u_k is not saturated, the controller with antiwindup compensation (7.5-3), (7.5-5) is identical to (7.5-1), (7.5-2).

If the controller is given in transfer function form

$$R(z^{-1})u_k = T(z^{-1})r_k - S(z^{-1})w_k, \quad (7.5-8)$$

where r_k is the reference command and z^{-1} is interpreted in the time domain as a unit delay of T seconds, antiwindup compensation may be incorporated as follows.

Select a desired stable observer polynomial $A_0(z^{-1})$ and add $A_0(z^{-1})u_k$ to both sides to obtain

$$A_0u_k = Tr_k - Sw_k + (A_0 - R)u_k. \quad (7.5-9)$$

A regulator with antiwindup compensation is then given by

$$A_0v_k = Tr_k - Sw_k + (A_0 - R)u_k \quad (7.5-10)$$

$$u_k = \text{sat}(v_k). \quad (7.5-11)$$

Example 7.5-1: Antiwindup Compensation for Digital PI Controller. From Example 7.3-1 a general digital PI controller is given by

$$u_k = k \left[1 + \frac{T}{T_I} \frac{1}{z - 1} \right] W_k, \quad (1)$$

where we have used design by the modified MPZ to obtain a delay of T seconds in the integrator to allow for computation time. The proportional gain is k and the reset time is T_I ; both are fixed in the design stage.

Multiply by z^{-1} and write

$$(1 - z^{-1}) u_k = k \left[(1 - z^{-1}) + \frac{Tz^{-1}}{T_I} \right] W_k, \quad (2)$$

which is in the transfer function form (7.5-8). The corresponding difference equation form for implementation is

$$u_k = u_{k-1} + kw_k + k \left(-1 + \frac{T}{T_I} \right) w_{k-1}. \quad (3)$$

This controller will experience windup problems since the autoregressive polynomial $R = 1 - z^{-1}$ has a root at $z = 1$, making it marginally stable. Thus, when u_k is limited, the integrator will continue to integrate, “winding up” beyond the saturation level.

To correct this problem, select an observer polynomial of

$$A_0(z^{-1}) = 1 - \alpha z^{-1}, \quad (4)$$

which has a pole at some desirable location $|\alpha| < 1$. The design parameter α may be selected by simulation studies. Then the controller with antiwindup protection (7.5-10)/(7.5-11) is given by

$$(1 - \alpha z^{-1}) v_k = k \left[1 + (-1 + T/T_I) z^{-1} \right] w_k + (1 - \alpha) z^{-1} u_k \quad (5)$$

$$u_k = \text{sat}(v_k). \quad (6)$$

The corresponding difference equations for implementation are

$$v_k = kw_k + \alpha v_{k-1} + k \left(-1 + \frac{T}{T_I} \right) w_{k-1} + (1 - \alpha) u_{k-1} \quad (7)$$

$$u_k = \text{sat}(v_k). \quad (8)$$

A few lines of FORTRAN code implementing this digital controller are given in Fig. 7.5-3. This subroutine may be used as the control update routine DIG with the digital simulation driver program in Section 7.2.

C DIGITAL PI CONTROLLER WITH ANTIWINDUP COMPENSATION

```

SUBROUTINE CONUP(T)
REAL K
COMMON/CONTROL/U
COMMON/OUTPUT/Z
COMMON/REF/R
DATA K,AL,TI,ULOW,UHIGH/ 0.5, 0.2, 5., -0.5, 0.5/
E= R-Z
V= K*E + V
U= AMAX1(ULOW,V)
U= AMIN1(UHIGH,U)
V= AL*V + K*(-1 + T/TI)*E + (1-AL)*U

RETURN
END

```

Fig. 7.5-3 FORTRAN code implementing PI controller with antiwindup compensation.

If $\alpha = 1$, we obtain the special case (2), which is called the *position form* and has no antiwindup compensation.

If $\alpha = 0$, we obtain the *deadbeat antiwindup compensation*

$$v_k = k[1 + (-1 + T/T_I)z^{-1}]w_k + u_{k-1}, \quad (9)$$

with corresponding difference equation implementation

$$v_k = u_{k-1} + kw_k + \left(-1 + \frac{T}{T_I}\right)w_{k-1}. \quad (10)$$

If u_k is not in saturation, this amounts to updating the plant control by adding the second and third terms in (10) to u_{k-1} . These terms are, therefore, nothing but $u_k - u_{k-1}$. The compensator with $\alpha = 0$ is thus called the *velocity form* of the PI controller. ■

Controller Realization Structures

Round-off errors can occur every time an arithmetic operation is performed. Moreover, since all the stable behavior of a discrete system is described by the location of the poles within the unit circle, great accuracy is required in the filter coefficients to obtain desired closed-loop pole locations.

A direct implementation of the digital filter would involve simply writing n difference equations describing (7.5-1)/(7.5-2) and would be virtually guaranteed to have severe numerical problems if n is large. Specifically, the controller and observer canonical forms [Kailath, 1980] are notoriously unstable numerically. That is, their poles are very sensitive to small variations in their coefficients. It can be shown that the sensitivity to coefficient variations

of the impulse response and frequency response is also high in direct implementations [Hanselmann, 1987]. For good numerical performance with fixed-point arithmetic, digital filters should be implemented as *cascade or parallel combinations of first- and second-order filters*.

A state-space transformation may be used to place the digital filter into an appropriate form for implementation. To obtain real coefficients, the *real Jordan form* is suitable [Phillips and Nagle, 1984; Hanselmann, 1987; Lewis, 1992]. This is a block diagonal form for (7.5-1), (7.5-2) having first- and second-order blocks in cascade and parallel. Corresponding to each real pole, there will be first-order blocks, and corresponding to each complex pole there will be second-order blocks.

A form suitable for implementation may also be found by performing a partial fraction expansion (PFE) on the transfer function. A technique for doing this in terms of the eigenstructure is given in Section 5.2. However, a *real PFE* should be found which will have the form (in the case of a simple matrix F)

$$H(z) = D + \sum_{i=1}^r H_i(z), \quad (7.5-12)$$

where $H_i(z)$ is first-order for real poles and second-order for complex poles. We are therefore concerned with implementing first-order filters of the form

$$H_1(z^{-1}) = \frac{b_1 z^{-1}}{1 + a_1 z^{-1}} \quad (7.5-13)$$

and second-order filters of the form

$$H_2(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}. \quad (7.5-14)$$

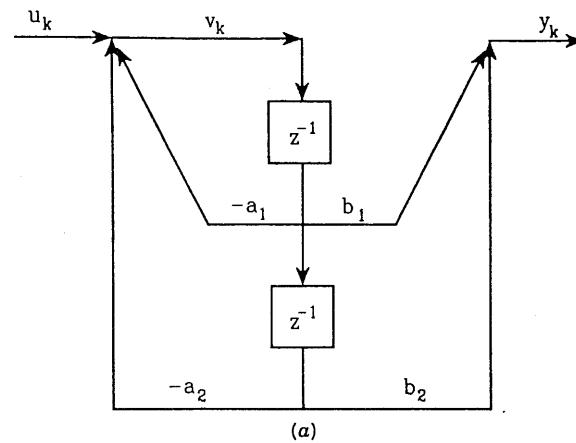
To implement $H_1(z^{-1})$, we may write

$$\begin{aligned} y_k &= H_1(z^{-1})u_k \\ (1 + a_1 z^{-1})y_k &= b_1 z^{-1}u_k \\ y_k &= -a_1 y_{k-1} + b_1 u_{k-1}, \end{aligned} \quad (7.5-15)$$

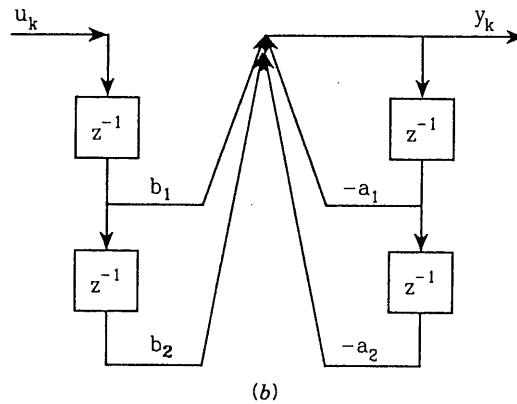
which is a difference equation that may easily be programmed.

There are many ways to implement the second-order transfer function [Phillips and Nagle, 1984]. Among these are *direct forms 1 through 4* (denoted D1, D2, D3, D4) and *cross-coupled forms 1 and 2* (denoted X1, X2).

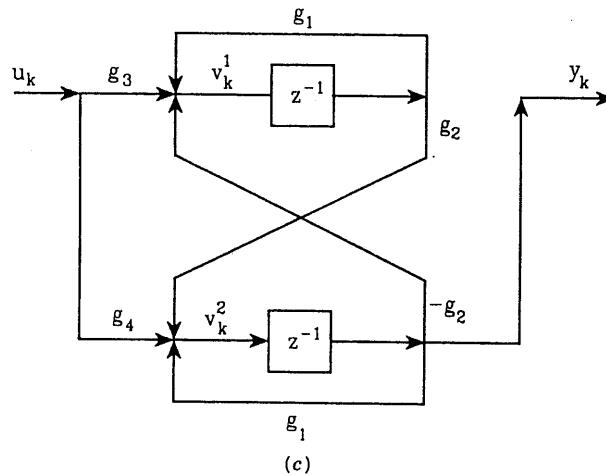
In Fig. 7.5-4 we show D1, D3, and X1. Forms D2, D4, and X2, respectively, are their duals (i.e., all arrows are reversed and the roles of the input



(a)



(b)



(c)

Fig. 7.5-4 Implementations of second-order digital filters: (a) direct form 1, D1; (b) direct form 3, D3; (c) cross-coupled form 1, X1.

TABLE 7.5-1. Elements of Second-Order Modules

	Structure		
	D1	D3	X1
Time-delay elements	2	4	2
Multipliers	4	4	6
Summing junctions	2	1	2

and the output are interchanged). In Table 7.5-1 we give a comparison of the number of time-delay elements, multipliers, and summing junctions for each form. Note that D1 and X1 conserve time-delay elements, while D3 conserves summing junctions.

The difference equation implementations of these second-order modules are given in Table 7.5-2, with $y_k = H_2(z^{-1})u_k$. It is interesting that the difference equations for the X1 module may be written from the complex PFE (and hence, with a little manipulation, from the usual complex Jordan form).

When implementing these modules on a fixed-point microprocessor, it is important to incorporate overflow protection [Slivinsky and Borninski, 1986]. When interconnecting them to produce $H(z)$, scaling may be introduced between the modules [Phillips and Nagle, 1984].

