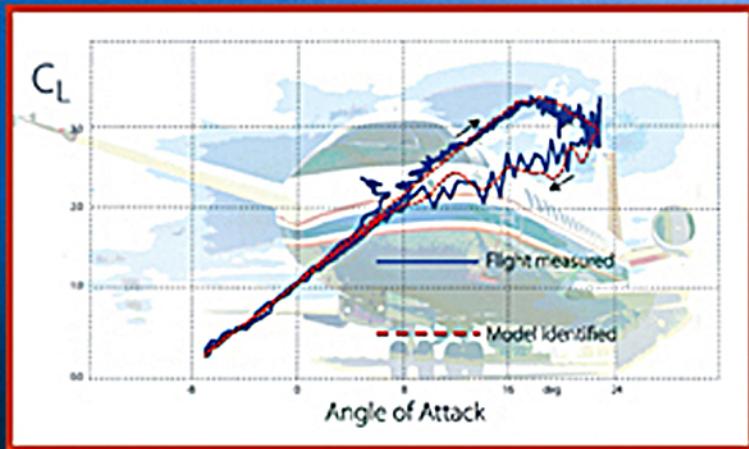


Flight Vehicle System Identification

A Time Domain Methodology

Ravindra V. Jategaonkar



PROGRESS IN ASTRONAUTICS AND AERONAUTICS

Frank K. Lu, Editor-in-Chief
Volume 216

Flight Vehicle System Identification: A Time Domain Methodology

This page intentionally left blank

Flight Vehicle System Identification: A Time Domain Methodology

by

Ravindra V. Jategaonkar

Senior Scientist

Institute of Flight Systems

DLR German Aerospace Center

Braunschweig, Germany

Volume 216
PROGRESS IN
ASTRONAUTICS AND AERONAUTICS

Frank K. Lu, Editor-in-Chief

University of Texas at Arlington
Arlington, Texas

Published by

American Institute of Aeronautics and Astronautics, Inc.
1801 Alexander Bell Drive, Reston, VA 20191

American Institute of Aeronautics and Astronautics, Inc., Reston, Virginia.

Copyright © 2006 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, distributed, or transmitted, in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

For the software, Copyright © 2006 by Ravindra V. Jategaonkar. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

1-56347-836-6

Data and information appearing in this book are for informational purposes only. AIAA is not responsible for any injury or damage resulting from use or reliance, nor does AIAA warrant that use or reliance will be free from privately owned rights.

Progress in Astronautics and Aeronautics

Editor-in-Chief

Frank K. Lu

University of Texas at Arlington

Editorial Board

David A. Bearden

The Aerospace Corporation

Richard C. Lind

University of Florida

John D. Binder

viaSolutions

Richard M. Lloyd

Raytheon Electronics Company

Steven A. Brandt

U.S. Air Force Academy

Ahmed K. Noor

NASA Langley Research Center

Fred R. DeJarnette

North Carolina State University

Albert C. Piccirillo

Institute for Defense Analyses

Philip D. Hattis

Charles Stark Draper Laboratory

Ben T. Zinn

Georgia Institute of Technology

Abdollah Khodadoust

The Boeing Company

Peter H. Zipfel

Air Force Research Laboratory

This page intentionally left blank

Foreword

This book covers a timely subject in the development of flight vehicles in spite of the long history of this topic. For example, through the development of the F-35 Joint Strike Fighter and recent various proposals to incorporate VSTOL in unmanned flight vehicles, the subject addressed in the book continues to be of significant importance, and of current and future interest.

Flight vehicle system identification is an area of aerospace engineering which requires a tight knitting of basic engineering disciplines, experience, wisdom, and intuition. Dr. Jategaonkar provides refreshing insights into flight vehicle system identification and clearly demonstrates its multiple facets by systematically developing the difficult topic. The author pulls together a wealth of knowledge from decades of experience. After an introductory chapter, Dr. Jategaonkar plunges right into the critical aspect of the issues of flight vehicle system identification by discussing flight testing, followed by the development of mathematical tools. Such an arrangement reveals the crucial linkage between analysis and practice. Dr. Jategaonkar then proceeds to introduce advanced topics, including nonlinear stochastic estimation, artificial neural networks and unstable aircraft identification. He rounds off the book, once again, by returning to flight testing for data compatibility and model validation. Finally, Dr. Jategaonkar provides examples that show the application of flight vehicle system identification. The wide selection of examples is useful for illustrating the principles discussed in the book. This volume provides a state-of-the-art overview that will also appeal to experts in the field.

Frank K. Lu
Editor-in-Chief
Progress in Astronautics and Aeronautics

This page intentionally left blank

To Aai and Bapu, my Parents

This page intentionally left blank

Table of Contents

Foreword	vii
Preface	xvii
Chapter 1. Introduction	1
What is System Identification?	2
Model Characterization	5
Interdisciplinary Flight Vehicle Modeling	6
Why System Identification?	8
Parameter Estimation in Flight Mechanics	9
Estimation Techniques of the Past	11
Modern Methods of Aircraft Parameter Estimation	12
General Aspects	14
Chapter 2. Data Gathering	25
Introduction	25
Flight Testing and Maneuvers	26
Optimal Input Design	33
Scope of Flight Testing	46
Flight Test Instrumentation and Measurements	49
Concluding Remarks	54
Chapter 3. Model Postulates and Simulation	59
Introduction	59
Model Description	60
Extensions of the Mathematical Models	61
Retarded Systems	63
Linearized Models	67
Pseudo-control Inputs	69
Treatment of Initial Conditions	71
Simulation	71
Concluding Remarks	76
Chapter 4. Output Error Method	79
Introduction	79
The Principle of Maximum Likelihood Estimation	80
Properties of Maximum Likelihood Estimates	83

TABLE OF CONTENTS

The Maximum Likelihood Function for Estimation of Parameters in Dynamic Systems	84
Basics of Cost Function Optimization	86
Gauss–Newton Algorithm	88
Method of Quasi-linearization	90
System Response and Sensitivity Coefficients	91
Automatic Gradient Computation	94
Step Size Control	95
Bounded-variable Gauss–Newton Method	98
Constrained Gauss–Newton Method Using the Interior-point Algorithm	100
Levenberg–Marquardt Method	103
Direct Search Methods	105
Regression Startup Procedure	107
Estimation Accounting for a Priori Information	109
Statistical Accuracy of Parameter Estimates	111
Algorithmic Implementations	112
OEM Software	115
Examples	119
Concluding Remarks	124
 Chapter 5. Filter Error Method	 131
Introduction	131
Filter Error Method for Linear Systems	133
Process Noise Formulations	135
Filter Error Algorithm	138
Filter Error Method for Nonlinear Systems	145
Initial Noise Covariance Matrix	154
Extension of Filter Error Method to Multiple Experiments	155
Explicit Modeling of Gust Spectrum	157
On the Equivalence of Output Error and Filter Error Methods	159
FEM Software	160
Examples	164
Concluding Remarks	174
 Chapter 6. Equation Error Methods	 177
Introduction	177
Least Squares Method	178
Weighted Least Squares Method	188
Nonlinear and Multi-output Regression	189
Total Least Squares	191
Instrumental Variable Method	194
Data Partitioning	196
Model Structure Determination	197
Examples	204
Concluding Remarks	216

TABLE OF CONTENTS

xiii

Chapter 7. Recursive Parameter Estimation	219
Introduction	219
Least Squares-based Recursive Methods	222
Filtering Methods	234
Algorithmic Implementation and Software	245
Examples	248
Comparative Evaluation of Recursive Algorithms	260
Concluding Remarks	261
Chapter 8. Artificial Neural Networks	265
Introduction	265
Basics of Neural Network Processing	268
Training Algorithms	270
Optimal Tuning Parameters	276
Extraction of Stability and Control Derivatives from Trained FFNN	278
FFNN Software	279
Examples	281
Concluding Remarks	289
Chapter 9. Unstable Aircraft Identification	295
Introduction	295
Basics of Unstable Aircraft Identification	297
Least Squares Method	299
Total Least Squares Method	303
Combined Output Error and Least Squares Approach	304
Equation Decoupling Method	304
Eigenvalue Transformation Method	306
Filter Error Method	308
Extended and Unscented Kalman Filters	309
Output Error Method	310
Output Error Method with Artificial Stabilization	310
Multiple Shooting Method	311
Output Error Method in Frequency Domain	313
Separate Surface Excitation	314
Programming Considerations	316
Examples	317
Concluding Remarks	331
Chapter 10. Data Compatibility Check	335
Introduction	335
Kinematic Equations	336
Flight Path Reconstruction Techniques	344
Estimation-before-modeling Approach	350

TABLE OF CONTENTS

Example	354
Calibration of Five-hole Flow Angle Probe	358
Calibration of Static Pressure Ports	363
Wind-box Maneuver Technique	368
Concluding Remarks	371
Chapter 11. Model Validation	375
Introduction	375
Statistical Accuracy of Parameter Estimates	376
Residual Analysis	378
Inverse Simulation	382
Model Plausibility	383
Model Predictive Capability	386
Range of Model Applicability in Frequency Domain	389
Concluding Remarks	392
Chapter 12. Selected Advanced Examples	395
Introduction	395
Modeling of Transit Time Lag Effects	396
Aerodynamic Effects of Landing Gear	409
Control Surface Malfunction Effects	411
Unsteady Aerodynamics Modeling	414
Quasi-steady Stall Modeling	418
Ground Effect Modeling	427
High-fidelity Databases for Training Simulators	433
X-31A Model Validation and Update	448
Wake Vortex Aircraft Encounter Model	450
Phoenix RLV Demonstrator	455
Rotorcraft Modeling and Simulation	465
Concluding Remarks	479
Epilogue	485
Appendix A. Power Spectrum of a Multistep Input Signal	489
Appendix B. Identifiability of Initial Conditions and Bias Parameters	493
Appendix C. Derivation of the Likelihood Function.	497
Appendix D. Statistical Properties of Maximum Likelihood Estimates.	501
Asymptotic Consistency	501
Asymptotic Normality	503
Asymptotic Efficiency	505

TABLE OF CONTENTS

xv

Appendix E. Minimization of Likelihood Function with Respect to Covariance Matrix R	507
Appendix F. Derivation of Kalman Filter and Extended Kalman Filter	511
Extended Kalman Filter	518
Index	521
Supporting Materials	535

This page intentionally left blank

Preface

THE OBJECTIVE of this book is to provide a consolidated account of flight vehicle system identification that has evolved during recent decades, focusing particularly on nonlinear systems and the time domain approach. It also aims to share the practical experience gained on aerodynamic modeling from flight data to a large number of flight vehicles, because experience and engineering judgment are critical to generate good results. Effective system identification becomes possible based on a coordinated approach and by following certain well-researched guidelines. This book attempts to provide not only details of such a systematic approach, summarizing the general underlying concepts, methodologies, and computational procedures, and present examples of practical applications, but also to show the pitfalls of these methods. It gives practical tips on how to overcome the problems one is likely to face in developing nonlinear, high-fidelity models and analyzing flight data from complex flight vehicles, for example, intermediate divergence of optimization algorithms or estimation subject to bounds, or application of filter error method to data with turbulence, which are generally not covered in theoretical books.

The layout of the book and the material presented here is partly based on the short courses on “Flight Vehicle System Identification in Time Domain” delivered as a part of the AIAA Professional Development program and at other educational and research organizations during the last few years, partly on the personal notes of discussions and experience gathered over two and half decades, and partly on several technical papers published jointly with my colleagues, including several guest scientists who I had the pleasure of guiding. If you locate any errors in the text and software, or have any other comments or questions, please send your suggestions and queries to me (jategaonkar@dlr.de).

With great pleasure, I would like to acknowledge several individuals who helped me during the various phases of writing this book. First, I sincerely thank Professor Peter Hamel for his support and for the use of materials from some of his and our joint papers. My personal discussions with him have shaped the book layout to some extent. Next, I greatly appreciate the help provided by my colleague Wulf Mönnich in the tedious job of reading the draft manuscript and for making many helpful suggestions. It has helped me directly and indirectly in the thought processes reflected in the book. I would also like to acknowledge my former colleague Dr. Ermin Plaetschke for his comments on the draft version, for the use of material from our joint papers and other notes. I am thankful to my colleague Dietrich Fischenberg for providing the case study on modeling of wake vortex encounter and some material on other examples. Likewise, I extend my thanks to Dr. Wilhelm Gockel for the consent to report on the case study pertaining to reusable launch vehicle demonstrator. Help from Dr. Wolfgang von Grünhagen on rotorcraft example is appreciated. I am also thankful to Professor Stefan Levedag and Dr. Frank

Thielecke for the facilities granted at the Institute. The interest of other present and former colleagues of the Institute is appreciated, particularly that of Dr. Karl Doherr. It has been a pleasure to work at the DLR Institute of Flight Systems for more than two decades. I also recall my past association with the former Systems Engineering Department of the National Aeronautical Laboratory, Bangalore, India.

I would also like to extend my special appreciation to my wife, Padma, for her patience throughout the extended period of writing this book. Without her understanding and support, this book would not have been possible. I would also like to mention here our daughters, Smita and Swati, for their continued interest.

Finally, I would like to acknowledge the interest of AIAA in publishing this book under the Progress Series. In particular, I would like to extend my appreciation to Dr. Peter Zipfel, member of the Editorial Board, for a discussion which led to embarking on this book project. I would like to acknowledge Rodger Williams, AIAA Publications Development for his help and encouragement. I extend my thanks to Alex McCray, Managing Editor, Books, AIAA, for overseeing the book production, to Janice Saylor, Marketing Strategist, AIAA, for the front and back cover design and marketing, and Nick Barber, Books Manager, Techset Composition Ltd., for copyediting, text composition and for incorporating text amendments efficiently. I am sure many other AIAA staff are involved down the line, whom I would like to thank as well.

Ravindra V. Jategaonkar
DLR Institute of Flight Systems
December 2005

Chapter 1

Introduction

SYSTEM IDENTIFICATION, as it is termed today, is a scientific discipline that provides answers to the age-old inverse problem of obtaining a description in some suitable form for a system, given its behavior as a set of observations. The inverse problem and, hence, system identification has been a fundamental part of obtaining knowledge of any physical system that is observed, and as such has far ranging areas of applications. It enables us not only to know more about the principle underlying (i.e., model formulation of) the process being observed, but also to determine it in adequate detail (i.e., analysis and parameter estimation). System identification is implicitly concerned with modeling from experimental data, and covers applications in all possible areas: biology, medicine, chemical processes, economics, geology, materials, civil and mechanical engineering, automobiles, and, of course, flight vehicles, to name just a few. In this book we are concerned with the last-mentioned subject, flight vehicle system identification. As a rule, a flight vehicle is a dynamic system, and since a state space representation in the time-domain is closer to physical reality than any of the frequency-domain transform techniques, we deal here mainly with the time domain methodology.