TABLE 7.5-2. Difference Equation Implementation of Second-Order Modules

D1:

$$\begin{aligned}v_k &= -a_1v_{k-1} - a_2v_{k-2} + u_k \\y_k &= b_1v_{k-1} + b_2v_{k-2}\end{aligned}$$

D3:

$$y_k = -a_1y_{k-1} - a_2y_{k-2} + b_1u_{k-1} + b_2u_{k-2}$$

X1:

$$\begin{aligned}v_k^1 &= g_1v_{k-1}^1 - g_2v_{k-1}^2 + g_3u_k \\v_k^2 &= g_1v_{k-1}^2 + g_2v_{k-1}^1 + g_4u_k \\y_k &= v_{k-1}^2\end{aligned}$$

where g_i are defined by

$$H_2(z^{-1}) = \frac{Nz^{-1}}{1 + pz^{-1}} + \frac{N^*z^{-1}}{1 + p^*z^{-1}}$$

$$g_1 = -\operatorname{Re}(p)$$

$$g_2 = -\operatorname{Im}(p)$$

$$g_3 = 2\operatorname{Im}(N)$$

$$g_4 = 2\operatorname{Re}(N)$$

7.6 SUMMARY

Since most aircraft control systems are implemented using digital signal processors, in this chapter we have outlined the basics of digital control. In Section 7.2 we discussed how to simulate digital control schemes using a Runge-Kutta integrator on the continuous aircraft dynamics. This approach yields the time responses not only at the sample points, but also between the samples. It is important to check the intersample performance of the closed-loop system before implementing a digital controller on an aircraft, since it can be unsatisfactory even though all is well at the sample points.

In Section 7.3 we gave a design technique for digital controllers that is based on *redesign of an existing continuous-time controller* by discretizing it using approximation techniques like the BLT and MPZ. This results in digital controllers that require small sample periods to work properly. To overcome the requirement for unreasonably small sample periods, in Section 7.4 we showed how to modify the continuous-time controller so that, after discretization, a better digital controller is obtained that works for larger sample periods. This modification allowed the delay properties of the sample-and-hold process to be taken into account.

Finally, in Section 7.5 we mentioned some digital controller implementation considerations. We showed how to design controllers with antiwindup protection to overcome the problems of saturation of the control signals due to control limitations such as elevator deflection stops and throttle maximum limits. We gave some low-order controller structures that allow the implementation of digital controllers with maximum accuracy and efficiency.

REFERENCES

- Åström, K. J., and B. Wittenmark, *Computer Controlled Systems*. Englewood Cliffs, N.J.: Prentice Hall, 1984.
- Franklin, G. F., and J. D. Powell, *Digital Control*. Reading, Mass.: Addison-Wesley, 1980.
- Franklin, G. F., J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Reading, Mass.: Addison-Wesley, 1980.
- Hanselmann, H., "Implementation of digital controllers: a survey," *Automatica*, vol. 23, no. 1, 1987, pp. 7-32.
- Kailath, T., *Linear Systems*, Englewood Cliffs, N.J.: Prentice Hall, 1980.
- Lewis, F. L., *Applied Optimal Control and Estimation*. Englewood Cliffs, N.J.: Prentice Hall, 1992.
- Oppenheim, A. V., and R. W. Schafer, *Digital Signal Processing*, Englewood Cliffs, N.J.: Prentice-Hall, 1975.
- Phillips, C. L., and H. T. Nagle, Jr., *Digital Control System Analysis and Design*, Englewood Cliffs, N.J.: Prentice-Hall, 1984.

Slivinsky, C., and J. Borninski, "Control system compensation and implementation with the TMS32010," in *Digital Signal Processing Applications*, K.-S. Lin ed., Englewood Cliffs, N.J.: Prentice Hall, 1987.

Su, K. L., *Time-Domain Synthesis of Linear Networks*. Englewood Cliffs, N.J.: Prentice Hall, 1971.

PROBLEMS

Section 7.3

- 7.3-1** Prove (7.3.13)/(7.3.14) by taking the Laplace transform of $\dot{x} = Ax + Bu$ and then using the BLT.
- 7.3-2** **Digital Pitch-Rate Controller.** Design a digital pitch rate controller (see Example 7.3-2) using the MPZ technique. Simulate the step response for a few sample periods T and compare to the digital controller designed using the BLT.
- 7.3-3** **Digital Normal Acceleration CAS.** A normal acceleration CAS was designed in Example 5.4-1.
- Using the BLT, design a digital normal acceleration CAS controller. Simulate the time response for various sampling periods.
 - Repeat using the modified MPZ.
- 7.3-4** **Digital Wing Leveler.** A wing leveler was designed in Example 5.5-4.
- Using the BLT, design a digital wing leveler. Simulate the time response for various sampling periods.
 - Repeat using the modified MPZ.

Section 7.4

- 7.4-1** A Padé approximant for e^{-sT} is

$$G(s) = \frac{1 - 2sT/3 + (sT)^2/6}{1 + sT/3}.$$

- Use long division to determine how many terms of the Taylor series of e^{-sT} are matched by $G(s)$.
 - Use $G(s)$ to derive one of the approximants for the ZOH plus sampler shown in Table 7.4-2. How many terms of the Taylor series are matched by this approximant?
- 7.4-2** **Digital Normal Acceleration CAS.** A normal acceleration CAS was designed in Example 5.4-1. Using the BLT, design a digital normal acceleration CAS controller. Use modified continuous design, includ-

ing the hold delay. Use $T = 0.1$ s. Simulate the time response and compare to the results of Problem 7.3-3.

- 7.4-3 Digital Wing Leveler.** A wing leveler was designed in Example 5.5-4. Using the modified MPZ, design a digital wing leveler. Use modified continuous design, including the hold delay. Use $T = 0.1$ s. Simulate the time response and compare to the results of Problem 7.3-4.

Section 7.5

7.5-1 Antiwindup Compensator

- (a) Write down the value of the state x_k in the antiwindup controller (7.5-3) for the deadbeat case where F_0 has all poles at the origin. Assume that w_k and u_k are constant and that $k > n$. Note that if F_0 has all poles at the origin, then $F_0^n = 0$, where n is the dimension of F_0 .
- (b) Repeat for the case where the controller is just an integrator so that $x_{k+1} = x_k + (T/T_I)w_k$, $u_k = \text{sat}(x_k)$. Simplify as far as possible.

- 7.5-2** Show how to determine the X1 difference equations in Table 7.5-2 directly from the complex Jordan form blocks corresponding to a complex pair of poles.

- 7.5-3 Antiwindup Protection for Normal Acceleration CAS.** In Example 5.4-1 a normal acceleration CAS was designed; it had a PI controller in the feedforward loop. In the problems for Section 7.3 this design was digitized.

- (a) Modify the digital normal acceleration CAS to add antiwindup protection.
- (b) Now, set limits into the elevator actuator in your simulation program. Obtain time responses with and without the antiwindup protection.

- 7.5-4 Antiwindup Protection for Pitch-Rate CAS.** Repeat the previous problem for the pitch-rate controller in Example 5.5-3, which was digitized in the problems for Section 7.3.

APPENDIX A

F-16 MODEL

This appendix contains the remainder of the data for the F-16 aircraft model given in Chapter 3. The usage of the lookup tables will be made evident by referring to the aircraft model. These data, Appendix B, and the other programs used in this book can be obtained on a floppy disc, at a nominal cost, from Dr. B. L. Stevens, 1051 Park Manor Terr., Marietta, GA 30064.

Mass Properties

$$\text{Weight (lbs)}: \quad W = 20,500$$

$$\text{Moments of Inertia (slug-ft}^2\text{)}: \quad J_{xx} = 9,496$$

$$J_{yy} = 55,814$$

$$J_{zz} = 63,100$$

$$J_{xz} = 982$$

Wing Dimensions

$$\text{Span} = 30 \quad \text{ft}$$

$$\text{Area} = 300 \quad \text{ft}^2$$

$$\text{m.a.c} = 11.32 \text{ ft}$$

Reference CG Location

$$X_{cg} = 0.35\bar{c}$$

Control Surface Actuator Models

	deflection limit	rate limit	time const.
Elevator	$\pm 25.0^{\circ}$,	$60^{\circ}/s$,	0.0495 s lag
Ailerons	$\pm 21.5^{\circ}$,	$80^{\circ}/s$,	0.0495 s lag
Rudder	$\pm 30.0^{\circ}$,	$120^{\circ}/s$,	0.0495 s lag

Engine Angular Momentum

Assumed fixed at 160 slug-ft²/s

Standard Atmosphere Model

```

SUBROUTINE ADC(VT,ALT,AMACH,QBAR) ! air data computer
DATA R0/2.377E-3/ ! sea-level density
TFAC = 1.0 - 0.703E-5 * ALT
T    = 519.0 * TFAC ! temperature
IF (ALT .GE. 35000.0) T= 390.0
RHO = R0 * (TFAC**4.14) ! density
AMACH= VT/SQRT(1.4*1716.3*T) ! Mach number
QBAR = 0.5*RHO*VT*VT ! dynamic pressure
C      PS   = 1715.0 * RHO * T ! static pressure
      RETURN
      END