The real processes are, in general, too complex and the exact internal behavior is unknown; the mathematical model, no matter how complex, at best represents a replica of what we purport to underlie the system behavior. It may be possible to postulate a very comprehensive model encompassing all conceivable influences or based on theoretical formulations. A large model, however, does not necessarily mean a good system representation. The principle of parsimony, also called the principle of simplicity or Ockham's Razor, named after an early fourteenth-century English theologian and philosopher, needs to be applied to determine the "best" model.^{1,2} Translated from Latin, in essence, it implies:

"The number of entities should not be increased beyond what is necessary to explain anything."

In our case entities refer to the assumptions or parameters appearing in the mathematical model. Speaking in layman's terms this implies *as simple as possible, and as complex as necessary*. It is a methodological principle which helps to minimize redundancies and inconsistencies in the model. Thus, the best model representation having fidelity within a specified tolerance is that

which has the minimum number of parameters. Simpler models are easier to interpret and are necessary for practical reasons of testing, estimation and cross-validation. The definition of the “best” model is, however, not unique and depends strongly on the intended purpose of the derived model. Here, engineering knowledge and experience play important roles.

The process of system identification involves certain fundamental assumptions:

- 1) The true state of the dynamic system is deterministic.
- 2) Physical principles underlying the dynamic process can be modeled.
- 3) It is possible to carry out specific experiments.
- 4) Measurements of system inputs and outputs are available.

The first and the second assumptions are somewhat related to each other. They suggest that we deal with systems that can be described mathematically, and not with chaotic systems whose input–output behavior cannot be described through fixed rules. It is also implied that, under ideal conditions, data analysis from repeated experiments converges to the truth, i.e., lead to the same model. The third assumption implies that specific experiments will be necessary to excite different modes of dynamic motion which may call for suitable experiments to be designed. This may be possible in some cases, for example in the case of flight vehicles, and not be in others (such as economic systems). The fourth assumption, as already pointed out, is implicit to system identification. Either directly measured or derived quantities (for example, engine thrust computed from other measured variables) must be available.

I. What is System Identification?

The subject of system identification is closely related to the other problems in dynamics. For a dynamic system described in state space, three quantities that mainly describe the system are the inputs u , the outputs z or y , and the system model functions f and g , see Fig. 1.1. Depending upon the availability of any two of these, three different types of problems are encountered in system theory:

- 1) Simulation deals with finding the outputs y for given inputs u and given system functions f and g .

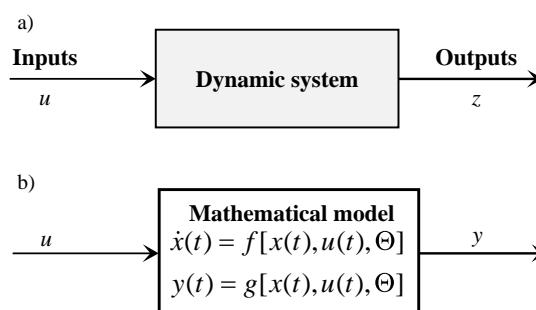


Fig. 1.1 Representation of dynamic system. a) Process under investigation. b) Model representation in time domain.

INTRODUCTION

3

- 2) The control problem aims to find the control inputs u for given f and g , and y .
- 3) The identification problem attempts to find the system model f and g from the given measurements of inputs u and outputs z .

Although the state variables x are not mentioned explicitly, they appear in the above description implicitly, because the outputs are functions of internal system state variables. Likewise, the system functions f and g contain unknown parameters Θ .

System identification is the process of determining an adequate mathematical model, usually containing differential equations, with unknown parameters which have to be determined indirectly from measured data. The process includes not only the model postulating and determining parameters that is our ultimate goal, but also performing suitable experiments, and gathering system inputs and responses. From the modeling aspect, this amounts to determining the equations and unknown model parameters Θ such that the model response y matches adequately the measured system response z (Fig. 1.1). By saying ‘adequately’, we introduce a notion that perfect fit may not be possible, which happens to be the case when dealing with real processes. The technical definition of system identification is provided by Zadeh:³

“System identification is the determination, on the basis of observation of input and output, of a system within a specified class of systems to which the system under test is equivalent.”

Several critical aspects of system identification are encompassed by this definition: availability of input and output (i.e., of experimental) data, postulating probable models (i.e., class of systems), and choosing the best model from this specified class (i.e., equivalency). Thus, the outcome of system identification will be heavily influenced by the models being considered in the process. It also implies that the mathematical model for the physical process may not be unique. In contrast to the above technical definition, Iliff provides a more philosophical definition:⁴

“Given the answer, what are the questions, i.e., look at the results and try to figure out what situation caused those results.”

This brings out the central approach behind the process of model building, namely that system identification constitutes an inverse problem.

System identification is mainly concerned with the determination of the mathematical model structure representing the dynamic system, which is in general unknown and not unique. As already mentioned, different model postulates may have to be investigated. For a specified model, the parameters within the particular model are quantified by applying a numerical, usually statistical, procedure. This part of the overall model building process, which happens to be the core activity, is called “parameter estimation” (Fig. 1.2). Parameter estimation needs to be followed by a step called “model validation” to assess model fidelity. If it turns out that the identified model does not meet the requirements, the model structure has to be changed and the whole process

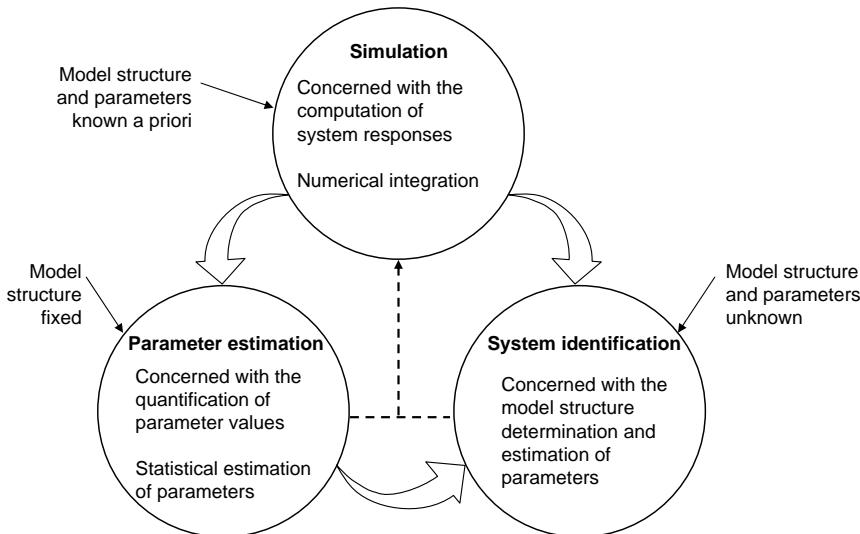


Fig. 1.2 Definitions of system identification, parameter estimation, and simulation.

repeated. It may happen that additional model structures or even new experimental data may be necessary to arrive at a final model with sufficient fidelity. Thus, system identification is, in general, an iterative process, calling for substantial effort. The amount of effort spent on modeling depends on the required fidelity and end use of the identified model.

If it happens that a structure is known from some a priori knowledge, or happens to be pre-defined (for example analysis of data generated through numerical simulation), then we deal with only part of the system identification, called “parameter estimation” to determine the numerical values of parameters (Fig. 1.2). In the literature this step of quantifying numerical values is loosely referred to as “parameter identification” (PID). In our opinion, it is more appropriate to refer to it as “parameter estimation,” because it is a quantification procedure. Where it is necessary to change the model structure, we then deal with “system identification,” which automatically encompasses “parameter estimation.”

As already pointed out, parameter estimates are determined by matching computed system responses y with the measured system responses z . Thus, having postulated a model, we need a numerical procedure, called simulation, to compute the system responses y for given inputs u . In this task the model structure as well as the numerical values of the parameters are known a priori and held fixed (Fig. 1.2). Thus, it is now apparent that system identification, parameter estimation, and simulation are three interrelated aspects. From the specific aspect of flight vehicle modeling from flight data, simulation and parameter estimation are integral parts of the overall system identification. Simulations are also widely used in the other disciplines related to flight vehicles, namely

INTRODUCTION

5

computational fluid dynamics and structural mechanics. Their scope, complexity and the purpose, as well as the associated model uncertainty, varies significantly compared with flight mechanical simulations. We will address this particular aspect in a little more detail in Sec. III.

II. Model Characterization

A suitable formulation of a model for the process being investigated is an aspect of paramount importance in system identification. Models can be broadly classified into two types characterized by the process of deriving the model:⁵ 1) phenomenological models, and 2) behavioral models. Typically, the phenomenological models are derived from basic principles and a theoretical formulation of the phenomenon involving the physics of the process. The theoretical formulation generally results from the Newtonian mechanics, and necessarily leads to parameters having physical interpretations. This approach usually leads to a system representation in the form of ordinary or partial differential equations. It is based on a detailed description of each part of the subprocess, and automatically leads to a predefined model structure based on the interactions in the subprocesses, providing a detailed representation of the real process in terms of model structure and system parameters. Such knowledge-based models may turn out to be highly complex. In some cases, simplifying assumptions are made to keep the theory tractable or in order to apply well-established analytical methods, or to arrive at lower order models.

Phenomenological models, being complex, may call for a high level of a priori information and effort in some applications. In other cases it may not even be possible to define the underlying physics. In yet other cases it may be adequate to approximate the observed behavior without specific intent on duplicating the internal system behavior. In all such cases behavioral models provide an alternative form to describe the overall cause–effect relationship, and are comparatively easier to derive. Models of this class can be developed using neural network techniques, which are characterized by: 1) input–output matching without any specific relevance to the internal behavior of the actual process, 2) reproduction of the overall system response, and 3) the so-called system parameters (or weights) are just numerical values without any physical meaning, which the engineers mostly look for.⁶

These two models differ in terms of imbedded a priori knowledge, system parameters, computation of system response, that is, simulation, and, as a result, the range of validity. Table 1.1 summarizes these important characteristics and differences between the two types of model formulations.⁵

Table 1.1 Important properties of two types of models

	Phenomenological models	Behavioral models
Parameters	Physical meaning	No concrete interpretation
Simulation	Complex and difficult	Quick and easy
A priori information	Included	Not necessary
Validity	Large	Restricted

In the case of flight vehicle system identification, we will mostly be dealing with phenomenological models which can be further classified into two categories: 1) parametric models and 2) nonparametric models. The state space or transfer function models belong to the category of parametric models which assume some suitable structure and order for the model. State space models could be linear or nonlinear, continuous or discontinuous, time-invariant or time-variant, and deterministic or stochastic. Transfer function representations are mostly based on linear models. Impulse or step response, frequency response, or power spectral density models belong to the second category of nonparametric models, which are not based on any specific model structure or order.

According to yet another classification, we can also characterize the models into the following three types: 1) white-box models, 2) black-box models, and 3) gray-box models. The “white-box” models are akin to the phenomenological models, whereas “black-box” models are akin to the behavioral models.

The “gray-box” models are, as the name implies, a combination of the two variants. Various degrees of combinations are possible. During the early stages of flight vehicle system identification and until recently, white-box models were mostly preferred. However, for specific operating conditions or combinations, this may not be possible, or if it is possible, it may involve substantial effort to derive white-box models satisfying the desired fidelity requirements. In other cases, safety critical issues may be based on white-box models, whereas further improvements could be obtained through appended models based on a black-box approach. In such cases the gray-box models play a useful role. Model cascading may be expanded to include a combination of incremental models based on any of the above concepts. For simulation purposes, it is basically immaterial in which form the model is available, as long as the simulation replicates the input–output behavior of the actual process.

III. Interdisciplinary Flight Vehicle Modeling

As pointed out in Sec. I, the scope and the basic purpose of simulation may differ significantly in the three interrelated disciplines: flight mechanics (FM), computational fluid dynamics (CFD), and structural mechanics (SM). For example, the flight mechanical simulations based on forces and moments, and structural applications involving generalized forces and moments, usually pertain to computing system responses (motion variables or structural deflections) to given inputs or disturbances. On the other hand, CFD simulations usually comprise computing pressure distributions which are used in the aircraft design and development phase for configuration optimization. In contrast to the purely flight mechanical models characterized adequately by low-order dynamics or to purely structural models characterized by higher-order dynamics, the aeroservoelastic (ASE) models cover both the low- and high-frequency ranges. An interdisciplinary flight vehicle modeling approach (Fig. 1.3) is necessary to arrive at appropriate models for aeroservoelastic applications.⁷ Advanced software tools catering to such interdisciplinary modeling and simulation have been developed.⁸ In such applications the model uncertainties play a dominant role.⁹

The ASE dynamics include coupling due to structural, control, sensor, aero, and actuator dynamics. Since accurate models can be obtained for structural

INTRODUCTION

7

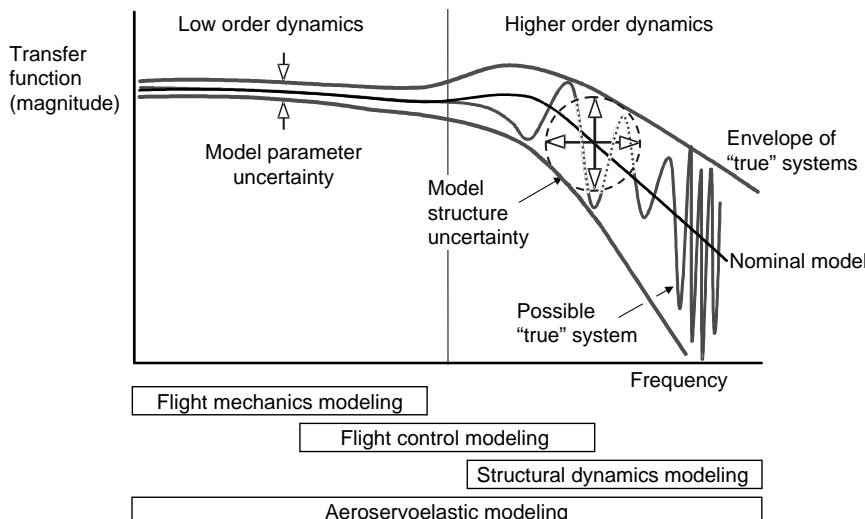


Fig. 1.3 Interdisciplinary flight vehicle modeling (source: Ref. 7).

dynamics through ground vibration tests or through finite-element method (FEM) computations, for the control dynamics through the design knowledge, and for the sensor dynamics through laboratory calibrations, the ASE modeling uncertainties are predominantly dictated by the aerodynamic and actuator models.¹⁰

Determination of actuator models from test data, mostly ground tests, is generally restricted to equivalent models. Such models, also validated on flight data, are usually low-order models, typically second to fourth order. They are obtained either in the time domain through state space representation by matching time histories of displacements or in the frequency domain by matching transfer function computed from the measured input–output data. However, for some applications it may be necessary to incorporate higher-order actuator models based on the physical properties of the system (main ram, servovalve, ram feedback, etc.) and its compliance with the structure.¹⁰

The aerodynamic models must include both the rigid body modes as well as the unsteady aerodynamics. The representative examples in this book will clearly demonstrate that the modern methods of system identification provide adequate tools to obtain high-fidelity flight-mechanics models. Hence, it is generally desirable to include such flight-derived and validated models in terms of aerodynamic derivatives instead of those based on the generalized aerodynamic forces. They are mostly used to investigate stability and control aspects, handling qualities investigations or control system design, and invariably for high-fidelity flight simulators. Such simulations are usually performed by solving an initial-value problem applying numerical integration procedures. The computational requirements are relatively low. Even for large transport aircraft, the influence of structural deformations can usually be described in a quasi-steady manner through a dynamic pressure dependent flex factor, provided the rigid-body

and structural frequencies are sufficiently separated.¹¹ Thereby the generalized equations of motion are usually avoided that are otherwise necessary to model the structural modes.¹² The structural models, particularly in the case of a large flexible airplane, can influence the accuracy of the flight mechanics models, i.e., of the stability and control derivatives. The structural effects on the stability and control derivatives can be removed through filtering of data, or by applying pseudostatic or dynamic structural modeling techniques.¹³ The advanced parameter estimation methods discussed in this book and applied widely to estimate flight mechanical models are also being used to derive flight-validated models for aeroelastic and flutter analysis.¹⁴

The CFD and ASE prediction and simulation codes are being applied in advanced research and industrial applications.^{15,16} Application of the CFD and computational unsteady aerodynamics (CUA) methods to predict and validate nonlinear and unsteady aerodynamic responses and aircraft motion time histories was initiated and postulated by Tobak and Schiff.^{17–19} There is no doubt that the CFD has revolutionized the aircraft design process and is capable of providing solutions even for complete configurations.²⁰ However, applicability of such solutions in the subsequent aircraft applications or flight vehicle life cycle is plagued with two restrictions: 1) enormous computational requirements compared with the classical FM models, and 2) validation based on experimental data.

Although forces and moments required for the FM models can be obtained by integration of the pressure distributions provided by the CFD solutions, the enormous computational burden makes them unsuitable for flight mechanical investigations. Looking at the tremendous advancements in the computer technology, workstations with significantly more computational power will become available in the future. However, the gap between the computational requirements of FM and CFD models is drastic. One possible approach would be to provide alternative model representations to the available CFD solutions, or approximate them, using system identification methods such as time series modeling, input–output subspace matching or other similar approaches.²¹ Integrating these related technologies is a challenge for the future, and the potential of system identification is likely to pave the way for the task. The other aspect of validation based on experimental data is equally involved²² and calls for measurements of flow field velocity and other surface variables such as pressures.²³ Today, the validation of CFD results is mostly attempted by comparing them with wind-tunnel measurements; CFD validation from flight test data is attempted only in very few cases.^{24–29} On the other hand, as will be demonstrated in this book, validation of FM models from data gathered during flight testing is a routine and well-established procedure. We will be dealing here with the FM models derived and validated from flight data.

IV. Why System Identification?

Having discussed the meaning of system identification and the classification of models, we now turn our attention to another important aspect, namely why and where do we need system identification. In flight vehicle development, it is a necessary step because system identification leads to adequately accurate

INTRODUCTION

9

and validated mathematical models of the flight vehicle, which are required to 1) understand the cause–effect relationship that underlies a physical phenomenon, 2) investigate system performance and characteristics, 3) verify wind-tunnel and analytical predictions, 4) develop high-fidelity aerodynamic databases for flight simulators meeting FAA fidelity requirements, 5) support flight envelope expansion during prototype testing, 6) derive high-fidelity and high-bandwidth models for in-flight simulators, 7) design flight control laws including stability augmentation systems, 8) reconstruct the flight path trajectory, including wind estimation and incidence analysis, 9) perform fault-diagnosis and adaptive control or reconfiguration, and 10) analyze handling qualities specification compliance.

Although each of these reasons is self-explanatory, we take a look at the two most important arguments. First, system identification yields a complete breakdown of the various components contributing to the observed response, and thereby provides an overall understanding of the flight vehicle’s dynamics. Second, system identification yields an accurate and comprehensive database for flight simulators and off-line digital simulations, which are extensively used for pilot training and to minimize risk during experimental testing, which is very costly. The uncertainties in the parameters (aerodynamic database) affect the simulation results and predictions based thereupon.

V. Parameter Estimation in Flight Mechanics

A flight vehicle is in general free to move in any direction and can be described by coupled equations of motion, which are derived from Newtonian mechanics, usually considering the flight vehicle as a rigid body. These equations involve the fundamental assumption that the forces and moments acting on the flight vehicle can be synthesized. Validity and utility of the mathematical models depend to a large degree on the adequacy and accuracy with which these external forces and moments acting on the flight vehicle can be modeled. The various forces and moments acting on a flight vehicle can be broadly classified as 1) aerodynamic, 2) inertial, 3) gravitational, and 4) propulsive. During the ground roll additional forces and moments due to landing gear are encountered. The central issue in system identification of flight vehicles, be it a conventional transport aircraft, a highly augmented unstable aircraft flying at high angles of attack, a rotorcraft, or a missile, is to postulate appropriate models for the aerodynamic forces. The aerodynamic forces and moments acting on the flight vehicle cannot be measured directly. However, aerodynamic modeling followed by parameter estimation allows determination of specific aerodynamic characteristics (lift, drag, and side force coefficients, and rolling, pitching, and yawing moment coefficients in terms of stability and control derivatives) from the related measurements such as accelerations, angular rates, and flow angles.

Aerodynamic modeling, which provides a means of obtaining relationships between the three forces X , Y , Z along the three Cartesian coordinates and the moments L , M , N about these axes as functions of the linear translational motion variables u , v , w , rotational rates p , q , r and control surface deflections, was introduced by Bryan in the early twentieth century.³⁰ It is more convenient to express the aerodynamic reactions in terms of nondimensional coefficients, since forces and moments depend on dynamic pressure. Different axes systems

are defined to facilitate aerodynamic modeling. For example, in the body-fixed system the three force coefficients and three moment coefficients are given by C_x , C_y , C_z , C_ℓ , C_m , C_n . Typically, all the coefficients depend on the flight condition, for example,

$$C_m = C_m(u, v, w, p, q, r, \delta_e, \dots) \quad (1.1)$$

which can be expanded as:

$$C_m = \frac{\partial C_m}{\partial u} u + \dots + \frac{\partial C_m}{\partial q} q + \dots + \frac{\partial C_m}{\partial \delta_e} \delta_e + \dots \quad (1.2)$$

The partial derivatives with respect to the motion variables and control inputs, appearing as coefficients in the above Taylor series expansion, are defined as the stability and control derivatives. This model postulate is amenable to parameter estimation and marks the beginning of the evolution of flight vehicle system identification. The developments over the last nine decades have led to three different but complementary techniques of determining aerodynamic coefficients: 1) analytical methods, 2) wind-tunnel methods, and 3) flight test methods.

The first two techniques are employed to generate basic information about the flight mechanical parameters. The analytical estimations may be based on certain simplified assumptions for a complex phenomenon. Nevertheless, as elaborated in Sec. III, computational fluid dynamics has in recent years positively influenced the analytical scenario by providing numerical solutions of complete configurations via sophisticated and advanced Euler and Navier–Stokes flow solvers.^{31,32} Experimental methods are essential to corroborate the analytical predictions. Wind-tunnel techniques, which are comparatively less expensive, have provided a huge amount of data on innumerable flight vehicle configurations and are, as a rule, a basis for any new flight vehicle design. These predictions are, however, associated with model scaling, Reynolds number, dynamic derivatives, cross coupling and aeroservoelasticity effects. Moreover, these measurements could also be affected by tunnel unsteadiness or model support vibrations and interferences. Furthermore, some of the operational test conditions cannot be adequately tested in wind tunnels. Determination of aerodynamic derivatives from flight measurements is, therefore, important and necessary to reduce limitations and uncertainties of the two methods.³³ Such an approach, although highly desirable, could turn out to be expensive if flight tests are to be performed at all flight conditions.

Because we are dealing with real processes and measured experimental data, certain complications arise owing to 1) the presence of sensor errors and measurement noise, 2) modeling errors, and 3) atmospheric turbulence. A block schematic of a system representation incorporating these practical issues is shown in Fig. 1.4. Determination of parameters in the presence of measurement noise requires an estimation procedure based on some statistical criterion; loosely speaking we have to average out the noise effects on the parameter estimates. The measured data is likely to contain errors owing to sensor calibration. Depending

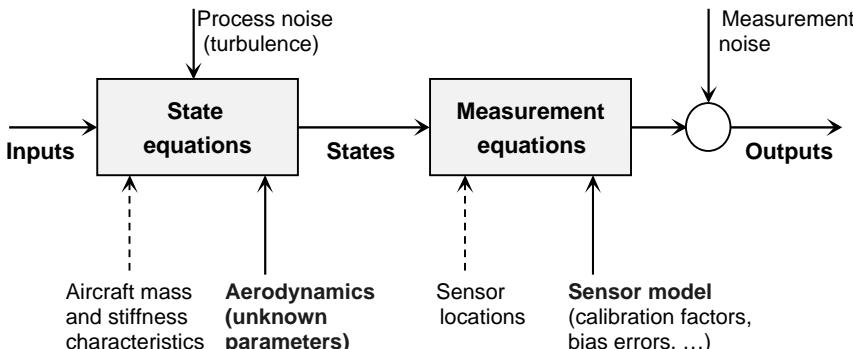


Fig. 1.4 Block schematic of system model for aircraft parameter estimation.

upon the location, the sensor may also measure contributions due to other motion variables, besides the primary variable itself. The signal processing and onboard recording system is likely to introduce time delays. Thus, appropriate sensor models will have to be included in the overall system representation.

Incorporation of the aerodynamic model in the equations of motion needs additional information in terms of aircraft mass characteristics (mass, moments of inertia, center of gravity position). Certain modeling errors are inevitable, because the postulated aerodynamic model is an approximation of complex flow phenomena including interference effects. These errors are treated in the estimation process simply as noise. Rigorously this is not justifiable, because modeling errors are deterministic, but in the absence of any better alternative this is probably the best approach. The last aspect, atmospheric turbulence, also called process noise, is likewise unavoidable. The atmospheric turbulence during the flight test can be derived indirectly from the measurement of flow variables and flight-path velocity vector and used as additional model inputs. In the absence of such information, the presence of turbulence makes the dynamic system stochastic and significantly complicates the computation of the states.

VI. Estimation Techniques of the Past

The importance of obtaining flight-derived aircraft parameters was recognized early in aircraft evolution. A couple of years after the introduction of the classical stability approach by Bryan, Glauert's work in 1919 on the analysis of phugoid motion and that of Norton during 1919–1923 on estimating a number of derivatives such as L_p , Y_v , L_v , N_v , and M_w mark the beginning of the experimental investigation of dynamic stability from actual flight data.^{34–36} Interest in the dynamic behavior of aircraft grew steadily over the period. During the late 1940s and early 1950s, several techniques were introduced, such as steady-state oscillatory excitations by Milliken,^{37,38} pulse-input methods incorporating Fourier analysis by Seamans et al,³⁹ weighted least-squares,⁴⁰ and response curve fitting by Shinbrot.⁴¹ An excellent account of dynamic stability and control research during this early period is found in Ref. 38 and a survey of methods for determining stability and control derivatives from dynamic flight

measurements in Ref. 42. During the 1950s and 1960s, the time-vector method was one of the commonly applied graphical procedures of determining aerodynamic derivatives from free oscillations initiated by an abrupt pulse.^{43–48} The time-vector decomposition had to be carried out separately for each force and moment equation. The analog matching techniques were also applied towards the end of this era.^{49,50}

The techniques of the 1940s to early 1960s were mostly frequency response methods, and were either limited to estimation of incomplete sets of coefficients, to simple motions, or restricted for other reasons.⁵¹ This sketch of the historical background is by no means exhaustive. Nevertheless, they represent important milestones in the history of aircraft parameter estimation prior to the advent of digital computers. A brief account of some of these early methods, which have practically become techniques of the past, is found in Refs. 7 and 52.

Although Shinbrot had introduced the response curve fitting method,⁴¹ which is equivalent to today's output error method, it was at that time found to be impracticable owing to the lack of adequate (digital) computational means. During the early 1960s, analog matching was a popular technique of updating and validating wind-tunnel predictions of stability derivatives based on flight tests.^{49,50} Briefly stated, the mathematical model, in several instances the decoupled equations of aircraft motion, was programmed on an analog computer, incorporating theoretical or wind-tunnel predictions of the stability derivatives as first approximations. The flight recorded control inputs, duplicated through function generators using diodes, amplifiers, etc., were fed into the simulation. Comparing qualitatively the simulated response with the actual aircraft response, the stability parameters were manually tuned to reduce the differences. The analog matching technique, although appealing, had several shortcomings, namely 1) the success or failure depended on the operator's ingenuity in adjusting the proper parameter, 2) it was time consuming, and 3) it was limited to a small number of primary derivatives. Furthermore, the quality of the flight data significantly affected the tuning procedure and involved qualitative judgment of the operator; these two aspects will be involved in the modern methods of system identification to be discussed next, but they provide a more systematic approach to tackle these problems and minimize the resulting uncertainties.

VII. Modern Methods of Aircraft Parameter Estimation

The modern era of system identification is marked by the implementation of the maximum-likelihood method, which we will discuss in Chapters 4 and 5, on a digital computer by Åström and Bohlin in 1965, and its application to an industrial plant represented by difference equations.⁵³ The automatic data processing capability provided by digital computers dramatically changed the focus of flight data analysis from frequency-domain methods to time-domain methods. It became possible to obtain a significantly larger number of stability and control derivatives from a single flight test. A coordinated approach based on flight test techniques, flight test instrumentation, and methods of data analysis gradually evolved for flight vehicle system identification:

- 1) Instrumentation and filters cover the entire flight data acquisition process including adequate instrumentation and airborne or ground-based

INTRODUCTION

13

recording equipment. The effects of all kinds of data quality have to be accounted for.

2) Flight test techniques are related to the selected flight vehicle maneuvering procedures. The input signals have to be optimized in their spectral composition in order to excite all response modes from which parameters are to be estimated.

3) Analysis of flight data includes the mathematical model of the flight vehicle and an estimation criterion which devises some suitable computational algorithm to adjust some kind of starting values or a priori estimates of the unknown parameters until a set of best parameter estimates is obtained which minimizes the response error.

Corresponding to these strongly interdependent topics, four important aspects of the art and science of system identification have to be carefully treated, namely maneuver, measurements, methods, and models (Fig. 1.5): 1) design of the control input shape in order to excite different modes of the vehicle dynamic motion; 2) selection of instrumentation and filters for high accuracy measurements; 3) quality of data analysis by selecting the most suitable time or frequency domain identification method; 4) type of flight vehicle under investigation in order to define the structure of a possible mathematical model.