```

Engine Model

The F-16 engine power response is modeled by a first-order lag (in function PDOT, given below), the rest of the model consists of the throttle gearing (in TGEAR) and the lookup tables for thrust as a function of operating power level, altitude, and Mach (in THRUST). In the thrust-lookup tables the rows correspond to a Mach number variation from 0 to 1.0 in increments of 0.2, and the columns correspond to altitudes from 0 to 50,000 ft in increments of 10,000 ft. There is a table for each of the power levels "idle," "military," and "maximum." The accompanying linear interpolation algorithm can extrapolate beyond the boundaries of a table, but the results may not be realistic.

```
FUNCTION TGEAR(THTL) ! Power command v. thtl. relationship
IF (THTL.LE.0.77) THEN
  TGEAR = 64.94*THTL
ELSE
  TGEAR = 217.38*THTL-117.38
END IF
RETURN
END

FUNCTION PDOT(P3,P1) ! PDOT= rate of change of power
IF (P1.GE.50.0) THEN ! P3= actual power, P1= power command
  IF (P3.GE.50.0) THEN
    T=5.0
    P2=P1
  ELSE
    P2=60.0
    T=RTAU(P2-P3)
  END IF
ELSE
  IF (P3.GE.50.0) THEN
    T=5.0
    P2=40.0
  ELSE
    P2=P1
    T=RTAU(P2-P3)
  END IF
END IF
PDOT=T*(P2-P3)
RETURN
END

FUNCTION RTAU(DP) ! used by function PDOT
IF (DP.LE.25.0)THEN
  RTAU=1.0          ! reciprocal time constant
ELSE IF (DP.GE.50.0)THEN
  RTAU=0.1
ELSE
  RTAU=1.9-.036*DP
END IF
RETURN
END
```

```

FUNCTION THRUST(POW,ALT,RMACH) ! Engine thrust model
REAL A(0:5,0:5), B(0:5,0:5), C(0:5,0:5)
DATA A/
+ 1060.0,   670.0,   880.0,   1140.0,   1500.0,   1860.0,
+ 635.0,    425.0,   690.0,   1010.0,   1330.0,   1700.0,
+ 60.0,     25.0,   345.0,   755.0,   1130.0,   1525.0,
+ -1020.0,  -710.0,  -300.0,   350.0,   910.0,   1360.0,
+ -2700.0,  -1900.0, -1300.0,  -247.0,   600.0,   1100.0,
+ -3600.0,  -1400.0, -595.0,  -342.0,  -200.0,   700.0/
C      mil data now
DATA B/
+ 12680.0,  9150.0,  6200.0,  3950.0,  2450.0,  1400.0,
+ 12680.0,  9150.0,  6313.0,  4040.0,  2470.0,  1400.0,
+ 12610.0,  9312.0,  6610.0,  4290.0,  2600.0,  1560.0,
+ 12640.0,  9839.0,  7090.0,  4660.0,  2840.0,  1660.0,
+ 12390.0,  10176.0, 7750.0,  5320.0,  3250.0,  1930.0,
+ 11680.0,  9848.0,  8050.0,  6100.0,  3800.0,  2310.0/
C      max data now
DATA C/
+ 20000.0, 15000.0, 10800.0, 7000.0, 4000.0, 2500.0,
+ 21420.0, 15700.0, 11225.0, 7323.0, 4435.0, 2600.0,
+ 22700.0, 16860.0, 12250.0, 8154.0, 5000.0, 2835.0,
+ 24240.0, 18910.0, 13760.0, 9285.0, 5700.0, 3215.0,
+ 26070.0, 21075.0, 15975.0, 11115.0, 6860.0, 3950.0,
+ 28886.0, 23319.0, 18300.0, 13484.0, 8642.0, 5057.0/
C
      H = .0001*ALT
      I = INT(H)
      IF(I.GE.5) I=4
      DH= H-FLOAT(I)
      RM= 5.0*RMACH
      M = INT(RM)
      IF(M.GE.5) M=4
      DM= RM-FLOAT(M)
      CDH=1.0-DH
      S= B(I,M) *CDH + B(I+1,M) *DH
      T= B(I,M+1)*CDH + B(I+1,M+1)*DH
      TMIL= S + (T-S)*DM
      IF( POW .LT. 50.0 ) THEN
      S= A(I,M) *CDH + A(I+1,M) *DH
      T= A(I,M+1)*CDH + A(I+1,M+1)*DH
      TIDL= S + (T-S)*DM
      THRUST=TIDL+(TMIL-TIDL)*POW*.02
      ELSE
      S= C(I,M) *CDH + C(I+1,M) *DH
      T= C(I,M+1)*CDH + C(I+1,M+1)*DH
      TMAX= S + (T-S)*DM
      THRUST=TMIL+(TMAX-TMIL)*(POW-50.0)*.02
      END IF
      RETURN
      END

```

Aerodynamic Data

The aerodynamic data tables and associated interpolation algorithms, given below, will provide values for the body-axes dimensionless aerodynamic coefficients of the F-16 model at arbitrary values of the independent variables. The angle-of-attack range of the tables is from -10° to 45° in 5° increments, and the sideslip angle range is from -30° to 30° in either 5° or 10° increments. The given interpolation algorithm interpolates linearly between the data points; it will extrapolate beyond the table boundaries, but the results may be unrealistic.

```

SUBROUTINE DAMP(ALPHA, D) ! various damping coefficients
REAL A(-2:9,9),D(9)
DATA A/
& -.267,   -.110,    .308,    1.34,    2.08,    2.91,    2.76,
&  2.05,    1.50,    1.49,    1.83,    1.21,
& .882,    .852,    .876,    .958,    .962,    .974,    .819,
& .483,    .590,    1.21,   - .493,   -1.04,
& -.108,   - .108,   - .188,   .110,    .258,    .226,    .344,
& .362,    .611,    .529,    .298,   -2.27,
& -8.80,   -25.8,   -28.9,   -31.4,   -31.2,   -30.7,   -27.7,
& -28.2,   -29.0,   -29.8,   -38.3,   -35.3,
& -.126,   -.026,   .063,    .113,    .208,    .230,    .319,
& .437,    .680,    .100,    .447,   - .330,
& -.360,   -.359,   -.443,   -.420,   -.383,   -.375,   -.329,
& -.294,   -.230,   -.210,   -.120,   -.100,
& -7.21,   -.540,   -5.23,   -5.26,   -6.11,   -6.64,   -5.69,
& -6.00,   -6.20,   -6.40,   -6.60,   -6.00,
& -.380,   -.363,   -.378,   -.386,   -.370,   -.453,   -.550,
& -.582,   -.595,   -.637,   -1.02,   -.840,
& .061,    .052,    .052,   - .012,   - .013,   -.024,    .050,
& .150,    .130,    .158,    .240,    .150/
C
S= 0.2 * ALPHA
K= INT(S)
IF(K .LE. -2) K= -1
IF(K .GE.  9) K=  8
DA= S - FLOAT(K)
L = K + INT( SIGN(1.1,DA) )
DO 1, I= 1,9
1  D(I)= A(K,I) + ABS(DA) * (A(L,I) - A(K,I))
END
C
C D1= CXq; D2= CYr; D3= CYP; D4= CZq; D5= Clr; D6= Clp
C D7= Cmq; D8= Cnr; D9= Cnp

```

```

FUNCTION CX(ALPHA,EL) ! x-axis aerodynamic force coeff.
REAL A(-2:9,-2:2)
DATA A/
& -.099, -.081, -.081, -.063, -.025, .044, .097,
& .113, .145, .167, .174, .166,
& -.048, -.038, -.040, -.021, .016, .083, .127,
& .137, .162, .177, .179, .167,
& -.022, -.020, -.021, -.004, .032, .094, .128,
& .130, .154, .161, .155, .138,
& -.040, -.038, -.039, -.025, .006, .062, .087,
& .085, .100, .110, .104, .091,
& -.083, -.073, -.076, -.072, -.046, .012, .024,
& .025, .043, .053, .047, .040/
C
S= 0.2 * ALPHA
K= INT(S)
IF(K .LE. -2) K= -1
IF(K .GE. 9) K= 8
DA= S - FLOAT(K)
L = K + INT( SIGN(1.1,DA) )
S= EL/12.0
M= INT(S)
IF(M .LE. -2) M= -1
IF(M .GE. 2) M= 1
DE= S - FLOAT(M)
N= M + INT( SIGN(1.1,DE) )
T= A(K,M)
U= A(K,N)
V= T + ABS(DA) * (A(L,M) - T)
W= U + ABS(DA) * (A(L,N) - U)
CX= V + (W-V) * ABS(DE)
RETURN
END

FUNCTION CY(BETA,AIL,RDR) ! sideforce coefficient
CY= -.02*BETA + .021*(AIL/20.0) + .086*(RDR/30.0)
END

FUNCTION CZ(ALPHA,BETA,EL) ! z-axis force coeff.
REAL A(-2:9)
DATA A/ .770, .241, -.100, -.416, -.731, -1.053,
& -1.366, -1.646, -1.917, -2.120, -2.248, -2.229/
S= 0.2 * ALPHA
K= INT(S)
IF(K .LE. -2) K= -1
IF(K .GE. 9) K= 8
DA= S - FLOAT(K)
L = K + INT( SIGN(1.1,DA) )
S= A(K) + ABS(DA) * (A(L) - A(K))
CZ= S*(1-(BETA/57.3)**2) - .19*(EL/25.0)
END

```

```

FUNCTION CM(ALPHA,EL) ! pitching moment coeff.
REAL A(-2:9,-2:2)
DATA A/
& .205, .168, .186, .196, .213, .251, .245,
& .238, .252, .231, .198, .192,
& .081, .077, .107, .110, .110, .141, .127,
& .119, .133, .108, .081, .093,
& -.046, -.020, -.009, -.005, -.006, .010, .006,
& -.001, .014, .000, -.013, .032,
& -.174, -.145, -.121, -.127, -.129, -.102, -.097,
& -.113, -.087, -.084, -.069, -.006,
& -.259, -.202, -.184, -.193, -.199, -.150, -.160,
& -.167, -.104, -.076, -.041, -.005/
C SAME INTERPOLATION AS CX *****
C
FUNCTION CL(ALPHA,BETA) ! rolling moment coeff.
REAL A(-2:9,0:6)
DATA A/12*0,
& -.001, -.004, -.008, -.012, -.016, -.022, -.022,
& -.021, -.015, -.008, -.013, -.015,
& -.003, -.009, -.017, -.024, -.030, -.041, -.045,
& -.040, -.016, -.002, -.010, -.019,
& -.001, -.010, -.020, -.030, -.039, -.054, -.057,
& -.054, -.023, -.006, -.014, -.027,
& .000, -.010, -.022, -.034, -.047, -.060, -.069,
& -.067, -.033, -.036, -.035, -.035,
& .007, -.010, -.023, -.034, -.049, -.063, -.081,
& -.079, -.060, -.058, -.062, -.059,
& .009, -.011, -.023, -.037, -.050, -.068, -.089,
& -.088, -.091, -.076, -.077, -.076/
C
S= 0.2 * ALPHA
K= INT(S)
IF(K .LE. -2) K= -1
IF(K .GE. 9) K= 8
DA= S - FLOAT(K)
L = K + INT( SIGN(1.1,DA) )
S= .2* ABS(BETA)
M= INT(S)
IF(M .EQ. 0) M= 1
IF(M .GE. 6) M= 5
DB= S - FLOAT(M)
N= M + INT( SIGN(1.1,DB) )
T= A(K,M)
U= A(K,N)
V= T + ABS(DA) * (A(L,M) - T)
W= U + ABS(DA) * (A(L,N) - U)
DUM= V + (W-V) * ABS(DB)
CL= DUM * SIGN(1.0,BETA)
RETURN
END