These "Quad-M" requirements must be carefully investigated for each flight vehicle, and are the key to successful flight vehicle system identification.⁷ They lead to an interdisciplinary task, covering the fields of statistical theory, control theory, numerical techniques, sensor and instrumentations, signal processing, flight test techniques, and flight mechanics. To apply this methodology

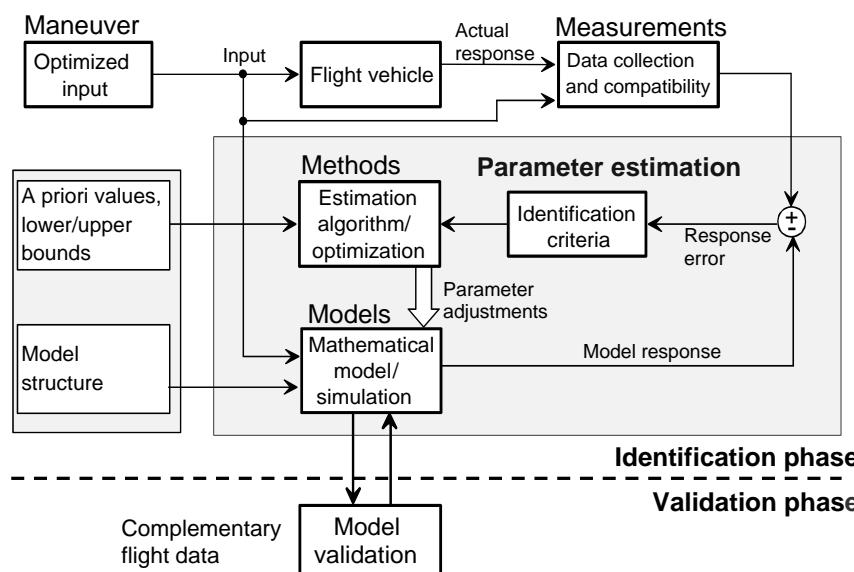


Fig. 1.5 Quad-M basics of flight vehicle system identification.

efficiently, appreciation of basic aspects of each of them is useful in terms of advantages, limitations, and likely problems. The Quad-M requirements fall within the framework of the contemporarily and widely accepted definition provided by Zadeh.³ A systematic treatment of these key issues is followed in all the applications presented in this book, focusing mainly on methods and modeling aspects.

VIII. General Aspects

The subject of statistical estimation theory being too vast, the number of flight vehicle applications too diversified, and the published literature too abundant, it is necessary to define the scope, the purpose, and overall organization of the book.

A. The Aim

The present book aims to provide a consolidated account of state of the art of time domain system identification with underlying mathematical principles from an engineer's view. It also aims to share the experience gained on aerodynamic modeling from flight test data applying time domain methods on a large number of flight vehicles, because experience and engineering judgment are critical to generate good results. The objective is to expose the art and science of system identification by providing adequate theory, and to focus on explanation of the algorithmic steps and intricacies involved in the applications of the various methods. We aim to raise some of the issues, which may deviate from procedures advocated in the past, and this may lead to some discussion, but then the reported schemes have worked in practice, providing an acceptable solution in some of the most demanding tasks. We will also bring out some topics for future research. We aim to address a broad spectrum of modeling cases with different degrees of complexities, in each case emphasizing practical issues.

B. The Literature

Flight vehicle system identification has been pursued by almost every organization dealing with the subject of atmospheric flight. In particular, the major contributions to theory and applications have resulted from research activities in several organizations: NASA Dryden and Langley RC, US Army Aeroflight-dynamics Directorate at NASA Ames RC, Calspan, USA; Institute of Flight Systems, DLR, Germany; TU-Delft and NLR, The Netherlands; DRA, England; NRC, Canada; and several other academic institutions. Pioneering contributions were made by individuals such as Taylor and Iliff,⁵⁴ Mehra,⁵⁵ and Gerlach.⁵⁶ For more than three decades the aerospace industry has been using this methodology for model validation, handling qualities evaluations, control law design, and flight vehicle design.

The symbiosis of these international efforts has culminated in the present state of maturity. This has become possible mainly because of dissemination of information through published literature, in which American Institute of Aeronautics and Astronautics (AIAA) journal publications have played a major role. On every facet of aerodynamic modeling there are excellent technical papers, mostly in the

AIAA Journal of Aircraft, including the most recent two special focus issues on “Flight Vehicle System Identification—Engineering Utility.”^{57,58} Broad overviews are found in the survey papers published by AIAA: 1) Iliff, with 76 references summarizing contributions to flight vehicle system identification up to 1980,⁵² and 2) Hamel and Jategaonkar, having 183 references presenting the state of the art up to 1995.⁷ Other overview contributions are by 1) Klein, giving an account of system identification methodology,⁵⁹ 2) Padfield, collecting 13 contributions in rotorcraft applications,⁶⁰ and 3) Hamel and Kaletka surveying the progress in rotorcraft system identification up to 1997.⁶¹

The former Advisory Group for Aerospace Research and Development (AGARD), and to some extent the successor organization Research and Technology Organization (RTO) effectively stimulated and fostered international scientific and technical interchange, collaboration, and documentation in the field of flight vehicle system identification. Notable among the various publications are the three AGARDOGraphs (AG), one on estimation theory by Maine and Iliff,⁶² and two on aircraft applications, one by Maine and Iliff,⁶³ and the other by Mulder et al.⁶⁴ An excellent repertoire on rotorcraft applications is provided by the Advisory Report (AR) edited by Hamel.⁶⁵ Likewise, the AG compiled by Stoliker is a significant volume, providing a consolidated account on flight testing.⁶⁶ An exhaustive treatment of the various aspects of flight test instrumentation for all possible applications is found in the 19 volumes of Ref. 67. The unique RTO effort in bringing together the experts from four streams, namely dynamics, system identification, control, and handling qualities, is also noteworthy addressing the land, air, sea, and space vehicle applications to highlight challenges faced and the similarity in the employed expertise.⁶⁸

To a smaller extent contributions come from the International Federation on Automatic Control (IFAC) symposia, whose focus has been mainly on the theory and applications in areas other than flight vehicles. Apart from the aforementioned literature related to flight vehicle applications, the basic estimation theory is covered in textbooks such as those by Walter and Pronzato,⁵ Eykhoff,⁶⁹ Goodwin and Payne,⁷⁰ Sorenson,⁷¹ Ljung and Söderström⁷² and others.

We include at the end of each chapter a fairly exhaustive list of references pertinent to the topics discussed. These references help to trace the development and also to show the widespread applicability of this methodology. We have also tried to provide some historical references, which help us to appreciate the engineering approach and deep understanding of the physical phenomenon being analyzed by the early pioneers. If for any reason one or another publication is missed out, the omission is unintentional and without any bias. In this book, it is not our claim that the references to the published literature provide a complete bibliography.

C. The Layout

In Chapter 1, we have provided a general background on system identification, including various definitions, the purpose and the need. It has been emphasized that system identification, parameter estimation, and simulation are three inter-related aspects. After a brief account of the historical background, the Quad-M basics of the modern approach to flight vehicle system identification have been

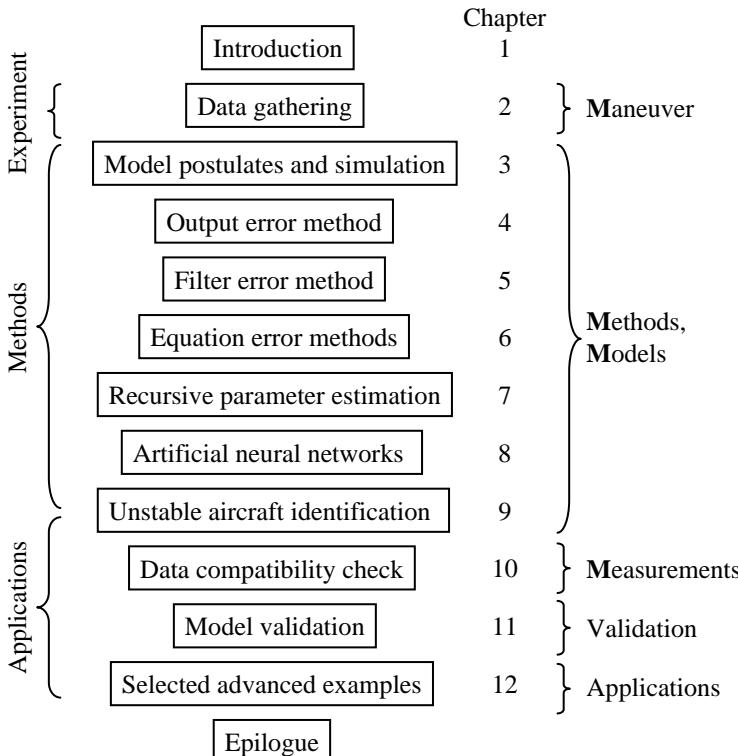


Fig. 1.6 The book layout.

introduced. The general layout of the book is summarized in Fig. 1.6, which shows at a glance that the complete material covered in this book can be broadly split into three important parts shown on the left-hand side, namely experiment, methods, and applications. Some overlap of methods and models is unavoidable. The references to the four Ms of the Quad-M basics and to model validation have also been indicated in the same figure on the right-hand side.

In Chapter 2, we deal with the first of the four Ms, namely maneuvers leading to data gathering. A brief introduction to flight testing relevant to system identification is provided. Typical flight maneuvers and commonly applied inputs to excite dynamic motion are presented. Although design of optimal inputs has been the subject of extensive research, we approach this aspect more pragmatically from the practical implementation viewpoint. A brief account of flight measurements required for parameter estimation is presented. This is part of the second M for measurement; the other part is covered later in the book for reasons mentioned subsequently.

Chapter 3 introduces various mathematical model formulations that form the basis of the models analyzed in the following chapters. Several model extensions are proposed to account for errors in the measurements, to analyze multiple

INTRODUCTION

17

experiments simultaneously, and to estimate time delays. Commonly applied numerical integration procedures are briefly presented, making a recommendation for the suitable choice.

In Chapter 4, we introduce the most widely used maximum-likelihood principle, extend the same to general nonlinear dynamic systems and develop the output-error method. Three gradient-based algorithms, namely Gauss–Newton, quasi-linearization, and Levenberg–Marquardt, are presented, in each case discussing the algorithmic details. Solutions necessary to overcome convergence problems in some of these methods are presented. This is followed by introducing more recent techniques, for example constrained optimization based on the bounded-variable Gauss–Newton method or the interior-point method. A brief summary of direct search optimization methods is provided. Some future possible extensions are brought out. Details of efficient implementation to handle large-scale systems are elaborated. The techniques discussed here are applied to two test cases.

Chapter 5 presents the theory and development of the most complex and most general aircraft parameter estimation problem, namely estimation accounting for atmospheric turbulence. The presence of process noise makes the system stochastic, requiring state estimation techniques to be incorporated in parameter estimation. Such techniques are called filter error methods. Extensions to nonlinear systems are presented, giving step-by-step computational procedures for linear as well as nonlinear systems. The filter-error methods are applied to typical examples, highlighting the need, advantages, and limitations of these methods.

The methods presented in Chapters 4 and 5 belong to the class of output-error methods and represent the two central topics of this book. In Chapter 6, we cover the methods belonging to the other class, namely equation-error methods. We present the ordinary least squares, weighted least squares, and total least squares methods, and discuss statistical properties of the estimates, practical aspects in the presence of noise and demonstrate extensions to state space models under certain assumptions. Nonlinear and multiple-output regression techniques are covered. We take a pragmatic look at the process of model-structure determination. Finally, we apply the one-shot and the iterative least squares methods to linear and nonlinear models and the total least squares method to linear model.

The methods presented in Chapters 4–6 evaluate the complete set of data together at one time and are suitable for off-line analysis. In Chapter 7, we cover some basic applications of recursive estimation methods that are suitable for on-line implementations. Five different techniques, namely the recursive least squares method, the discrete Fourier transform method, the extended Kalman filter, the unscented Kalman filter in both simplified and general form, and the extended forgetting factor recursive least squares, are presented. They are applied to flight data and a comparative study based on the results is presented.

In Chapter 8 we cover a related approach to overall system characterization by mapping the input–output subspace. The basic concepts of neural processing are elaborated, concentrating on the feedforward neural networks that lead to black-box models, without explicit knowledge about the internal behavior and physics

of the phenomenon. Various training algorithms are presented. These techniques are applied to three different examples of increasing complexity. The advantages, limitations, and issues related to practical applications of such networks in providing a general function approximation of any nonlinear phenomenon are brought out.

Chapter 9 addresses exclusively unstable aircraft identification. To start with, the problems encountered in the open-loop identification of inherently unstable aircraft are discussed. It is demonstrated that the difficulties encountered are practical issues of applying system identification and not of estimation methodology being applicable or not. This is followed by a summary of estimation methods applicable to unstable aircraft, highlighting that mostly the same estimation algorithms covered in the preceding chapters or some variants of them are applicable using measured data within the simulation and gradient computation. It is demonstrated that the major problem is due to data collinearity resulting from highly correlated control inputs and aircraft motion variables, which can be eliminated efficiently through separate surface excitations. A detailed comparative study is presented to evaluate different algorithms and finally three of them are applied to flight data from an unstable aircraft.

Chapter 10 covers the various aspects of data compatibility checking, which ensures that recorded data are consistent and error free. Techniques applied to estimate the aircraft states and/or instrument errors are presented. It is argued that the output-error method developed in Chapter 4 is often adequate for this purpose. It is applied to typical flight data gathered from longitudinal and lateral-directional motions. Calibrations of a five-hole probe for flow angles and of static pressure ports are presented in some details. According to the Quad-M basics presented earlier in the current chapter, the logical sequence of presentation would have been the data gathering covered in Chapter 2 pertaining to the first M for maneuver, followed by the second M for measurements, and then the other two Ms, namely models and methods, which are covered in Chapters 3–9. Maneuvers and measurements are prerequisites for the off-line data analysis covering models and methods. We have, however, presented the discussion on data compatibility checking corresponding to the second M for measurements in Chapter 10, mainly because verification of data and estimation of instrument error requires algorithms that are part of the methods covered in Chapters 4–6.

Chapter 11 deals with model validation aspects, covering different techniques applied to assess the accuracy of estimated parameters and the fidelity of the identified models. Besides the statistical accuracy, residual analysis and inverse-simulation techniques are presented. Model predictive capability is the decisive factor in judging the fidelity; some practical issues in this connection are covered.

Chapter 12 provides an account of a wide variety of aerodynamic modeling and parameter estimation problems, and analyzes flight data from different flight vehicles. A total of 11 different cases covering nonlinear models, primary and secondary aerodynamic effects, control surface malfunction, unsteady aerodynamics, symmetric and asymmetric quasi-steady stall, verification and update of high-fidelity databases, wake vortex encounters, and rotorcraft are presented. The analyzed flight vehicles range from fixed-wing aircraft (high-wing and

low-wing configurations), two-propeller aircraft, and unstable aircraft to reusable launch vehicle demonstrators and rotorcraft vehicles.

Finally, we conclude with an Epilogue, summarizing the adopted system identification philosophy, which has worked well in practice. We attempt to indicate probable future directions of flight vehicle system identification.

The chapters are mostly self-contained. Although reading the chapters in a sequence as presented in the book will help to gain a systematic insight into the complete complex of flight vehicle system identification, some of the specific topics, like the filter error method covered in Chapter 5 or unstable aircraft identification dealt with in Chapter 9, can be handled separately. Certain familiarity with matrix algebra, statistical theory and system theory will be helpful, whereas basic knowledge of flight mechanics in terms of equations of motion and modeling of forces and moments acting on flight vehicles is presumed.^{73,74}

D. Software and Flight Data

The book includes software programs which can be downloaded from the AIAA website (for details see Supporting Materials). The programs are mostly sorted chapterwise and implement most of the algorithms described in the book. The software for the filter-error method capable of handling general nonlinear systems is provided for the first time. The flight data analyzed in the test cases is supplemented to the book. It is in the form of data files containing time histories of measured aircraft inputs and responses. The main aim of providing software is to help the reader better trace and correlate theory, computational steps, and results presented here. This will be possible through the algorithmic implementations elaborated in the text and comments provided in the programs, including equation numbers as they appear in the individual chapters. To facilitate this purpose, we have included the description of software in the individual chapters to allow treatment of a particular topic at one stretch. This differs from the usual practice of providing the software description as an Appendix at the end; it leads to a better continuity and awakening of the reader's specific interests. We do not wish to provide a frozen software, that is, only the executable version, with few options to modify or extend to suit individual needs. Rather, we wish to provide the reader with an opportunity to become himself involved in the algorithmic details and programming. A deeper understanding of the estimation algorithms is helpful in appreciating the advantages and limitations of the methods.

The software is developed under the Matlab[®] computing environment.⁷⁵ It has been applied to test cases presented at the end of the chapters, describing the models used and data files analyzed. These fairly simple test cases enable familiarity with the software itself and also aid in tracing the results presented. The software has been developed using the basic Matlab only. Additional toolboxes would lead to different ways of implementing the algorithms and possibly to a much simpler programming. For example, 1) availability of the Control Toolbox would allow the Riccati equation in Chapter 5, Sec. IV.A to be solved using the function CARE, although we have provided a simple basic version to do this job; 2) definition of a state space model is through user-defined functions. Availability of Matlab/Simulink[®] would change not only the interface, but also the loops over the data points. Such modifications are left to the reader.

The software has been tested with three Matlab® Versions: 1) Version 6.0.0, Release 12, 2) Version 6.5.1, Release 13 (Service Pack 1), and 3) Version 7.0.0, Release 14. Matlab® itself is not a part of the book or of the software provided, but a product of The MathWorks, Inc.⁷⁵

References

- ¹Boehner, P. (ed. and trans.), *William of Ockham: Philosophical Writings*, Hackett, Indianapolis, IN, 1990.
- ²Maurer, A. A., "Method in Ockham's Nominalism," *The Monist*, Vol. 61, No. 3, 1978, pp. 426–443.
- ³Zadeh, L. A., "From Circuit Theory to System Theory," *Proceedings of the IRE*, Vol. 50, May 1962, pp. 856–865.
- ⁴"He Has a Question for Any Answer," Interview with K. Iliff by M. McCall, Antelope Valley Press, CA, 1 Nov. 1994.
- ⁵Walter, É. and Pronzato, L., *Identification of Parametric Models*, Springer, Berlin, 1997.
- ⁶Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Deylon, B., Gorenne, P.-Y., Hjalmarsson, H., and Juditsky, A., "Nonlinear Black-Box Modelling in System Identification: A Unified Overview," *Automatica*, Vol. 31, No. 12, 1995, pp. 1691–1724.
- ⁷Hamel, P. G. and Jategaonkar, R. V., "Evolution of Flight Vehicle System Identification," *Journal of Aircraft*, Vol. 33, No. 1, 1996, pp. 9–28.
- ⁸Gupta, K. K., "STARS—an Integrated, Multidisciplinary, Finite-Element, Structural, Fluids, Aeroelastic, and Aeroservoelastic Analysis Computer Program," NASA TM-4785, May 1997.
- ⁹Livne, E., "Future of Airplane Aeroelasticity," *Journal of Aircraft*, Vol. 40, No. 6, 2003, pp. 1066–1092.
- ¹⁰Brenner, M. J., "Actuator and Aerodynamic Modeling for High-Angle-of-Attack Aeroservoelasticity," NASA TM-4493, June 1993.
- ¹¹Hamel, P. G. and Jategaonkar, R. V., "The Role of System Identification for Flight Vehicle Applications: Revisited," RTO-MP-11, Paper 2, March 1999.
- ¹²Waszak, M. R. and Schmidt, D. K., "Flight Dynamics of Aeroelastic Vehicles," *Journal of Aircraft*, Vol. 15, No. 6, 1988, pp. 563–571.
- ¹³Iliff, K. W., "Aircraft Identification Experience," AGARD LS-104, Paper 6, Nov. 1979.
- ¹⁴Perangelo, H. J. and Waisanen, P. R., "Application of Advanced Parameter Identification Methods for Flight Flutter Data Analysis with Comparison to Current Techniques," AGARD, CP-373, Paper 5, April 1984.
- ¹⁵Meijer, J. J., Hounjet, M. H. L., Eussen, B. J. G., and Prananta, B. B., "NLR-TUDelft Experience in Unsteady Aerodynamics and Aeroelastic Simulation Applications," in "Numerical Unsteady Aerodynamics and Aeroelastic Simulation," AGARD R-822, Paper 11, March 1998.
- ¹⁶Vos, J. B., Rizzi, A., Darracq, D., and Hirschel, E. H., "Navier–Stokes Solvers in European Aircraft Design," *Progress in Aerospace Sciences*, Vol. 38, 2002, pp. 601–697.
- ¹⁷Tobak, M. and Schiff, L. B., "Aerodynamic Mathematical Modeling—Basic Concepts," AGARD LS-114, Paper 1, May 1981.
- ¹⁸Tobak, M. and Chapman, G. T., "Nonlinear Problems in Flight Dynamics Involving Aerodynamic Bifurcation," AGARD CP-386, Paper 25, Nov. 1985.

INTRODUCTION

21

- ¹⁹Hui, W. H. and Tobak, M., "Bifurcation Theory Applied to Aircraft Motions," AGARD CP-386, Paper 26, Nov. 1985.
- ²⁰Tinoco, E. N., "An Assessment of CFD Prediction of Drag and Other Longitudinal Characteristics," AIAA Paper, 2001-1002, Jan. 2001.
- ²¹Cowan, T. J., Arena, Jr., A. S., and Gupta, K., "Accelerating Computational Fluid Dynamics Based Aeroelastic Predictions Using System Identification," *Journal of Aircraft*, Vol. 38, No. 1, 2001, pp. 81–87.
- ²²"Verification and Validation Data for Computational Unsteady Aerodynamics," RTO-TR-26, Oct. 2000.
- ²³Marvin, J. G., "Perspective on Computational Fluid Dynamics Validation," *AIAA Journal*, Vol. 33, No. 10, 1995, pp. 1778–1787.
- ²⁴Hawkins, R. W. and Dilley, A. D., "CFD Comparisons with Wind Tunnel and Flight Data for the X-15," AIAA Paper 1992-5047, 1992.
- ²⁵Roberts, A. T. and Henderson, T. L., "Baseline CFD and Flight Test Comparison for SOFIA," AIAA Paper 1999-0533, Jan. 1999.
- ²⁶Shivananda, T., McKeel, S., Salita, M., and Zabrensky, E., "Space Launch Vehicle Aerodynamics—Comparison of Engineering and CFD Predictions with Wind Tunnel Data," AIAA Paper 2001-0258, Jan. 2001.
- ²⁷Hübner, A.-R. and Loeser, T., "Methods for Determination of the Unsteady Aerodynamic Derivatives for Transport Aircraft Configurations," RTO-MP-AVT-123, Paper 12, March 2005.
- ²⁸Klausmeyer, S. M., "Drag, Lift, and Pitching Moment Estimates of Transonic Aircraft Using the Navier-Stokes Equations," AIAA Paper 2004-0533, Jan. 2004.
- ²⁹Babcock, D. A. and Arena, Jr., A. S., "Estimating Aircraft Stability Derivatives Through Finite Element Analysis," AIAA Paper 2004-5174, Aug. 2004.
- ³⁰Bryan, G. H., *Stability in Aviation: An Introduction to Dynamical Stability as Applied to the Motions of Aeroplanes*, Macmillan, London, 1911.
- ³¹Kroll, N., Radespiel, R., and Rossow, C.-C., "Accurate and Efficient Flow Solvers for 3D Applications on Structured Meshes," VKI Lecture Series 1994-05, Brussels, March 1994.
- ³²Slooff, J. W. and Schmidt, W. (eds), "Computational Aerodynamics Based on the Euler Equations," AGARD AG-325, Sept. 1994.
- ³³Hamel, P. G., "Determination of Aircraft Dynamic Stability and Control Parameters from Flight Testing," AGARD LS-114, Paper 10, May 1981.
- ³⁴Glaert, H., "Analysis of Phugoids Obtained by a Recording Airspeed Indicator," A.R.C. R&M No. 576, Jan. 1919.
- ³⁵Norton, F. H., "The Measurement of the Damping in Roll on a JN4h in Flight," NACA Report 167, 1923.
- ³⁶Norton, F. H., "A Study of Longitudinal Dynamic Stability in Flight," NACA Report 170, 1923.
- ³⁷Milliken, W. F. Jr., "Progress in Dynamic Stability and Control Research," *Journal of the Aeronautical Sciences*, Vol. 14, No. 9, 1947, pp. 493–519.
- ³⁸Milliken, W. F. Jr., "Dynamic Stability and Control Research," *Proceedings of the 3rd Anglo-American Aeronautical Conference*, Brighton, UK, 1951, pp. 447–524.
- ³⁹Seamans, R. C. Jr., Blasingame, B. P., and Clementson, G. C., "The Pulse Method for the Determination of Aircraft Dynamic Performance," *Journal of the Aeronautical Sciences*, Vol. 17, No. 1, 1950, pp. 22–38.

- ⁴⁰Shinbrot, M., "On the Analysis of Linear and Nonlinear Dynamical Systems from Transient Response Data," NACA TN-3288, Dec. 1954.
- ⁴¹Shinbrot, M., "A Least-Squares Curve Fitting Method with Applications to the Calculation of Stability Coefficients from Transient Response Data," NACA TN 2341, April 1951.
- ⁴²Greenberg, H., "A Survey of Methods for Determining Stability Parameters of an Airplane from Dynamic Flight Measurements," NACA TN-2340, April 1951.
- ⁴³Mueller, R. K., "The Graphical Solution of Stability Problems," *Journal of the Aeronautical Sciences*, Vol. 4, June 1937, pp. 324–331.
- ⁴⁴Breuhaus, W. O., "Resumé of the Time-Vector Method as a Means for Analyzing Aircraft Stability Problems," WADC TR-52-299, Wright Air Development Center, US Air Force, Nov. 1952.
- ⁴⁵Doetsch, K. H., "The Time Vector Method for Stability Investigations," RAE Report, Aero 2495, Aug. 1953; also as ARC R&M No. 2945, 1957.
- ⁴⁶Larrabee, E. E., "Application of the Time-Vector Method to the Analysis of Flight Test Lateral Oscillation Data," Cornell Aero. Lab. Inc., Report FRM No. 189, Sept. 1953.
- ⁴⁷Sternfield, L., "A Vector Method Approach to the Dynamic Lateral Stability of Aircraft," *Journal of the Aeronautical Sciences*, Vol. 21, No. 4, 1954, pp. 251–256.
- ⁴⁸Wolowicz, C. H. and Holleman, E. C., "Stability Derivative Determination from Flight Data," AGARD Report 224, Oct. 1958.
- ⁴⁹Rampy, J. M. and Berry, D. T., "Determination of Stability Derivatives from Flight Test Data by Means of High Speed Repetitive Operation Analog Matching," FTC-TDR-64-8, Edwards, CA, May 1964.
- ⁵⁰Wolowicz, C. H., "Considerations in the Determination of Stability and Control Derivatives and Dynamic Characteristics from Flight Data," AGARD Report 549, Part 1, 1966.
- ⁵¹Rynaski, E. G., "Application of Advanced Identification Techniques to Nonlinear Equations of Motion," *Proceedings of the Stall/Post-Stall/Spin Symposium*, Wright-Patterson AFB, OH, Dec. 1971, pp. 0.1–0.18.
- ⁵²Iliff, K. W., "Parameter Estimation for Flight Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 5, 1989, pp. 609–622.
- ⁵³Åström, K. J. and Bohlin, T., "Numerical Identification of Linear Dynamic Systems from Normal Operating Records," *Proceedings of the 2nd IFAC Symposium on Theory of Self-Adaptive Control Systems*, England, 1965, edited by P. H. Hammond, Plenum Press, New York, 1966, pp. 96–111.
- ⁵⁴Taylor L. W. Jr. and Iliff, K. W., "A modified Newton–Raphson Method for Determining Stability Derivatives from Flight Data," *Computing Methods in Optimization Problems*, edited by L. A. Zadeh, L. W. Neustadt, and A. V. Balakrishnan, Academic Press, New York, 1969, pp. 353–364.
- ⁵⁵Mehra, R. K., "Identification of Stochastic Linear Dynamic Systems Using Kalman Filter Representation," *AIAA Journal*, Vol. 9, Jan. 1971, pp. 28–31.
- ⁵⁶Gerlach, O. H., "Determination of Performance, Stability and Control Characteristics from Measurements in Non-Steady Manoeuvres," AGARD CP-17, Sept. 1966, pp. 499–523.
- ⁵⁷Jategaonkar, R. V. (guest ed.), Special Section: Flight Vehicle System ID—Part 1, *Journal of Aircraft*, Vol. 41, No. 4, 2004, pp. 681–764.
- ⁵⁸Jategaonkar, R. V., (guest ed.), Special Section: Flight Vehicle System ID—Part 2, *Journal of Aircraft*, Vol. 42, No. 1, 2005, pp. 11–92.