```

```

FUNCTION CN(ALPHA,BETA) ! yawing moment coeff.
REAL A(-2:9,0:6)
DATA A/12*0,
& .018, .019, .018, .019, .019, .018, .013,
& .007, .004, -.014, -.017, -.033, .039, .030,
& .038, .042, .042, .042, .043, .039, .030,
& .017, .004, -.035, -.047, -.057, .053, .032,
& .056, .057, .059, .058, .058, .053, .032,
& .012, .002, -.046, -.071, -.073, .057, .029,
& .064, .077, .076, .074, .073, .062, .049,
& .007, .012, -.034, -.065, -.041, .080, .068,
& .074, .086, .093, .089, .080, .062, .049,
& .022, .028, -.012, -.002, -.013, .096, .080, .068,
& .079, .090, .106, .106, .096, .080, .068,
& .030, .064, .015, .011, -.001/
C NOW USE SAME INTERPOLATION AS CL *****
C
FUNCTION DLDA(ALPHA,BETA) ! rolling mom. due to ailerons
REAL A(-2:9,-3:3)
DATA A/-0.041, -0.052, -0.053, -0.056, -0.050, -0.056, -0.082,
& -0.059, -0.042, -0.038, -0.027, -0.017,
& -0.041, -0.053, -0.053, -0.053, -0.050, -0.051, -0.066,
& -0.043, -0.038, -0.027, -0.023, -0.016,
& -0.042, -0.053, -0.052, -0.051, -0.049, -0.049, -0.043,
& -0.035, -0.026, -0.016, -0.018, -0.014, -0.014,
& -0.040, -0.052, -0.051, -0.052, -0.048, -0.048, -0.042,
& -0.037, -0.031, -0.026, -0.017, -0.012,
& -0.043, -0.049, -0.048, -0.049, -0.043, -0.042, -0.042,
& -0.036, -0.025, -0.021, -0.016, -0.011,
& -0.044, -0.048, -0.048, -0.047, -0.042, -0.041, -0.020,
& -0.028, -0.013, -0.014, -0.011, -0.010,
& -0.043, -0.049, -0.047, -0.045, -0.042, -0.037, -0.003,
& -0.013, -0.010, -0.003, -0.007, -0.008/
S= 0.2 * ALPHA
K= INT(S)
IF(K .LE. -2) K= -1
IF(K .GE. 9) K= 8
DA= S - FLOAT(K)
L = K + INT( SIGN(1.1,DA) )
S= 0.1 * BETA
M= INT(S)
IF(M .EQ. -3) M= -2
IF(M .GE. 3) M= 2
DB= S - FLOAT(M)
N= M + INT( SIGN(1.1,DB) )
T= A(K,M)
U= A(K,N)
V= T + ABS(DA) * (A(L,M) - T)
W= U + ABS(DA) * (A(L,N) - U)
DLDA= V + (W-V) * ABS(DB)
RETURN
END

```

```

FUNCTION DLDR(ALPHA,BETA) ! rolling moment due to rudder
REAL A(-2:9,-3:3) ! use same interpolation as DLDA *****
DATA A/ .005, .017, .014, .010, -.005, .009, .019,
& .005, -.000, -.005, -.011, .008,
& .007, .016, .014, .014, .013, .009, .012,
& .005, .000, .004, .009, .007,
& .013, .013, .011, .012, .011, .009, .008,
& .005, -.002, .005, .003, .005,
& .018, .015, .015, .014, .014, .014, .014,
& .015, .013, .011, .006, .001,
& .015, .014, .013, .013, .012, .011, .011,
& .010, .008, .008, .007, .003,
& .021, .011, .010, .011, .010, .009, .008,
& .010, .006, .005, .000, .001,
& .023, .010, .011, .011, .011, .010, .008,
& .010, .006, .014, .020, .000/

```

C

```

FUNCTION DNDA(ALPHA,BETA) ! yawing moment due to ailerons
REAL A(-2:9,-3:3) ! use same interpolation as DLDA *****
DATA A/ .001, -.027, -.017, -.013, -.012, -.016, .001,
& .017, .011, .017, .008, .016,
& .002, -.014, -.016, -.016, -.014, -.019, -.021,
& .002, .012, .016, .015, .011,
& -.006, -.008, -.006, -.006, -.005, -.008, -.005,
& .007, .004, .007, .006, .006,
& -.011, -.011, -.010, -.009, -.008, -.006, .000,
& .004, .007, .010, .004, .010,
& -.015, -.015, -.014, -.012, -.011, -.008, -.002,
& .002, .006, .012, .011, .011,
& -.024, -.010, -.004, -.002, -.001, .003, .014,
& .006, -.001, .004, .004, .006,
& -.022, .002, -.003, -.005, -.003, -.001, -.009,
& -.009, -.001, .003, -.002, .001/

```

C

```

FUNCTION DNDR(ALPHA,BETA) ! yawing moment due to rudder
REAL A(-2:9,-3:3)
DATA A/ -.018, -.052, -.052, -.052, -.054, -.049, -.059,
& -.051, -.030, -.037, -.026, -.013,
& -.028, -.051, -.043, -.046, -.045, -.049, -.057,
& -.052, -.030, -.033, -.030, -.008,
& -.037, -.041, -.038, -.040, -.040, -.038, -.037,
& -.030, -.027, -.024, -.019, -.013,
& -.048, -.045, -.045, -.045, -.044, -.045, -.047,
& -.048, -.049, -.045, -.033, -.016,
& -.043, -.044, -.041, -.041, -.040, -.038, -.034,
& -.035, -.035, -.029, -.022, -.009,
& -.052, -.034, -.036, -.036, -.035, -.028, -.024,
& -.023, -.020, -.016, -.010, -.014,
& -.062, -.034, -.027, -.028, -.027, -.027, -.023,
& -.023, -.019, -.009, -.025, -.010/

```

C NOW USE SAME INTERPOLATION AS DLDA *****

APPENDIX B

SOFTWARE

This appendix contains the FORTRAN code that is required to use the aircraft models given in the text and is not otherwise readily available. For the steady-state trim algorithm (Section B1) we give the basic trimmer subroutine, part of the constraint subroutine, a cost function, and the simplex minimization algorithm. The user must write a driver program and add additional flight path constraints, as required. In Section B2 a subroutine for numerical linearization is given, and the user need only add a driver program. Software for time-history simulation and control systems design is readily available from other sources and so, in the rest of this appendix, we have given only the Runge-Kutta algorithm that was used for most of the examples, two algorithms that can be used to simulate onboard inertial navigation as indicated in Example 3.7-2, and the discrete-time transition matrix algorithm.

Appendix A, Appendix B, and the programs used in this book can be obtained on a floppy disc, at a nominal cost, from Dr. B. L. Stevens, 1051 Park Manor Terr., Marietta, GA 30064.

B1. AIRCRAFT STEADY-STATE TRIM CODE

The subroutine "TRIMMER" (below) sets up a function minimization algorithm to determine a steady-state trim condition for either a 6-DOF or 3-DOF (longitudinal only) aircraft model. The subroutine arguments are the number of degrees of freedom (NV) and the "COST" function (which must

be declared "EXTERNAL" in the main program). Labeled COMMON storage is used to pass the state and control vectors to and from the main program and the cost function (and, in the case of the control vector, the aircraft model also).

The main program must initialize the state vector according to the trim condition required, the control vector can simply be set to zero initially. It must also set the turn-rate, roll-rate, or pull-up rate, set flags for coordinated turns or stability-axis roll, and pass these through a common block ("CONSTRNT") to the constraint routine. A simplex routine (given below) is used for function minimization, and it returns the coordinates of the cost function minimum in the simplex vector S. The cost function is then called once more to set the state and control vectors to their final values, and these values are passed through COMMON to the main program to be placed in a data file. Subroutine "SMPLX" can easily be replaced by "ZXMWD" from the IMSL library, or "AMOEBA" from "Numerical Recipes" if desired. The author is indebted to Dr. P. Vesty for this simplex routine.

```

SUBROUTINE TRIMMER (NV, COST)
PARAMETER (NN=20, MM=10)
EXTERNAL COST
CHARACTER*1 ANS
DIMENSION S(6), DS(6)
COMMON/ STATE/ X(NN)
COMMON/ CONTROLS/ U(MM)
COMMON/ OUTPUT/ AN, AY, AX, QBAR, AMACH ! common to aircraft
DATA RTOD /57.29577951/
S(1)= U(1)
S(2)= U(2)
S(3)= X(2)
IF(NV .LE. 3) GO TO 10
S(4)= U(3)
S(5)= U(4)
S(6)= X(3)
10   DS(1) = 0.2
DS(2) = 1.0
DS(3) = 0.02
IF(NV .LE. 3) GO TO 20
DS(4) = 1.0
DS(5) = 1.0
DS(6) = 0.02
20   NC= 1000
      WRITE(*,'(1X,A,$)')'Reqd. # of trim iterations (def. = 1000) : '
      READ(*,* ,ERR=20) NC
      SIGMA = -1.0
      CALL SMPLX(COST,NV,S,DS,SIGMA,NC,F0,FFIN)
      FFIN = COST(S)
      IF (NV .GT. 3) THEN
      WRITE(*,'(/1X,A)')'Throttle      Elevator,     Ailerons,      Rudder'
      WRITE(*,'(9X,4(1PE10.2,3X),/))' U(1), U(2), U(3), U(4)
      WRITE(*,99)'Angle of attack',RTOD*X(2), 'Sideslip angle',RTOD*X(3)
      WRITE(*,99)'Pitch angle', RTOD*X(5), 'Bank angle', RTOD*X(4)
      WRITE(*,99)'Normal acceleration', AN, 'Lateral acceln', AY
      WRITE(*,99)'Dynamic pressure', QBAR, 'Mach number', AMACH
      ELSE
      WRITE(*,'(/1X,A)')' Throttle      Elevator      Alpha      Pitch'
      WRITE(*,'(1X,4(1PE10.2,3X))')U(1),U(2),X(2)*RTOD,X(3)*RTOD
      WRITE(*,'(/1X,A)')'Normal acceleration      Dynamic Pressure      Mach '
      WRITE(*,'(5X,3(1PE10.2,7X))') AN,QBAR,AMACH
      END IF
      WRITE(*,99)'Initial cost function ',F0,'Final cost function',FFIN
99    FORMAT(2(1X,A22,1PE10.2))
40    WRITE(*,'(/1X,A,$)') 'More Iterations ? (def= Y) : '
      READ(*,'(A)',ERR= 40) ANS
      IF (ANS .EQ. 'Y' .OR. ANS .EQ. 'y' .OR. ANS .EQ. '/') GO TO 10
      IF (ANS .EQ. 'N' .OR. ANS .EQ. 'n') RETURN
      GO TO 40
      END