INTRODUCTION

23

- ⁵⁹Klein, V., "Estimation of Aircraft Aerodynamic Parameters from Flight Data," *Progress in Aerospace Sciences*, Vol. 26, No. 1, 1989, pp. 1–77.
- ⁶⁰Padfield, G. D. (guest ed.), "Applications of System Identification in Rotorcraft Flight Dynamics," *Vertica*, Vol. 13, No. 3, 1989, pp. 207–412.
- ⁶¹Hamel, P. G. and Kaletka, J., "Advances in Rotorcraft System Identification," *Progress in Aerospace Sciences*, Vol. 33, No. 3/4, 1997, pp. 259–284.
- ⁶²Maine, R. E. and Iliff, K. W., "Identification of Dynamic Systems," AGARD AG-300, Vol. 2, Jan. 1985.
- ⁶³Maine, R. E. and Iliff, K. W., "Identification of Dynamic Systems—Applications to Aircraft. Part 1: The Output Error Approach," AGARD AG-300, Vol. 3, Part 1, Dec. 1986.
- ⁶⁴Mulder, J. A., Sridhar, J. K., and Breeman, J. H., "Identification of Dynamic Systems—Applications to Aircraft, Part 2: Nonlinear Analysis and Manoeuvre Design," AGARD AG-300, Vol. 3, Part 2, May 1994.
- ⁶⁵Hamel, P. G. (ed.), "Rotorcraft System Identification," AGARD AR-280, Sept. 1991.
- ⁶⁶Stoliker, F. N. (ed.), "Introduction to Flight Test Engineering," AGARD AG-300, Vol. 14, Sept. 1995.
- ⁶⁷AGARD Flight Test Instrumentation Series, AGARD AG-160, Vols 1–19, 1971–1991.
- ⁶⁸A'Harrah, R. C., "Technical Evaluation Report," on "Challenges in Dynamics, System Identification, Control and Handling Qualities for Land, Air, Sea and Space Vehicles," RTO-MP-095, Jan. 2003, pp. T-1–T-4.
- ⁶⁹Eykhoff, P., *System Identification, Parameter and State Estimation*, John Wiley & Sons, New York, 1974.
- ⁷⁰Goodwin, G. C. and Payne, R. L., *Dynamic System Identification: Experiment Design and Data Analysis*, Academic Press, New York, 1977.
- ⁷¹Sorenson, H. W., *Parameter Estimation—Principles and Problems*, Marcel Dekker, New York, 1980.
- ⁷²Ljung, L. and Söderström, T. S., *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA, 1983.
- ⁷³Etkin, B., *Dynamics of Atmospheric Flight*, John Wiley & Sons, New York, 1972.
- ⁷⁴McRuer, D., Ashkenas, I., and Graham, D., *Aircraft Dynamics and Automatic Control*, Princeton University Press, Princeton, NJ, 1973.
- ⁷⁵"MATLAB, The Language of Technical Computing, Version 6.5.1, Release 13 (Service Pack 1)," The MathWorks Inc., Natick, MA, 2003.

This page intentionally left blank

Chapter 2

Data Gathering

I. Introduction

THE PROCESS of performing experiments, and recording system inputs and outputs, constitutes data gathering. Besides the ability to efficiently extract unknown system parameters applying advanced estimation techniques, which we will cover in the following chapters, data gathering is the other crucial aspect of flight vehicle system identification, because the basic rule “If it is not in the data, it cannot be modeled” applies to all exercises that attempt parameter estimation from experimental data. This is true irrespective of the type of flight vehicle we may attempt to model. Gathered data basically limits, both in terms of scope and accuracy, the model development and parameter estimation. Although data acquisition-related aspects like parameters (aircraft motion variables and control surface deflections) to be recorded, quality of sensors in terms of accuracy and noise, sampling rate, signal conditioners, and data recording equipment play a role in the overall process of data analysis, the most important aspects of data gathering are to 1) define the scope of the flight testing, 2) define a suitable sequence of flight maneuvers to be performed at each test point, and 3) choose an adequate form of the inputs to excite the aircraft motion in some optimum sense. The last topic is commonly called in the literature “optimal input design.”

The accuracy and reliability of parameter estimates, obtained by applying either the recent modern methods like maximum likelihood or past methods, elaborated in Chapter 1, Secs. VII and VI, respectively, depend heavily on the amount of information available in the vehicle response, which can be maximized through suitable excitations. This fact was recognized at an early stage, as evident from Milliken’s statement in 1951:¹

It would appear that the optimum input in a given case is that which best excites the frequency range of interest, and hence the harmonic content of the input should be examined before the test to ensure that it is suitable.

Throughout the history of flight vehicle system identification this has been, in essence, the guiding principle in designing a proper flight test maneuver for extracting aerodynamic parameters.² The early development in the mid 1940s and early 1950s led to heuristic input signals such as continuous sinusoidal input or pulse input. These designs were mainly governed by the method of

analysis applied to extract stability and control derivatives from flight data. In the general field of system identification, theoretical developments on input design started during the 1960s.^{3–5} Around the same time, optimal design for flight vehicle system identification started with the work of Gerlach.^{6,7} During the early 1970s the subject of optimal input designs for aircraft parameter estimation was extensively researched by Mehra, Gupta, Mulder, Koehler, Marchand, Plaetschke et al., both from theoretically motivated design as well as more practical approaches.^{8–13} They led to inputs based on some suitable combinations of harmonic signals or multiple pulses. An overview on optimal input and maneuver design is provided by Mulder et al.¹⁴

Although a lot of effort has been and continues to be spent on optimal input design,^{15–17} the methodology is plagued by a fundamental problem, namely that the design is based on an *a priori* model of the flight vehicle. Better *a priori* models lead to better designs; in other words for a good input design a good model must be known ahead of flight testing. However, in such a case the estimation from flight data would no longer be needed. There is no exact solution to this paradoxical situation; the only way out is to approach the problem pragmatically, and attempt to design an input that is sufficiently broadband in frequency content to cover uncertainties in the *a priori* model and at the same time easy to adapt to changing flight conditions.

In this chapter we discuss the scope and types of flight testing, providing explanation of maneuvers in a nutshell. The rigorous approaches to optimal input design based on theoretical formulations are briefly summarized without going deeply into mathematical detail. A detailed explanation is, however, provided of the basic ideas behind a simple, yet efficient, engineering approach to designing a multistep input. Simple software programs are provided to trace the discussion on multiple step inputs. From the basic knowledge about the natural frequencies of various modes of motion, typical empirical rules to arrive at time steps for multistep inputs are elaborated. From our experience of modeling different types of flight vehicles, we aim to suggest a general procedure for performing dynamic flight maneuvers for system identification purposes. Finally, we take a look at other related aspects of data acquisition like a set of measurements required for estimation of stability and control derivatives, sensors, sampling rate, and signal pre-processing.

II. Flight Testing and Maneuvers

Pertinent to our specific interest, flight testing can be broadly classified into two categories: 1) flight testing for performance evaluation, and 2) flight testing for system identification. The flight testing for aircraft certification usually falls into the first category, whereas that for the aerodynamic database development falls into the second group. Usually a large number of specific maneuvers is required in each case. A part of these maneuvers is common to both, whereas the rest is characteristically different. Flight maneuvers used in the proof-of-match to demonstrate the fidelity of aerodynamic database for flight simulators are defined in the Acceptance Test Guide (ATG). As defined by the FAA (Federal Aviation Administration) and JAA (Joint Aviation Authorities), roughly 100–120 test cases, covering various configurations and maneuvers

including special test cases for military aircraft, are usually needed for this purpose.^{18,19} Although special care is exercised in performing these maneuvers in terms of exactness and to minimize external disturbances, basically these maneuvers fall within the two classes already mentioned. Flight testing and its broad spectrum is a vast subject in itself; a detailed account is found, for example, in Refs. 20–22. Here, it is not our aim to give a consolidated account of these two types of testing; rather, we will take a cursory look at them to understand the basic procedures and differences, providing a brief summary of common flight maneuvers.

A. Flight Testing for Performance Evaluation

Flight testing for performance evaluation aims to determine the ability to fulfill the mission in terms of range, fuel consumption, achievable maximum speed, rate of climb, altitude, and so on. These tests are usually longer duration tests, requiring specific maneuvers such as, for example, 1) acceleration and deceleration, 2) pushover–pullup, 3) wind-up turn, 4) climb/sawtooth climb, 5) bank-to-bank roll, 6) steady sideslip, and 7) landings and takeoffs.

A level flight acceleration–deceleration maneuver is performed as follows: starting from a trimmed horizontal level flight, the aircraft is decelerated by changing the power lever to idle position, and maintaining altitude using the pitch control. Once the minimum speed (just above the stall speed) is reached, the power lever is moved from idle to full throttle position, accelerating the aircraft. Once the maximum speed is reached, the power setting is changed to idle, decelerating to the initial trim speed, at which the test is terminated. In some cases the test is performed for smaller speed variations about the trim condition. The acceleration–deceleration maneuver covers a large range of speed and is usually performed at different altitudes. It allows determination of acceleration capability and longitudinal speed stability; hence it is also called the longitudinal static stability or speed stability test. The speed variations from minimum to maximum result in a large variation in the angle of attack, and as such the acceleration–deceleration maneuver is used for system identification purposes as well. For such purposes, during the acceleration phase, elevator inputs, usually doublers, which we will discuss in the next section, are applied at regular interval to increase the information content in the data.

Pushover–pullups, also called roller-coaster, are primarily performed to determine lift and drag characteristics, longitudinal stability, and the elevator trim requirements. The maneuver is initiated from a trimmed level flight by slowly pushing the stick, reducing the angle-of-attack continuously which leads to a slow diving motion. Once the maximum speed is reached, the stick is pulled slowly to result in increasing angle of attack, which leads to a slow climbing motion. Once the minimum speed (just above the stall speed) is reached, the stick is once again pushed slowly to return to the starting trim condition. In contrast to the acceleration–deceleration maneuver, pushover–pullup maneuvers are performed at constant thrust. This maneuver results in a significant variation in the vertical acceleration, typically covering the range of 0–2 g, and as such is used mostly to determine the drag polars. The maneuver is performed in about 40–50 s.

Wind-up turns are used to determine the gradient of “stick force per g ,” which is a design and certification criterion. It is a measure of the stick force required to change the load factor; it is an indicator of the compromise between the two conflicting requirements of longitudinal stability and maneuverability. A higher slope of stick force vs load factor curve implies higher apparent stability and less maneuverability, because a larger pilot force will be necessary to move away from the trim, and vice versa. The maneuver starts by initiating a level turn, and continuously increasing the bank angle. Increasing bank angle leads to a nose-down motion at increasing speed. Through pitch control the angle of attack is increased to maintain the speed constant. The throttle position is kept fixed. The process of simultaneously increasing the bank angle and angle of attack results in a descending spiraling and wind-up motion with decreasing turn radius, resulting in an increasing load factor. The maneuver is terminated when a g -limit is reached. This is one of the most demanding maneuvers for the pilot, particularly for fighter aircraft with higher g -limits. The duration of a wind-up turn is roughly 60–80 s, and it is repeated at different nominal speeds. Besides the primary purpose of finding the gradient of “stick force per g ,” this maneuver also aids determination of drag and lift coefficients at different angles of attack for a constant Mach number, and also system identification.

The main aim of the climb/sawtooth climb is to determine the best climb rate at different altitudes to arrive at a best climb schedule to reach a desired altitude in a minimum amount of time. Starting from a level flight, repeated climbs and dives are performed to result in predefined variations in the altitude by applying maximum and idle power alternatively. The airspeed is maintained constant using pitch control. This maneuver is rarely used in system identification; however, it is part of the proof-of-match process that is necessary to demonstrate the model fidelity.

Determination of the maximum roll capability is important for fighter aircraft, as it mainly governs the ability to turn quickly and change direction of flight. In the case of transport aircraft adequate roll performance is necessary to quickly compensate for disturbances such as turbulence or asymmetric configurations. Determination of aerodynamic coupling between roll and yaw axes and ability to compensate adverse yaw following an aileron input is part of this test. It is performed in several trim conditions. The bank-to-bank roll, also called rapid roll, tests are widely used for parameter estimation purposes.

The primary aim of steady sideslip maneuvers is to determine the gradient of rudder deflection and aileron deflection per degree of sideslip. These maneuvers are particularly of interest at low speeds for the landing configuration, because larger side slipping may be necessary to compensate for strong crosswind during landing phases. Starting from a horizontal level flight, the maneuver is initiated by applying and holding a constant pedal force, leading to a gradual increase in the angle of sideslip. Constant heading flight is maintained in a banked position by applying ailerons. These maneuvers are widely used in system identification.

The landings and takeoffs are critical flight phases. A careful planning is necessary during these tests, because the performance is greatly affected by wind gradients, particularly during landing phases. The ground effect also affects the aircraft performance at low altitudes. Besides the flybys at different

low altitudes over the runway, landing maneuvers are also used in the estimation of ground effect. Landings under different external weather conditions and with one engine out on takeoff are an important part of the data gathering for simulator validation.

Besides the above-elaborated performance evaluation specific tests, other stability-related maneuvers enabling frequency and damping determination are also performed as a part of overall flight test program. They are primarily used for handling quality evaluations and for estimation of aerodynamic derivatives. Such maneuvers and those commonly used for system identification purpose, for example steady heading sideslips, are elaborated in the next section.

B. Flight Testing for System Identification

Estimation of stability and control derivatives is carried out primarily from the dynamic response of an aircraft to specific control inputs. In general, dynamic motion is excited by applying control pulse, step, multistep, or harmonic inputs. A variety of maneuvers is usually necessary to excite dynamic motion about different axes using independent inputs on every control. Dynamic maneuvers allow extraction of a larger number of static as well as dynamic parameters from a single test. Combining several stationary maneuvers in different trim conditions, principally allows determination of some of the aerodynamic derivatives, for example slope of the lift coefficient or the longitudinal static stability parameter. Compared with dynamic maneuvers, this approach is less efficient because a larger number of test points will be necessary and the uncertainties in the estimates may be larger. Usually the best choice is to combine the dynamic maneuvers and different stationary conditions to extract a single set of aerodynamic derivatives.

A set of common flight maneuvers for system identification consists of 1) short period maneuver, 2) phugoid maneuver, 3) pushover–pullup, 4) level-turn, 5) thrust variation, 6) bank-to-bank roll, 7) Dutch roll maneuver, and 8) steady heading steady sideslip. Besides these maneuvers, as already mentioned, acceleration–deceleration is also used for the purpose at hand. The main source for estimation of ground effect is flybys at different altitudes over the runway and landings. Generally, for control applications, for safety purposes, and for passenger comfort, excitation of the eigenmodes is suppressed. However, for parameter estimation purposes, it is essential that the dynamic response exhibits frequencies and damping of the oscillatory modes. In general, it is recommended to start each maneuver from a trimmed level flight, and allow about 5–10 s of steady flight before applying specific control inputs, and, depending upon the mode of motion, to allow sufficient time after the input to allow the aircraft to oscillate.

The short period motion, which is a fast responding longitudinal mode, provides the most information to enable estimation of derivatives pertaining to the vertical and pitching motion. The short-period maneuver is initiated from a horizontal level flight by applying an elevator or other longitudinal control input, which is usually a doublet or multistep input. Doublet input is a two-sided pulse input, whereas multistep input consists of a suitable combination of more than two pulses. The time step of such inputs is chosen such that the short-period mode is excited well. We will deal with the aspects of arriving at a suitable

form of multistep input and choosing proper time steps for such input in the next section. Input amplitude should be such as to result in about $\pm 3\text{--}4$ deg. variation in angle of attack about the trimmed value, or in terms of the load factor the variation should be typically $\pm 0.4\text{--}0.5$ g. A typical duration of such a maneuver is about 15–20 s. It is performed at different trim speeds. Larger amplitude maneuvers may be necessary to enable estimation of nonlinearities. While applying a multistep elevator input, pilots have to pay attention because in special cases they tend to maintain the angle of attack, particularly during approach; this practice is in contrast to the basic principle of exciting the longitudinal mode, resulting in variation in the angle of attack.

The phugoid mode is a long-duration mode, which can be excited through either an elevator pulse or thrust variation. Typically, one full cycle of phugoid motion is required for parameter estimation purposes. For the validation of databases for simulators, the model accuracy has to be demonstrated for three full cycles.

As already mentioned, the pushover–pullups result in variation in the vertical acceleration and are useful to determine lift and drag characteristics. A level-turn maneuver is initiated from a horizontal level flight by banking the aircraft smoothly into a 30 (45 or 60) deg. bank at a rate of 10 deg./s, holding the bank attitude steady for several seconds, then rolling the aircraft to the opposite direction, again holding the bank angle steady for several seconds, and finally returning to the wings level conditions. Altitude is maintained during the maneuvers being through additional power. It results in an S-type flight path in the horizontal plane, the heading at the start and at the end of the maneuver being roughly the same. As will be elaborated in Chapter 12, Sec. II, this maneuver combined with an elevator doublet input in the banked position is useful to separate out the pitch rate and rate of change of angle of attack components of longitudinal motion.

Dynamic thrust input maneuvers are performed by applying a longer duration doublet input to throttle. They are useful to determine the dynamic effects owing to thrust variations on the longitudinal motion, for example on the downwash effect on the horizontal tail. They also aid determination of the effect of the thrust coefficient on the various aerodynamic derivatives, particularly for propeller aircraft where such effects are dominant. The dependencies of the aerodynamic derivatives on the thrust coefficient can also be determined by performing other longitudinal and lateral-directional dynamic maneuvers starting from trims with different thrust levels.

To enable estimation of lateral-directional derivatives, bank-to-bank roll and Dutch roll maneuvers are necessary. A bank-to-bank maneuver is performed by applying a series of aileron pulses. Starting from a level flight, sharp aileron input is applied, holding it for some time to roll the aircraft from wings-level to at least a 30 deg. bank on one side; this is followed by rapidly changing the input and going smoothly through wings-level to the opposite bank angle, followed by returning to the wings-level condition. The test is repeated for different levels of bank angles (30, 45 and 60 deg.). The bank-to-bank maneuver is also sometimes called the rapid-roll maneuver, because the abrupt changes in aileron pulses result in rapid variation in roll rate and acceleration. A typical duration of such a maneuver is 30–40 s. The doublet or multistep aileron input with short time steps may not be quite adequate, because sufficient time should be allowed for bank angle buildup; rolling motion corresponds to a first-order system.

In many aircraft, ailerons are coupled to roll spoilers, which are additional flaps on the upper surface of the wings. They are drawn out either on the left or the right wing corresponding to the upper aileron deflection, that is, on the wing going down, and help to improve the overall roll efficiency without being too sensitive at small aileron deflections. For aileron deflections less than a threshold value, the spoilers are inactive; they are deflected proportional to aileron deflection larger than the threshold. For a given aircraft, the threshold aileron deflection for roll spoiler activation is a fixed value, which usually ranges from 3 to 5 deg. Owing to this basic construction, ailerons and roll spoilers are deflected proportionately, except for small aileron deflections. Where we are interested in estimating from flight data both aileron and roll-spoiler effectiveness, then it is necessary to perform the bank-to-bank maneuver with small aileron deflection (smaller than the threshold values), during which roll spoilers are inactive, followed by larger aileron input, in which case both control surfaces would contribute to the rolling motion. A similar situation arises for aircraft having aileron-to-rudder interconnection. In this case, automatic rudder deflection following an aileron input is detrimental to extraction of independent derivatives, although both surfaces move. Basically, it amounts to the fundamental aspect of generating independent inputs on every control; otherwise it would not be possible to separate out the effects due to multiple surfaces moving proportionately.

The Dutch roll maneuver provides maximum information on the frequency and damping of this oscillatory mode. It is excited by applying rudder inputs. Usually several cycles of oscillations are recorded; the duration of the total maneuver could be typically 30–35 s or even longer. The resulting maximum peak-to-peak variation in the angle of sideslip is typically of the order of ± 4 deg. or 0.1 g lateral acceleration. The Dutch roll mode of the basic aircraft is usually lightly damped, and can be excited very well with a simple doublet input. Many aircraft are equipped with a yaw damper, which is turned off during maneuvers performed for system identification purposes, because we are mainly interested in the aerodynamic characteristics of the bare airframe. It is required to perform the Dutch roll and bank-to-bank maneuvers at different trim speeds, because most of the lateral-directional derivatives depend on the angle of attack.

Both aileron and rudder input maneuvers are necessary not only to allow estimation of the respective control surface effectiveness, but also to improve estimation of other derivatives due to roll and yaw rates. The aileron input maneuver yields significantly larger roll rates which are characteristically different in shape compared with that from the rudder input. In general, it is our experience that better estimation results are obtained by performing the aileron and rudder input maneuvers separately, starting with a bank-to-bank maneuver followed by a Dutch roll. This sequence is easy for pilots to execute manually. In some cases we have also applied computerized aileron and rudder inputs simultaneously. In such cases, the starting points of the two inputs should be time skewed, that is, displaced in time. If the starting point is time-synchronized, the duration of the two inputs should be different, or an independent maneuver with a single control surface input would be desirable to improve separation of the components due to these two control surfaces.

The steady heading steady sideslip (SHSS) maneuver provides additional information on the directional stability, lateral-directional control, and cross coupling effects. It is usually flown in both the positive and negative side-slipping conditions to allow detection of any asymmetric effects. Such tests are particularly important for landing configurations. SHSS are performed in different variations to cover a large range in angle of sideslips in both directions and at different angles of attack and thrust levels. For example, starting from a horizontal level flight, SHSS is performed by applying abrupt pedal step, and maintaining the sideslip angle at some chosen level for roughly 10–15 s, followed by releasing the pedal to a neutral position, allowing few seconds for dynamic recovery response. This is followed by repeating the procedure in the other direction of side slipping. In each case, the track angle is maintained by adjusting the bank angle applying aileron input. The side slipping conditions should be achieved rapidly, ideally as shown schematically in Fig. 2.1a. However, in practice some overshoot is unavoidable, and strong excitation of the Dutch roll is to be suppressed by regulating the pedal input. For larger sideslips, the procedure is similar. The alternative is to build up the larger sideslip angles in two steps, before releasing the controls to neutral. Yet another variation is to change abruptly from one steady sideslip in one direction to the other direction, without a stabilizing phase for no side slipping conditions, as shown in the two plots on the right side of Fig. 2.1a. If SHSS are carried out for performance evaluations, then the overall procedure is similar, but for maintaining the steady side slipping conditions for a considerably longer time than indicated above.

Besides SHSS with abrupt changes in asymmetric trim conditions, a quasi-steady type of maneuver, called wings-level steady sideslip, with gradual variation in side slipping is also useful. Starting from horizontal level flight, rudder is applied gradually to increase the angle of sideslip from zero to some chosen value (say 5 or 10 deg.), then gradually releasing the pedal pressure to reduce

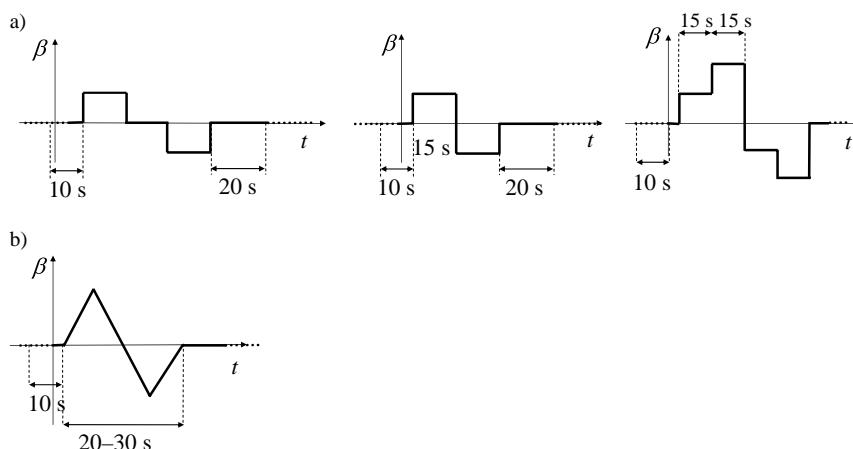


Fig. 2.1 Schematic of steady heading sideslip and sweep input maneuvers. a) Steady heading sideslip, b) beta sweep.

the side slipping, crossing over through zero to angle of sideslip in the opposite direction, finally returning to neutral, see Fig. 2.1b. Throughout the maneuver the wings-level conditions are maintained by applying lateral-stick cross-control. In steady-heading sideslip as well as wings-level steady sideslip maneuvers, the variations in speed are compensated for through the longitudinal control.

Precautions and constant monitoring are necessary during large constant side-slipping conditions owing to large aerodynamic loads acting on the vertical tail, and to make sure that the structural limits are not exceeded. The piloting techniques to perform SHSS are demanding, because considerable power is necessary for rudder deflections required to achieve asymmetric trim conditions. New safety recommendations have been defined to avoid structural damage resulting from large rudder deflections for transport-category airplanes.²³

Depending upon the aircraft, if there are additional controls available, for example direct-lift-control flaps or an all-moving trimmable horizontal tail, then maneuvers with appropriate inputs (either multistep or longer duration pulses) will be necessary. In the case of unstable aircraft, the flight control laws prevent dynamic excitation of the aircraft motion, and result in highly correlated control surface deflections and aircraft motion. Special techniques are necessary to generate flight data suitable for parameter estimation; we defer this specific case to Chapter 9, dealing exclusively with unstable aircraft identification.

The frequency sweep test techniques, although rarely used for fixed-wing aircraft, are used more routinely in the field of rotorcraft identification.²⁴ These techniques are useful and necessary, not only for the next generation specification requirements,²⁵ but also as an integral part of the interdisciplinary modeling aspects. Critical flight incidences have, however, occurred while sweep testing owing to, for example, exceeding the aeroservoelastic range or the flight permissible maximum loads.²⁶ Proper coordination is, hence, necessary through careful preparation, buildup, real-time monitoring, and analysis to prevent possible structural damage and to avoid any increase in the risk factor.

The central idea behind system identification maneuvers described here is to excite pertinent modes of aircraft motion independently and sufficiently. In general, while exciting a particular mode, excitation of the other modes is to be minimized. The amplitudes are usually based on heuristic considerations, which have been discussed. Thus, the task now boils down to finding suitable shapes and durations of inputs that determine the frequency content.

III. Optimal Input Design

There are two approaches that have been applied in the past to design optimal inputs. The first, based on the estimation error criterion, is more rigorous, involved, and tends to be theoretical. The second is an engineering approach, based on the spectral behavior of the model.

A. Input Design by Estimation Error Analysis

As described in Chapter 4, statistical properties of the estimates are most conveniently expressed in terms of the bias and covariances of the estimates. It will

be shown in the same chapter that the maximum likelihood estimation is based on the maximization $p(z|\Theta)$, the conditional probability density function of measurements z for a given parameter vector Θ , and that the Fisher information matrix which is an indicator of the information content in the data being analyzed is given by

$$\mathcal{F}_{ij} = E \left\{ -\frac{\partial^2 L(z|\Theta)}{\partial \Theta_i \partial \Theta_j} \right\} \quad (2.1)$$

where $L(z|\Theta) = \ln[p(z|\Theta)]$, and $E\{ \}$ denotes the expected value. It will also be shown that the maximum likelihood estimation is bias-free and efficient in a statistical sense. For further development of the input signals, we simply accept these properties at this point. Under such conditions, the inverse of the Fisher information matrix given by Eq. (2.1) provides a good approximation to P , the parameter error covariance matrix.^{27,28} Without going into the exact mathematical derivation which will follow in the subsequent chapters, we also draw on the expression for the information matrix \mathcal{F} in terms of $\partial y / \partial \Theta$, the gradient of the system responses with respect to the parameters.

$$P \approx \mathcal{F}^{-1} \approx \left\{ \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \left[\frac{\partial y(t_k)}{\partial \Theta} \right] \right\}^{-1} \quad (2.2)$$

Thus, the parameter error covariance matrix P depends upon the response sensitivities to the model parameters $\partial y / \partial \Theta$, the data points being analyzed N , and a weighting matrix R , which depends on the measurement noise. The system responses y and its gradients $\partial y / \partial \Theta$ are obtained from the a priori model used for designing the inputs. At this stage the parameters Θ are known and kept fixed at their a priori values. Thus, the information content in \mathcal{F} will be mainly determined by the response gradients which depend on the input excitation. Proper tuning of the input shapes will result in an optimal excitation of the modes of a system defined by the a priori model. In other words, it leads to maximization of the information matrix. In turn, this is the same as minimizing the error covariance matrix P , which suggests that the parameters which will be subsequently estimated from such inputs will have lowest statistical errors and hence will be more accurate. This is the basic idea behind designing the optimal inputs based on the theoretical approach.

Different measures, all based on the Fisher information matrix, can be chosen as criteria to design optimal inputs.^{11,28} The first criterion is:

$$\min_{u(t)} \{ \det(P) \} \implies u_{\text{opt}}(t) \quad (2.3)$$

which is the same as maximizing the determinant of the Fisher information matrix \mathcal{F} . The criterion of Eq. (2.3) is the overall measure in terms of the volume of the estimation error ellipsoid, which is proportional to the determinant of P . Input signals that maximize the determinant of \mathcal{F} (or equivalently

minimize the determinant of P) are commonly called in the literature D-optimal, where “D” stands for “determinant.” It can be shown that the optimum is invariant to scaling of the system states. Let us take a look at the physical significance of D-optimality. The columns of the information matrix \mathcal{F} represent contributions of the individual parameters. If the columns are independent, the determinant of the matrix will have a maximum value subject to the given constraints in terms of the postulated model. On the other hand, if parameters are correlated, the determinant will be smaller; in the worst case, linear dependency of two parameters results in a zero determinant, indicating that the two parameters cannot be estimated independently. Thus, D-optimality minimizes the redundancy and leads to better identifiable parameters.

The other criteria are the sum and product of the diagonal elements of the matrix P :

$$\min_{u(t)} \{tr(P)\} \implies u_{\text{opt}}(t) \quad (2.4)$$

and

$$\min_{u(t)} \left\{ \prod_i P_{ii} \right\} \implies u_{\text{opt}}(t) \quad (2.5)$$

The diagonal elements of P represent the variances of the estimates, and are indicators of the accuracy of the estimates. As discussed in Chapters 4 and 11, the standard deviations are given by the square root of the variances. Thus, the last two criteria attempt to minimize the standard deviations, or in other words increase the accuracy of the estimates. Input signals resulting from Eq. (2.4), i.e., by minimizing the sum of the variances of the parameter estimation errors, are commonly called in the literature A-optimal, where ‘A’ stands for “average.” In contrast to the D-optimal design, the A-optimality criterion of Eq. (2.4) or (2.5) is strongly dependent on the units and is not scale-invariant. This aspect needs to be carefully considered when designing A-optimal inputs.