```

```

FUNCTION CLF16 (S)      ! F16 cost function (see text)
PARAMETER (NN=20)
REAL S(*), XD(NN)
COMMON/STATE/X(NN)      ! common to main program
COMMON/CONTROLS/THTL,EL,AIL,RDR      ! to aircraft
THTL = S(1)
EL   = S(2)
X(2) = S(3)
AIL  = S(4)
RDR  = S(5)
X(3) = S(6)
X(13)= TGEAR (THTL)
CALL  CONSTR (X)
CALL  F (TIME,X,XD)
CLF16 = XD(1)**2 + 100.0*( XD(2)**2 + XD(3)**2 )
&      + 10.0*( XD(7)**2 + XD(8)**2 + XD(9)**2 )
RETURN
END

SUBROUTINE CONSTR (X)    ! used by COST, to apply constraints
DIMENSION X(*)
LOGICAL COORD, STAB
COMMON/CNSTRNT/RADGAM,SINGAM,RR,PR,TR,PHI,CPhi,SPHI,COORD,STAB
C common to main program.
CALPH = COS(X(2))
SALPH = SIN(X(2))
CBETA = COS(X(3))
SBETA = SIN(X(3))
IF (COORD) THEN
  ! coordinated turn logic here
ELSE IF (TR .NE. 0.0) THEN
  ! skidding turn logic here
ELSE      ! non-turning flight
  X(4)= PHI
  D = X(2)
  IF(PHI .NE. 0.0) D = -X(2)      ! inverted
  IF( SINGAM .NE. 0.0 ) THEN      ! climbing
    SGOCB = SINGAM / CBETA
    X(5) = D + ATAN( SGOCB/SQRT(1.0-SGOCB*SGOCB)) ! roc constraint
  ELSE
    X(5) = D                      ! level
  END IF
  X(7)= RR
  X(8)= PR
  IF (STAB) THEN                  ! stab.-axis roll
    X(9)= RR*SALPH/CALPH
  ELSE
    X(9) = 0.0                     ! body-axis roll
  END IF
END IF
RETURN
END

```

```

SUBROUTINE SMPLX(FX,N,X,DX,SD,M,Y0,YL)
C This simplex algorithm minimizes FX(X), where X is (Nx1).
C DX contains the initial perturbations in X. SD should be set according
C to the tolerance required; when SD<0 the algorithm calls FX M times
REAL X(*), DX(*)
DIMENSION XX(32), XC(32), Y(33), V(32,32)

NV=N+1
DO 2 I=1,N
DO 1 J=1,NV
V(I,J)=X(I)
V(I,I+1)=X(I)+DX(I)
Y0=FX(X)
Y(1)=Y0
DO 3 J=2,NV
Y(J)=FX(V(1,J))
K=NV
YH=Y(1)
YL=Y(1)
NH=1
NL=1
DO 5 J=2,NV
IF(Y(J).GT.YH) THEN
YH=Y(J)
NH=J
ELSEIF(Y(J).LT.YL) THEN
YL=Y(J)
NL=J
ENDIF
CONTINUE
YB=Y(1)
DO 6 J=2,NV
YB=YB+Y(J)
YB=YB/NV
D=0.0
DO 7 J=1,NV
D=D+(Y(J)-YB)**2
SDA=SQRT(D/NV)
IF((K.GE.M).OR.(SDA.LE.SD)) THEN
SD=SDA
M=K
YL=Y(NL)
DO 8 I=1,N
X(I)=V(I,NL)
RETURN
END IF
DO 10 I=1,N
XC(I)=0.0
DO 9 J=1,NV
IF(J.NE.NH) XC(I)=XC(I)+V(I,J)
XC(I)=XC(I)/N
DO 11 I=1,N
X(I)=XC(I)+XC(I)-V(I,NH)
K=K+1
YR=FX(X)
IF(YR.LT.YL) THEN
DO 12 I=1,N
XX(I)=X(I)+X(I)-XC(I)
K=K+1

```

```

YE=FX(XX)
IF(YE.LT.YR) THEN
  Y(NH)=YE
  DO 13 I=1,N
    V(I,NH)=XX(I)
13 ELSE
  Y(NH)=YR
  DO 14 I=1,N
    V(I,NH)=X(I)
14 END IF
  GOTO 4
ENDIF
Y2=Y(NL)
DO 15 J=1,NV
15 IF((J.NE.NL).AND.(J.NE.NH).AND.(Y(J).GT.Y2)) Y2=Y(J)
  IF(YR.LT.YH) THEN
    Y(NH)=YR
    DO 16 I=1,N
      V(I,NH)=X(I)
16    IF(YR.LT.Y2) GO TO 4
  ENDIF
  DO 17 I=1,N
17 XX(I)=0.5*(V(I,NH)+XC(I))
  K=K+1
  YC=FX(XX)
  IF(YC.LT.YH) THEN
    Y(NH)=YC
    DO 18 I=1,N
      V(I,NH)=XX(I)
18  ELSE
    DO 20 J=1,NV
    DO 19 I=1,N
19  V(I,J)=0.5*(V(I,J)+V(I,NL))
20  IF(J.NE.NL) Y(J)=FX(V(I,J))
  K=K+N
ENDIF
GO TO 4
END

```

B2. NUMERICAL LINEARIZATION SUBROUTINE

Subroutine JACOB will calculate Jacobian matrices for the set of nonlinear state equations contained in the subroutine F (specified as an argument of JACOB). Subroutine F(TIME, X, XD) should contain "CONTROLS" and "OUTPUT" common blocks as used in the text. The argument FN is a double-precision function used to determine an approximation to each partial derivative that is required.

To calculate the A, B, C, D matrices the main program should be designed to call JACOB four times, with FN replaced in turn by each of the partial derivative functions FDX, FDU, YDX, and YDU (given below). The partial derivative functions must be declared "EXTERNAL" in the main program. The vectors X and XD are respectively the state vector and its derivative.

The vector V must contain the equilibrium condition and should be replaced by X or U respectively, depending on whether the partial derivatives with respect to X or U are being calculated. The array IO is used to specify the set of integers corresponding to the rows of the Jacobian matrix, and JO is used to specify the set corresponding to the columns. NR and NC are respectively the number of rows and the number of columns in the Jacobian matrix and the linear array ABC contains the columns of the Jacobian matrix, stacked one after the other.

The linearization algorithm chooses smaller and smaller perturbations in the independent variable and compares three successive approximations to the particular partial derivative. If these approximations agree within a certain tolerance, then the size of the perturbation is reduced to determine if an even smaller tolerance can be satisfied. The algorithm terminates successfully when a tolerance TOLMIN is reached or if a tolerance of at least OKTOL can be achieved. If the algorithm does not terminate successfully, then the successive approximations are displayed and the user is asked to decide on the value of the partial derivative.

```

SUBROUTINE JACOB (FN,F,X,XD,V,IO,JO,ABC,NR,NC)
DIMENSION X(*),XD(*),V(*),IO(*),JO(*),ABC(*)
EXTERNAL FN,F
LOGICAL FLAG, DIAGS
CHARACTER*1 ANS
REAL*8 FN,TDV
DATA DEL,DMIN,TOLMIN,OKTOL /.01, .5, 3.3E-5, 8.1E-4/
C
DIAGS=.TRUE.
PRINT '(1X,A,$)', 'DIAGNOSTICS ? (Y/N, "/"= N) '
READ(*,'(A)') ANS
IF (ANS .EQ. '/') .OR. ANS .EQ. 'N' .OR. ANS .EQ. 'n')DIAGS=.FALSE.
IJ= 1
DO 40 J=1,NC
DV= AMAX1( ABS( DEL*V(JO(J)) ), DMIN )
DO 40 I=1,NR
FLAG=.FALSE.
1   TOL= 0.1
OLTOL= TOL
TDV= DBLE( DV )
A2= 0.0
A1= 0.0
A0= 0.0
B1= 0.0
B0= 0.0
D1= 0.0
D0= 0.0
IF (DIAGS .OR. FLAG) WRITE(*,'(/1X,A8,I2,A1,I2,11X,A12,8X,A5)')
& 'Element ',I,',',J, 'perturbation','slope'
& 'Element ',I,',',J, 'slope'
DO 20 K= 1,18           ! iterations on TDV
A2= A1
A1= A0
B1= B0
D1= D0
A0= FN(F,XD,X,IO(I),JO(J),TDV)
B0= AMIN1( ABS(A0), ABS(A1) )

```

```

DO= ABS ( A0 - A1 )
IF (DIAGS .OR. FLAG) WRITE(*,'(20X,1P2E17.6)') TDV,A0
IF(K .LE. 2) GO TO 20
IF (A0 .EQ. A1 .AND. A1 .EQ. A2) THEN
  ANS= A1
  GO TO 30
END IF
IF (A0 .EQ. 0.0) GO TO 25
10 IF( DO .LE. TOL*B0 .AND. D1 .LE. TOL*B1) THEN
  ANS= A1
  OLTL= TOL
  IF(DIAGS .OR. FLAG) WRITE(*,'(1X,A9,F8.7)') 'MET TOL= ',TOL
  IF (TOL .LE. TOLMIN) THEN
    GO TO 30
  ELSE
    TOL= 0.2*TOL
    GO TO 10
  END IF
END IF
20 TDV= 0.6DO*TDV
25 IF (OLTL .LE. OKTOL) THEN
  GO TO 30
ELSE IF (.NOT. FLAG) THEN
  WRITE(*,'(/1X,A)') 'NO CONVERGENCE *****'
  FLAG= .TRUE.
  GO TO 1
ELSE
 21 WRITE(*,'(1X,A,$)') 'Enter estimate : '
  READ(*,*ERR=21) ANS
  FLAG= .FALSE.
  GO TO 30
END IF
30 ABC(IJ)= ANS
IF (DIAGS) THEN
  PRINT '(27X,A5,1PE13.6)', 'Ans= ',ANS
  PAUSE 'Press "enter"'
END IF
40 IJ= IJ+1
RETURN
END

DOUBLE PRECISION FUNCTION FDX(F,XD,X,I,J,DDX)
REAL*4 XD(*), X(*)
DOUBLE PRECISION T, DDX, XD1, XD2
EXTERNAL F
TIME= 0.0
T = DBLE( X(J) )
X(J)= SNGL( T - DDX )
CALL F(TIME,X,XD)
XD1 = DBLE( XD(I) )
X(J)= SNGL( T + DDX )
CALL F(TIME,X,XD)
XD2 = DBLE( XD(I) )
FDX = (XD2-XD1)/(DDX+DDX)
X(J)= SNGL( T )
RETURN
END

```

```
DOUBLE PRECISION FUNCTION FDU(F,XD,X,I,J,DDU)
PARAMETER (NIN=10)
REAL*4 XD(*), X(*)
COMMON/CONTROLS/U(NIN)
DOUBLE PRECISION T, DDU, XD1, XD2
EXTERNAL F
TIME= 0.0
T = DBLE( U(J) )
U(J)= SNGL( T - DDU )
CALL F(TIME,X,XD)
XD1 = DBLE( XD(I) )
U(J)= SNGL( T + DDU )
CALL F(TIME,X,XD)
XD2 = DBLE( XD(I) )
FDU = (XD2-XD1)/(DDU+DDU)
U(J)= SNGL( T )
RETURN
END
```

```
DOUBLE PRECISION FUNCTION YDX(F,XD,X,I,J,DDX)
PARAMETER (NOP=20)
REAL*4 XD(*), X(*)
COMMON/OUTPUT/Y(NOP)
DOUBLE PRECISION T, DDX, YD1, YD2
EXTERNAL F
TIME= 0.0
T = DBLE( X(J) )
X(J)= SNGL( T - DDX )
CALL F(TIME,X,XD)
YD1 = DBLE( Y(I) )
X(J)= SNGL( T + DDX )
CALL F(TIME,X,XD)
YD2 = DBLE( Y(I) )
YDX = (YD2-YD1)/(DDX+DDX)
X(J)= SNGL(T)
RETURN
END
```

```
DOUBLE PRECISION FUNCTION YDU(F,XD,X,I,J,DDU)
PARAMETER (NIN=10, NOP=20)
REAL*4 XD(*), X(*)
COMMON/CONTROLS/U(NIN)
COMMON/OUTPUT/Y(NOP)
DOUBLE PRECISION T, DDU, YD1, YD2
EXTERNAL F
TIME= 0.0
T = DBLE( U(J) )
U(J)= SNGL( T - DDU )
CALL F(TIME,X,XD)
YD1 = DBLE( Y(I) )
U(J)= SNGL( T + DDU )
CALL F(TIME,X,XD)
YD2 = DBLE( Y(I) )
YDU = (YD2-YD1)/(DDU+DDU)
U(J)= SNGL(T)
RETURN
END
```

B3. RUNGE – KUTTA INTEGRATION

This subroutine implements “Runge’s fourth-order rule” as described in Chapter 3. Its arguments are the subroutine F containing the nonlinear state equations, the current time TT, the integration time-step DT, the state and state-derivative vectors XX and XD, and the number of state variables NX. Subroutine F should be declared EXTERNAL in the main program unit.

```

SUBROUTINE RK4(F,TT,DT,XX,XD,NX)
PARAMETER (NN=30) ! same as main prog.
REAL*4 XX(*),XD(*),X(NN),XA(NN)
CALL F (TT,XX,XD)
DO 1 M=1,NX
1   XA(M)=XD(M)*DT
    X(M)=XX(M)+0.5*XA(M)
    T=TT+0.5*DT
    CALL F (T,X,XD)
    DO 2 M=1,NX
      Q=XD(M)*DT
      X(M)=XX(M)+0.5*Q
2    XA(M)=XA(M)+Q+Q
      CALL F (T,X,XD)
      DO 3 M=1,NX
        Q=XD(M)*DT
        X(M)=XX(M)+Q
3    XA(M)=XA(M)+Q+Q
      TT=TT+DT
      CALL F (TT,X,XD)
      DO 4 M=1,NX
        XX(M)=XX(M)+(XA(M)+XD(M)*DT)/6.0
4    RETURN
END

```

B4. ADAMS – BASHFORTH INTEGRATION SUBROUTINE

The subroutine “ABM” (below) implements the closed, third-order, Adams–Bashforth formula

$$x_{n+1} = x_n + (T/12) * (5\dot{x}_{n+1} + 8\dot{x}_n - \dot{x}_{n-1}).$$

This integration formula is not self-starting and so lower-order formulae have been used to get the integration started. The subroutine arguments are the input vector X, the output vector Y, the time-step T, the dimension N of X and Y, and a logical flag. FLAG should be set to TRUE to initialize the Y vector when the subroutine is first called.

```

SUBROUTINE ABM(X,Y,T,N,FLAG)
PARAMETER (NN=3)
DIMENSION BUF(NN,0:2),X(*),Y(*),YOLD(NN)
LOGICAL START,FIRST,SECOND,FLAG
DATA IB/0/
C
  IF (FLAG) THEN
    START=.TRUE.
    FIRST=.TRUE.
    SECOND=.FALSE.
  END IF
  DO 10 I=1,N
10  BUF(I,IB)=X(I)
  IF (START) THEN
    IF(FIRST) THEN
      IB= MOD((IB+1),3)
      DO 15 I=1,N
15    YOLD(I)=Y(I)
      FIRST=.FALSE.
      SECOND=.TRUE.
      RETURN
    ELSE IF (SECOND) THEN
      IB= MOD((IB+1),3)
      J = MOD((IB+1),3)
      DO 18 I=1,N
18    Y(I)= YOLD(I) + T * ( X(I) +   BUF(I,J) )/2.0
      SECOND=.FALSE.
      RETURN
    ELSE
      IB= MOD((IB+1),3)
      J = MOD((IB+1),3)
      DO 20 I=1,N
20    Y(I)= YOLD(I) + T*(X(I) + 4.0*BUF(I,J) + BUF(I,IB))/3.0
      YOLD(I)= Y(I)
      START=.FALSE.
      RETURN
    END IF
  ELSE
    IB= MOD((IB+1),3)
    J = MOD((IB+1),3)
    DO 30 I=1,N
30    Y(I)= YOLD(I) + T*(5.0*X(I)+8.0*BUF(I,J)-BUF(I,IB))/12.0
    YOLD(I)= Y(I)
  END IF
  RETURN
END

```

B5. QUATERNION UPDATE SUBROUTINE

This subroutine implements the third-order quaternion update algorithm given by McKern (see Chap. 3 references). The subroutine arguments are the vector DTH containing the increments in the three Euler angles ϕ , θ , ψ during one sample period, and the quaternion vector S which is updated for that sample period. If used with rate gyros, rather than rate-integrating gyros, then simple trapezoidal integration can be incorporated to calculate the Euler angle increments from the angular rates (P, Q, R) and the time step. The entry point QUATINI is used to initialize the quaternion or, alternatively, the initialization scheme indicated in Example 3.7-2 can be used.

```

SUBROUTINE QUAT(DTH,S)
DIMENSION DTH(3), DTHP(3), S(4), SP(4), S0(4)
C
AL= DTH(1)*DTH(1) + DTH(2)*DTH(2) + DTH(3)*DTH(3)
D1= (AL*DTH(1) + 2.0*(DTH(2)*DTHP(3) - DTH(3)*DTHP(2)))/48.0
D2= (AL*DTH(2) + 2.0*(DTH(3)*DTHP(1) - DTH(1)*DTHP(3)))/48.0
D3= (AL*DTH(3) + 2.0*(DTH(1)*DTHP(2) - DTH(2)*DTHP(1)))/48.0
C
AQ= 1.0 - .125*AL
BQ= -0.5*DTH(1) + D1
CQ= -0.5*DTH(2) + D2
DQ= -0.5*DTH(3) + D3
C
S(1)= AQ*SP(1) + BQ*SP(2) + CQ*SP(3) + DQ*SP(4)
S(2)= -BQ*SP(1) + AQ*SP(2) - DQ*SP(3) + CQ*SP(4)
S(3)= -CQ*SP(1) + DQ*SP(2) + AQ*SP(3) - BQ*SP(4)
S(4)= -DQ*SP(1) - CQ*SP(2) + BQ*SP(3) + AQ*SP(4)
C
DO 10 I=1,3
10 DTHP(I)= DTH(I)
SP(I)= S(I)
SP(4)= S(4)
RETURN
C
ENTRY QUATINI (S0)
DO 1 I=1,3
1 DTHP(I)= 0.0
SP(I)= S0(I)
SP(4)= S0(4)
RETURN
END

```

B6. DISCRETE-TIME TRANSITION MATRIX

The subroutine EXPAT, given below, calculates the matrix exponential functions from the series definitions given by (3.7-2) and (3.7-4). The subroutine inputs are the time step T, the A matrix and its dimension N, and the minimum number of terms, MIN, to be used in the series approximation. The outputs are the matrix $P = \exp(AT)$, the matrix Q defined in (3.7-4), and the number of series terms, NCOUNT, that was actually used. When the maximum number of series terms (20 terms) is exceeded, an "alternate return" is executed. This allows control to be returned to the main program at the label number specified after the asterisk in the subroutine call. This simple series approximation algorithm is good enough for many purposes. The time step should not normally be so large that the series fails to converge to the built-in tolerance (1.0D-8) with ten or fewer terms. The convergence criterion will fail to work with some pathological A matrices (e.g., a *nilpotent* matrix) and too few terms may be used. More sophisticated algorithms are available with modern software packages (MATLAB, MATRIX_X, CONTROL-C, etc.). Subroutine EXPAT also requires the function ANORM which is used to calculate matrix row norms.

```

C  FUNCTION TO CALCULATE MATRIX ROW NORMS
      REAL*8 FUNCTION ANORM (ARRAY, N)
      REAL*8 S, ARRAY(N,N)
      ANORM= 0.0
      DO 10 I=1,N
      S=0.0
      DO 20 J=1,N
20    S= S+ DABS(ARRAY(I,J))
10    IF(S .GT. ANORM) ANORM= S
      RETURN
      END

C  SUBROUTINE TO EVALUATE THE MATRIX EXPONENTIALS
C  P=EXP(A*T), Q=(A**-1)*(EXP(A*T)-I)/T
C
      SUBROUTINE EXPAT(T,A,P,Q,N,MIN,NCOUNT,*)
      PARAMETER (MM=10, NN=30)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      REAL*8 AT(NN,NN), A(NN,NN), P(NN,NN), Q(NN,NN), TERM(NN,NN), TEMP(NN)
C
      TOL= 1.D-8
      DO 20 I=1,N
      DO 10 J=1,N
      P(I,J)=0.
      Q(I,J)=0.
      AT(I,J)= T*A(I,J)
10    TERM(I,J)=0.
      P(I,I)=1.0
      Q(I,I)=1.0
20    TERM(I,I)=1.0
      NCOUNT=1
      C

```

```

30    DO 60 I=1,N
      DO 50 J=1,N
      SUM=0.
      DO 40 K=1,N
40    SUM= SUM + TERM(I,K) * AT(K,J)
50    TEMP(J)= SUM
      DO 60 J=1,N
      TERM(I,J)= TEMP(J)
60    P(I,J)= P(I,J) + TERM(I,J)
      TNORM = ANORM (TERM,N)
      NCOUNT= NCOUNT + 1
      DO 70 I=1,N
      DO 70 J=1,N
      TERM(I,J)= TERM(I,J)/FLOAT(NCOUNT)
70    Q(I,J)= Q(I,J) + TERM(I,J)
      IF (TNORM/ANORM(P,N) .LT. TOL .AND. NCOUNT .GE. MIN) THEN
        DO 80 I= 1,N
        DO 80 J= 1,N
        Q(I,J)= T*Q(I,J)
        RETURN
      ELSE IF (NCOUNT.GE.20) THEN
        RETURN1
      ELSE
        GO TO 30
      END IF
      END

```

B7. OUTPUT-FEEDBACK DESIGN

Output-feedback design is not an easy problem. Finding the optimal output-feedback gains to minimize a quadratic performance index (PI)

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt, \quad (\text{B.7.1})$$

involves solving coupled nonlinear matrix design equations of the form (Chapter 5)

$$0 = \frac{\partial H}{\partial S} = A_c^T P + P A_c + C^T K^T R K C + Q \quad (\text{B.7.2})$$

$$0 = \frac{\partial H}{\partial P} = A_c S + S A_c^T + X \quad (\text{B.7.3})$$

$$0 = \frac{1}{2} \frac{\partial H}{\partial K} = R K C S C^T - B^T P S C^T. \quad (\text{B.7.4})$$

where

$$A_c = A - B K C, \quad X = x(0) x^T(0).$$

In the design of tracking systems, the equations are even worse.

We have used two general approaches to solving such equation sets. In the first, the PI J is computed based on (B.7.2) using

$$J = \frac{1}{2} \operatorname{tr}(PX). \quad (\text{B.7.5})$$

The simplex routine in Appendix B1 was used to minimize J . In the second approach, a gradient-algorithm (e.g. Davidon–Fletcher–Powell) was used.* There, the gradient $\partial J / \partial K$ is computed using all three design equations (B.7.2)–(B.7.4).

*Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*, New York: Cambridge, 1986.

INDEX

Acceleration:

- lateral, 274–279,
- normal, 123, 155, 228, 254, 263–270, 386, 401

Accelerometer:

- equations of, 25–27
- feedback from, 263–270
- position of, 263–265

Ackermann's formula, 488

Actuator:

- hydraulic, 120–123
- limiting/saturation, 325–332, 573

Adams–Bashforth integration, 144–145

Additive uncertainty, 470

Aerodynamic angles (α , β):

- defined, 61
- equations for, 64

Aerodynamic center, 53, 58, 75

Aerodynamic data:

- plots, 79, 272–273
- tables, 78–80

Aerodynamic derivatives, *see* Derivatives

Aerodynamic forces and moments:

- coefficients for, 55–56, 65–80
- component buildup, 67
- defined, 65
- rate-dependent, 66

Aileron:

- doublet response, 253
- effectiveness, 73, 271
- rudder interconnect, 186, 274–285

Air-data:

- computer, 123, 238
- sensors, 200

Aircraft:

- altitude-speed envelope, 201
- electromechanical controls, 202
- F-16 computer model, 124
- planforms, 58–61
- transport aircraft model, 123

Airfoil properties, 52–58

Aliasing, 558

Alpha-dot effects, 66, 76, 82

Altitude:

- feedback, 293
- hold autopilot, 292–298
- pole, 173, 175
- state, 155, 175

Analog computer simulation, 119

Angle of attack:

- definition, 53, 61, 64
- estimation of, 502
- feedback, 237–244
- sensor, 238
- trim value, 239

Angular-acceleration state equation, 30
 Angular momentum:
 rate of change, 28–29
 of spinning rotors, 32–33
 Angular velocity vector, 7, 17, 21, 38
 Anhedral angle, 71
 Antwindup compensation, 574
 Artificial feel, 203, 231
 Aspect ratio, 58–59
 Atmosphere:
 mathematical model, Appendix A
 standard, 123
 Attitude:
 -*hold* autopilot, *see* Autopilots
 pitch, 36
 Attitude representation, 35–44
 Augmentation:
 control, 254–285
 stability, 236–254
 Autocorrelation, 495
 Autopilots:
 altitude hold, 292–298
 automatic landing, 298–316, 410
 bank-angle hold, 316–318, 406
 bank-angle-steering, 319–324
 flare control, 309–316, 429
 general description of, 204, 285
 heading-hold, 324
 Mach hold, 292
 navigational modes, 324
 pitch attitude hold, 285–292
 terrain-following, 325
 turn coordination, 319, 521
 VOR-hold, 324
 Axes (aircraft):
 body-fixed, 62
 definitions, 62
 stability, 62
 wind, 62, 84
 Bank-angle:
 -*hold* autopilot, 316–318, 406
 -*steering* autopilot, 319–324
 Bending-mode filter, 246
 Beta-dot:
 effects, 66, 82
 feedback, 276
 Bilinear transformation, 547

Bode plots:
 for classical design, 177, 290, 296,
 303–304
 multivariable, 458
 Bryan, G. H., 89, 197
 Ceiling, service, 201
 Center of gravity:
 as coordinate origin, 3
 effects of shifting, 73–75, 139, 211–213
 Center of pressure, 53
 Center of rotation, 264
 Centripetal acceleration, 8, 24, 26, 131
 Characteristic polynomial, 12, 180, 206,
 211, 215
 Chord:
 definition, 53
 mean geometric, 58, 65
 Command generator tracker, 423
 Command limiters, 326
 Companion form matrix, 119
 Compensation:
 networks, 116
 phase-lead/phase-lag, 182, 291,
 294–295, 302–305, 310
 proportional plus integral, 255, 266,
 294, 303, 551
 Compensator with desired structure, 378
 Compressibility, 57, 67 ff., 221
 Computation delay, 562
 Constrained feedback design, 393
 Control augmentation systems (CAS):
 description of, 203, 254
 lateral-directional, 270–285, 406
 normal-acceleration, 263–270, 386, 401
 pitch-rate, 255–262, 402, 480, 553, 567
 stability-axis-roll, 270
 Control mixer, 186
 Control surface:
 effectiveness, 73, 77, 271
 sign convention, 127
 Control system:
 electromechanical, 202
 feel, 203, 221
 irreversible, 164, 202
 model following, 314
 multi-input/multi-output (MIMO), 185
 single-input/single-output (SISO), 179
 Type, 180

- Controllability, 343, 487
 Convolution integral, 158
 Coordinated turn, *see* Turn
 Coordinate frames:
 aircraft-body (ABC), 18
 Earth-centered inertial (ECI), 19
 north-east-down (NED), 19
 Coordinate rotations, *see* Rotation matrix
 Coriolis, theorem of, 17
 Cost function, 133–135, 226
 Coupling:
 inertia, 32, 270
 numerators, 187
 Cramer's rule, 205
 Cross-correlation, 494
 Crossover frequency, 181
 Cross product:
 definition, 6
 matrix, 7
 C-star:
 criterion, 228
 response, 268
 Damping:
 aerodynamic, 66–67
 derivatives, 66
 dutch-roll, 215
 phugoid, 209
 short-period, 207
 Data tables (aerodynamic), 78–80
 DC gain, 468
 Deadbeat observer, 579
 Decoupling:
 in aircraft dynamics, 88, 101, 345, 357
 in LQR design, 410
 Derivatives:
 dimensionless, 103 ff.
 force dimensional, 94
 moment dimensional, 97
 weighting in LQR design, 394, 405
 Detectability, 366, 441
 Deviation system, 381
 Decst filter, 502
 Difference equations, 148
 Differentiation:
 of cross-product, 18
 of vectors, 15
 Digital control:
 realization structures, 577
 techniques, 341, 543
 Dihedral:
 angle, 71
 effect, 70, 108, 214
 Dimensional derivative, *see* Derivatives
 Dimensionless derivative, *see* Derivatives
 Discretization of continuous controller, 546
 Disturbance, 453
 Dot product, 5
 Downwash at aircraft tail, 76
 Drag:
 coefficient, 53, 56, 65, 68

 induced, 59
 polar, 68
 D-star-criterion, 229
 Dutch-roll:
 approximation, 214
 dependence on stability derivatives, 109
 eigenvectors, 165
 frequency and damping formulae, 216
 -mode described, 166
 Dynamic pressure:
 definition, 55
 variation over envelope, 201
 Earth:
 angular rotation rate of, 24
 centered inertial reference frame, 19
 equatorial radius of, 23
 flattening parameter, 33
 mass gravitational constant, 22
 sea-level radius, 34
 shape of, 33
 Eigenvalues:
 definition, 11
 of F-16 model, 164–165
 of rotation matrix, 14
 Eigenvectors, 162
 assignment of, 339, 347
 of lateral dynamics, 347
 short period, 165, 346, 354
 Elevator:
 effectiveness, 77
 pulse simulation, 150

- Envelope, *see* Aircraft, altitude-speed envelope
- Equilibrium, stable/unstable, 54
- Equilibrium point, 90
- Error covariance, 497
- Estimate of state vector, 484
- Euler angles:
- definition, 37
 - differential equation for, 38
 - from rotation matrix, 39
 - rotation sequence, 36
- Euler integration, 141
- Euler's theorem, 14, 40
- Exogenous input, 415
- Expected value, 493
- F-16 aircraft model, 124
- Feedback:
- of angle of attack, 237–244
 - control, 179 ff.
 - output, 343
 - state, 342
- Flaps, 59, 298
- Flare:
- controller, 309
 - LQR design, 429
- Flat-Earth equations, *see* State equations
- Flexibility, 3, 65, 246, 480, 525
- Flight-path angle, 88, 130
- Flying qualities:
- Cooper–Harper rating of, 223
 - definitions, 233
 - equivalent system specification, 226
 - frequency response specifications for, 227
 - human operator based, 229
 - military specifications for, 223, 232
 - pole-zero specifications for, 224
 - requirements, 222
 - time-response specifications for, 227
- Free fall, 25
- Frequency response:
- definition of, 175
 - elevator to pitch-rate, 177
 - from poles and zeros, 176
 - short-period approximation, 177
- Gain:
- margin, 181–182
 - scheduling, 186, 200, 202, 261, 376, 449, 452, 478, 539
- Gaussian pdf, 493
- g-command system:
- classical design of, 254
 - LQR design of, 386, 401
- Glide path:
- controller, 298
 - coupler, LQR design, 410
- Goldstein, H., 14
- Gravitational attraction, 22–24
- Gravity vector, 27
- Ground effect, 65, 69, 316
- Gyros, *see* Rate gyro
- Gyroscopic coupling, *see* Inertia coupling
- Gyroscopic equations, 32
- Hamiltonian, 363, 384, 396, 500
- Harmonic oscillator, 451
- H-infinity norm, 465
- Homogeneous solution, 15, 158
- Householder transformation, 171
- Human-operator models:
- optimal control model, 230
 - transfer function model, 229
- Hydraulic actuator, *see* Actuator
- Hypersonic aircraft, 199, 200
- Implicit model-following, 314
- Implicit state equations, *see* State equations
- Inertia coupling, 32, 197, 270
- Inertia matrix:
- definition, 30
 - inverse of, 31
 - transformation to wind/stability axes, 86
- Inertial navigation, 17, 160–162
- Inertial position vector, 20, 35
- Inertial reference frame, 19
- Initial conditions for LQR problem, 364, 439
- Initial stabilizing gain, 367, 393
- Integration, *see* Numerical integration

- Jacobian matrix:
 defined, 93
 numerical examples, 156–157,
 164–165, 174, 238, 247, 277, 286, 289,
 300
- Jordan-form matrix, 13, 162
- Kalman, R.E.:
 definition of state, 2
 filter, 341, 492
 gain, 438, 501
- Kinematic:
 coupling, 271
 equations, 33 ff.
 linearized equations, 96
- Lagrange multiplier, 363
- Lanchester, F. W., 210
- Laplace transform:
 initial/final value theorems, 171
 of state equations, 166
- Lateral:
 acceleration, 274–279
 -dynamics Bode plot, 462
 eigenvectors, 347
 regulator, LQR design, 370, 399
- Latitude:
 geocentric, 23, 34
 geodetic, 34
- Leverrier's algorithm, 169
- Lift:
 coefficient, 55, 65, 69
 curve, 56, 69
 -curve slope, 57, 107
 definition, 53
- Limiters, 325–332
- Linearization:
 algebraic, 92 ff.
 algebraic versus numerical, 157
 numerical, 152–155
- Linear quadratic regulator (LQR):
 loop transfer recovery (LTR), 510
 with output feedback, 364
 with state feedback, 437
- Load factor, 225
- Longitude, 33–35
- Loop gain, multivariable, 458
- Lyapunov equation, 362, 396
- Mach number:
 critical Mach number, 58
 defined, 57
 drag-divergence Mach number, 58
 -hold autopilot, 293
- Matched pole-zero (MPZ) discretization, 550
- Matrix inversion lemma, 170
- McKern, R. A., 160
- Military flying-qualities, 232
- MIMO, *see* Transfer function
- Minimality, 345
- Minimization:
 gradient method, 365, 384
 of performance index, 360
 simplex, 133, 365, 384
- Modal coordinates, 13, 162
- Modal decomposition, 162–164, 344
- Model following:
 control, 340
 explicit, 421, 432
 implicit, 314, 369, 427
- Model reduction, 472
- Modes, *see* Natural modes
- Multi-loop and MIMO systems, 184–189
- Multiplicative uncertainty, 471
- Natural modes:
 algebraic derivation, 205
 of business jet, 218
 definition, 15
 effect of flight conditions, 204, 221
 of F-16 model, 151, 164–166, 220
 in multivariable design, 339
 stick-fixed, 164
 third oscillatory mode, 211
- Neal-Smith criterion, 230
- Neutral point, 75
- Newton's laws applied to aircraft motion, 19, 28
- Newton's system:
 LQR design, 442
 observer, 489
- Noise:
 measurement, 455, 493
 nonwhite, 496
 nonzero mean, 540
 process, 492
 white, 495

- Nonlinear simulation, 259–262, *see also* Simulation
 Non-minimum-phase LTR design, 520
 Normal acceleration, *see* Acceleration
 Normal pdf, *see* Gaussian pdf
 North–east–down frame, 19, 27, 36, 46
 Numerical integration, 139–147
Nyquist:
 criterion, 181
 frequency, 548
- Observability:**
 in PI weighting matrices, 368, 397
 properties, 344, 368, 441, 487
- Observer, 341, 484
 Operator gain, L2, 464
 Optimal return difference relation, LQR, 511
Orthogonal:
 matrix, 8
 random processes, 495
- Output:**
 injection, 486
 measured, 378
 performance, 378
- Output feedback:**
 design, 343
 eigenvector assignment, 351
 LQR, 364
 robust, 479
- Padé approximant, 565
Parameter:
 identification, 117
 variation modeling error, 478
 Partial fraction expansion, 345, 472
Performance index, LQR:
 constrained feedback, 393
 derivative weighting, 394, 405
 introduced, 340, 360
 time weighting, 395, 401, 416
 tracker, 382
 weighting matrices, 367, 372, 388, 397, 444
Performance specifications, frequency-domain, 464
Phase lead/lag, *see* Compensation
Phase margin, 181–182
- Phugoid:**
 approximation, 208, 210
 eigenvectors, 165
 frequency and damping, 209
 -mode described, 151
 variation with flight conditions, 209
- Pio, R. L., 10
- PID controller:**
 antiwindup protection, 576
 digital, 551
- Pitch-attitude hold**, 285–292
- Pitch pointing**, 357
- Pitch-rate control system:**
 classical design, 255
 LQR design 402, 480, 553, 567
- Pitch stability**, 54–55, 73–75, 211
- Pitching moment coefficient**, 56, 65, 73
- Pole-placement design**, 339
- Power-boosted controls**, 198
- Power curve:**
 back side of, 137, 298
 for F-16 model, 136
- Prandtl–Glauert correction**, 57
- Principal axes**, 31
- Probability density function (PDF)**, 493
- Probability theory, review**, 492
- Quaternion:**
 definition, 41
 determined from rotation matrix, 42
 differential equation for, 43
 discrete time propagation, 159–160
 from Euler angles, 41
 rotation matrix from quaternion, 41
- Random process**, 494
- Rate gyro:**
 feedback, 237
 location, 237
- Rate of climb**, 131
- Reachability**, *see* Controllability
- Regulator:**
 dynamic, 341, 353, 506
 problem, 339
 reduced-order, 519
- Relaxed static stability**, 74
- Return difference, multivariable**, 458
- Riccati equation, 438, 501
- Robinson, A. C., 40, 42

- Robust design:
 concepts, 341
 with output feedback, 479
- Robustness:
 of LQR, 511
 performance, 453, 466
 to plant parameter variations, 478
 stability, 452, 470
- Roll:
 damper, 248
 stability-axis, 270
- Rolling moment coefficient, 65, 70
- Roll-rate feedback, 244 ff., 270 ff., 317 ff.
- Roll stiffness, *see* Stiffness
- Roll-subsidence:
 effect of flight conditions, 217, 247
 eigenvector, 165
 -mode described, 166
 time constant, 217
- Roll-yaw coupling, 214
- Root locus design, 183, 189, 240, 249,
 257, 267, 278, 287, 318
- Rosenbrock system matrix, 169
- Rotary derivatives, 66–67
- Rotation matrix:
 ABC to wind axes, 63
 definition, 8
 ECI to NED, 36
 from Euler angles, 37
 from quaternion, 41
 NED to ABC, 36
- Rudder:
 control effects, 78
 effectiveness, 271
- Runge–Kutta integration, 140 ff., 545
- Sampling:
 frequency, 547
 period, 560
 theorem, 559
- Sensitivity, multivariable, 455
- Separation principle, 508
- Service ceiling, 201
- Servo design, 340
- Shaping filter, 496, 503
- Short-period:
 approximation 173, 206, 210
 effect of flight conditions, 207
- eigenvectors, 165, 346, 354
 flying-qualities requirements, 234
 frequency and damping, 207
 -mode described, 151
- Sideforce coefficient, 65, 70
- Sideslip:
 adverse, 270
 -angle definition, 62, 64
 -angle feedback, 275
- Simplex algorithm, 133
- Simpson's rule, 144, 159
- Simulation:
 diagram, 119
 digital, 147–148
 numerical examples, 149–152, 253, 258,
 259–262, 268, 279, 288, 292, 297,
 304–309, 312–316, 319–323, 326–332
- Singular points, 90
- Singular value decomposition:
 balancing, 468
 introduced, 459
 LQR constraint, 512
- SISO, *see* Transfer function
- Spectral density matrix, 495
- Speed stability, 232
- Spiral mode:
 approximation, 216
 described, 166
 eigenvector, 165
 stability of, 218
 time constant, 217
 variation with flight conditions, 218
- Spoilers, 73
- Stability:
 axes, 62
 -axis roll, 270
 static/dynamic, 55
 using LQR design, 441
- Stability augmentation systems (SAS):
 description of, 201, 203, 236
 lateral-directional, 244, 370, 399
 pitch axis, 237
- Stability derivatives:
 dimensional, 94
 dimensionless, 103–109
- Stabilizability, 441
- Stagnation point, 52, 56

- State equations:**
 aircraft, linear, 89 ff.
 for aircraft angular motion, 30
 for aircraft translational motion, 22
 body-axes, nonlinear, 81
 decoupled, linear, 101–103
 decoupled, nonlinear, 88–89
 definition, 1–2
 explicit time-domain solution, 158–159
 with feedback, 189
 flat-Earth, 28, 46
 general forms, 1–2, 188
 implicit, 82, 90
 round-Earth, 45
 from transfer functions and ODE's, 118
 wind-axes, nonlinear, 84–88
- State feedback:**
 eigenvector assignment, 347
 introduced, 342
 LQR, 437
- Static loop sensitivity, 171
- Static margin, 75
- Stationary random process, 494
- Steady-state control error, 180, 381, 467
- Steady-state flight:**
 definition of, 90–91
 determination of conditions for, 129–136
 numerical examples of, 134, 136, 137, 139
 trim algorithm, 132
- Stick force per g , 232
- Stiff equations, 145–146
- Stiffness:**
 definition, 55
 pitch, 73–75
 positive, 55
 roll, 72
 yaw, 77
- Strakes, 61
- Strapdown equation, 17–18, 37
- Successive loop closure, 338
- Target feedback loop, LTR, 526
- Taylor series, 141
- Third oscillatory mode, 211, 239
- Throttle:**
 automatic control of, 300
 pulse response simulation, 151
- Thrust:**
 coefficient, 69
 derivative, 212
 variation, 123, 127
- Time-scaling of aircraft equations, 165
- Time weighting in LQ performance index, 395, 401
- Tracker, LQ:**
 introduced, 340
 with output feedback, 376, 385
- Transfer function:**
 algebraic derivation, 204–214
 coupling numerators, 187
 elevator to pitch-rate, 172
 interpretation of, 171–175
 matrix, 166–168
 MIMO, 167, 186
 of networks, 116
 poles and zeros, 168
 relative degree, 168
 SISO, 167
 throttle to speed, 174
- Transformation:**
 congruence, 11
 coordinate, 9
 linear, 10
 similarity, 11
 state-space, 11
- Transition matrix:**
 continuous-time, 158
 discrete-time, 159
- Transport aircraft model, 123
- Trapezoidal integration, 141
- Tuck, 108, 212
- Turn:**
 coordinated, 131–132, 137
 coordination/turn compensation, 275, 319, 522
 simulation of, 149
 steady-state, 91
- Tustin's approximation, 547
- Type, feedback control system, 180
- Unbiased estimate, 498
- Unit vectors, 5–9
- Unmodeled dynamics, 452, 470
- Vortex lift, 60

- Vashout filter, 243, 245, 276
Weathercock stability, 77, 108
Whittaker, E. T., 43
 γ_{ind} :
 gust, 480, 502, 525
 included in state equations, 47
 γ_{ind} axes, *see* Axes
 γ_{indup} , integrator, 573
Wing leveler:
 introduced, 316
LQR design, 406
Wing planform parameters, 59
World geodetic system, 24
-series aircraft, 197–199
- Yaw:
 adverse/proverse, 271
 angle, 36–37
 damper, 248
 stability, 77, 108
Yawing moment coefficient, 65, 77
- Zero-angle root locus, 183
Zero-order hold, 140, 159, 543, 561
Zeros:
 computation of, 168–171
 effect of accelerometer position, 264
 in LQR design, 380, 386, 413
 of MIMO systems, 169
 non-minimum-phase, 172, 174–175,
 248, 264, 278, 520