Both time domain and frequency domain methods have been applied to solve the above optimal design problem. To apply any of these criteria, it is necessary to define an input space (in terms of forms and amplitudes) to be searched for the optimal input. Different researchers have followed different approaches. The best known two designs (shown in Fig. 2.2) are¹³ 1) Mehra input, based on optimization of harmonic signals selecting integer multiples of the basic frequency, where the basic frequency corresponds to the observation time and 2) Delft University of Technology (DUT) input, based on the sum of sine functions. Both these methods and other similar approaches lead to continuous and smoothly varying inputs. These signals are tuned to the a priori model used in their design, but they are not quite suitable to be flown manually. In

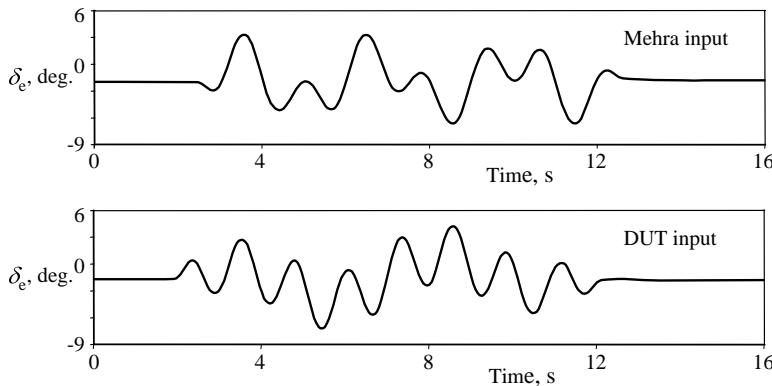


Fig. 2.2 Time histories of typical harmonic inputs.

all these cases, an onboard computer implementation is necessary to generate such harmonic inputs.

B. Design of Multistep Input Signals

The design of multistep input signals is a two-step procedure, first investigating the range of frequencies needed for accurate estimation of parameters, and second designing suitable multiple step input to cover the desired frequency range. As in the previous case, the design procedure requires an a priori model of the aircraft.

Investigation of the range of frequencies for accurate estimation of parameters is based on synthesizing the contributions due to each of the parameters appearing in each of the force and moment equations. This is conveniently done using the Bode diagram, which helps to determine which of the frequencies must be included in the input signal to enable extraction of particular derivatives. The following procedure developed by Marchand^{10,29} is briefly illustrated here considering the following linearized model pertaining to the longitudinal motion:

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & X_q & -g \\ Z_u/U_0 & Z_\alpha/U_0 & 1 & 0 \\ M_u & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_{\delta e} \\ Z_{\delta e} \\ M_{\delta e} \\ 0 \end{bmatrix} \delta_e \quad (2.6)$$

where U_0 is nominal trim speed, u the component of true airspeed along the X -axis, α the angle of attack, q the pitch rate, θ the pitch attitude, δ_e the elevator input and $(X_{(\cdot)}, Z_{(\cdot)}, M_{(\cdot)})$ the dimensional derivatives. For the case considered here for demonstration, the a priori values of the dimensional derivatives for

a forward speed of 98 m/s lead to the system state and control matrices in Eq. (2.6), which may be denoted as A and B , given by

$$A = \begin{bmatrix} -0.0091 & 9.43 & 0 & -9.80665 \\ -0.0022 & -0.867 & 1 & 0 \\ 0 & -3.49 & -2.04 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B = [0 \ 0 \ -0.11 \ -5.09]^T$$

Now, for each equation in Eq. (2.6), the frequency response magnitudes of the various terms in that equation are plotted as a function of the input signal frequency. To demonstrate the procedure, we consider here the pitching moment equation. We compute the frequency response magnitudes for each of the terms, namely M_u , M_α , M_q , and M_{δ_e} as well as for the total pitch acceleration \dot{q} with respect to the elevator input δ_e . In other words, it amounts to computing the magnitudes:

$$\left| \frac{\tilde{q}(\omega)}{\tilde{\delta}_e(\omega)} \right|, \left| \frac{M_u \tilde{u}(\omega)}{\tilde{\delta}_e(\omega)} \right|, \left| \frac{M_\alpha \tilde{\alpha}(\omega)}{\tilde{\delta}_e(\omega)} \right|, \left| \frac{M_q \tilde{q}(\omega)}{\tilde{\delta}_e(\omega)} \right|, \left| \frac{M_{\delta_e} \tilde{\delta}_e(\omega)}{\tilde{\delta}_e(\omega)} \right| \quad (2.7)$$

where \sim denotes the Fourier transform. The individual components for \dot{q} , α , q , and δ_e can be computed from the output equation $y = C[u \ \alpha \ q \ \theta]^T + D\delta_e$ by defining the observation matrices C and D appropriately as follows: 1) $C_{\dot{q}} = [0 \ -3.49 \ -2.04 \ 0]$, $D_{\dot{q}} = [-5.09]$; 2) $C_\alpha = [0 \ -3.49 \ 0 \ 0]$, $D_\alpha = [0]$; 3) $C_q = [0 \ 0 \ -2.04 \ 0]$, $D_q = [0]$; and 4) $C_{\delta_e} = [0 \ 0 \ 0 \ 0]$, $D_{\delta_e} = [-5.09]$, where the subscripts denote matrices corresponding to the respective variables. These components are computed using the program “/FVSysID/chapter02/FreqCompInput.m” and plotted in Fig. 2.3. The contribution due to the term M_u is not shown, because the a priori value of this derivative is zero. The magnitude of the control term corresponding to M_{δ_e} with respect to δ_e is largest compared with the other terms in the higher frequency region. This suggests that the control effectiveness is best identifiable at such frequencies. The identifiability of other parameters depends on the frequency range, as will be discussed next. The procedure elaborated above evaluates the contributions due to control surface deflections, flow angles, velocity components and angular rates, but does not include the indirect influence of attitude angles which are used in the output error method, which we will study in detail subsequently. In Sec. V it has been demonstrated that attitude angles are also useful in the estimation of aerodynamic models.

The procedure is repeated for the X- and Z-equations for the longitudinal and vertical motion. From such plots, the identifiability of the derivatives can be ascertained. At any given frequency, a large magnitude of any particular term compared with the other contributions suggests a dominant influence of that derivative, which indicates good information content necessary for estimation

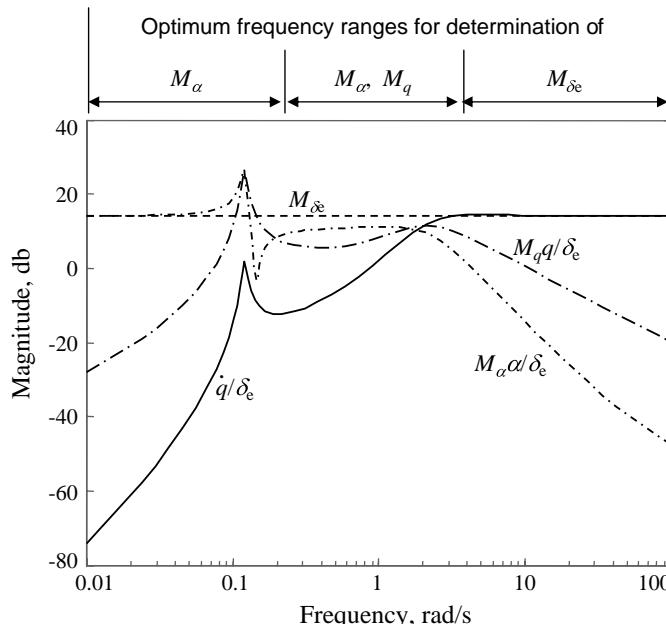


Fig. 2.3 Bode magnitude plot of the pitching moment equation terms.

of the parameter. Conversely, the derivative cannot be accurately estimated, if the contribution is small. It is now necessary to specify a measure to decide upon relative contributions. As a rule of thumb, a derivative is considered identifiable when its term has a magnitude of at least 10% of the largest term's magnitude.¹³

From frequency response magnitude plots for the X- and Z-forces, and for the pitching moment, the ranges of frequencies are determined in which contributions owing to each derivative are appreciable in the sense just described. These ranges are plotted by horizontal lines in Fig. 2.4, which shows that all the derivatives can be identified from excitation in the two regions marked in gray. The solid lines indicate regions where the derivatives are directly identifiable and the dashed line the regions in which they can be estimated only as the ratio of two parameters. The regions surround the natural frequencies of the phugoid and the short period mode. This is not really surprising, because from our basic intuition, we know that the system excited at its natural mode will exhibit dominant dynamic motion. Excitation of the long period phugoid mode is easily obtained through a longer duration pulse input, but the excitation of the fast responding short period motion is more critical. The above analysis combined with additional issues provides a good basis for designing elevator input signals for longitudinal motion, namely 1) the optimum range for frequencies covers a range below and above the natural frequency of the short period mode, 2) the eigenfrequencies of the modes based on the a priori model are

DATA GATHERING

39

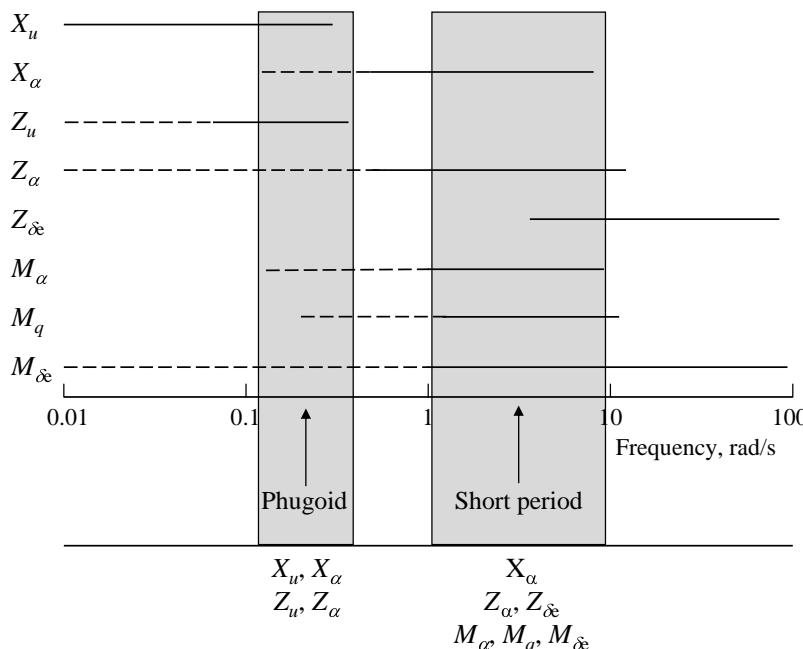


Fig. 2.4 Regions of identifiability.

subjected to uncertainties, and 3) the eigenfrequency changes with the flight condition. These considerations imply that besides the eigenfrequency, it is also necessary to excite the frequencies around the eigenfrequency as well. Based on these considerations, a bandwidth of 1:10 is recommended for the design of a multistep input signal. Thus, the task now is to design an input signal covering this required frequency range.

A multistep input signal of arbitrary shape can be synthesized using a suitable sequential combination of pulse inputs. For an input signal consisting of an arbitrary number of equidistant time steps, in any combination of input levels, as derived in Appendix A, the power (or energy) spectrum is given by³⁰

$$E(\omega) = 2\Delta t^2 \frac{1 - \cos \Omega}{\Omega^2} \left[\sum_{i=1}^N V_i^2 + 2 \sum_{j=1}^{N-1} \cos j\Omega \sum_{i=1}^{N-j} V_i V_{i+j} \right] \quad (2.8)$$

where $\Omega = \omega \Delta t$ is the normalized frequency, $T = N \Delta t$ the total duration of the input consisting of N impulses each of duration Δt , and V_i the amplitude for the current impulse ($t_{i-1} < t < t_i$). Equation (2.8) has been coded in the function “/FVSysID/chapter02/EnergySpectrum.m,” which is used in the following to compute the spectra of various input signals. It is also possible to compute the spectra by fast Fourier transform (FFT) using the function “InputSigPSD.m,” which however requires the signal processing toolbox of Matlab®.

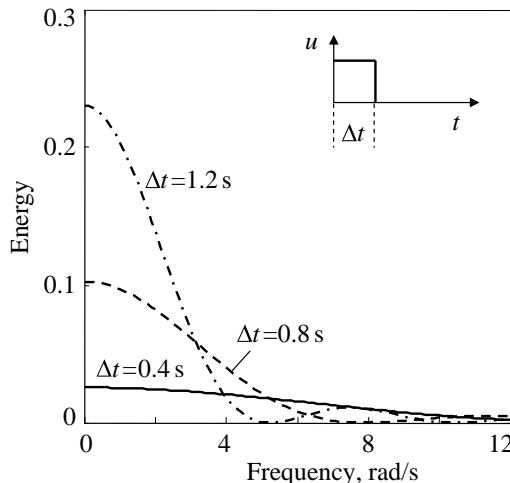


Fig. 2.5 Energy spectra of pulse inputs.

The simplest input to excite the oscillatory motion is a pulse input. After the input over a specified time the control is released to allow the aircraft to oscillate freely without pilot inputs. Equation (2.8) leads to the energy spectra shown in Fig. 2.5 for such an input corresponding to different durations of Δt , but with the same input amplitude. The spectra in Fig. 2.5 are computed and plotted using the program “/FVSysID/chapter02/EnergyImpulse.m.” From this figure, we observe that a pulse with $\Delta t = 0.4 \text{ s}$ has energy spread over a larger range of frequencies, but the magnitude is very small, which is not enough to excite the system adequately. Increasing the duration leads to larger energy content in the signal, but at the same time decreasing the spread to lower frequencies. For $\Delta t = 1.2 \text{ s}$, it has large energy, but unfortunately concentrated at low frequencies with rapid decay. The pulse input is asymmetric about the starting trim values, which leads to nonzero energy at zero frequency. The asymmetric excitation can lead to changes in the stationary conditions, in many cases deviating significantly compared with the starting trim conditions from which the maneuver might have been initiated. This in itself is not a serious drawback, provided we can account for such changes in the identified model, which may turn out to be flight condition-dependent or nonlinear. If linear models are to be identified, it may lead to violation of the range of validity. The major limitation of this input is that it is less suitable for exciting the rapid eigenmodes at higher frequencies, like short period or Dutch roll.

A more commonly used input to excite the dynamic motion is a doublet: control moved abruptly in one direction, held fixed for certain time Δt , then moved abruptly to other direction, held fixed for same specified time before releasing it to neutral. The doublet input, which is a two-sided pulse, results in a symmetrical signal, having energy concentrated at a frequency which varies with Δt . Figure 2.6 shows the energy spectra for three different time steps

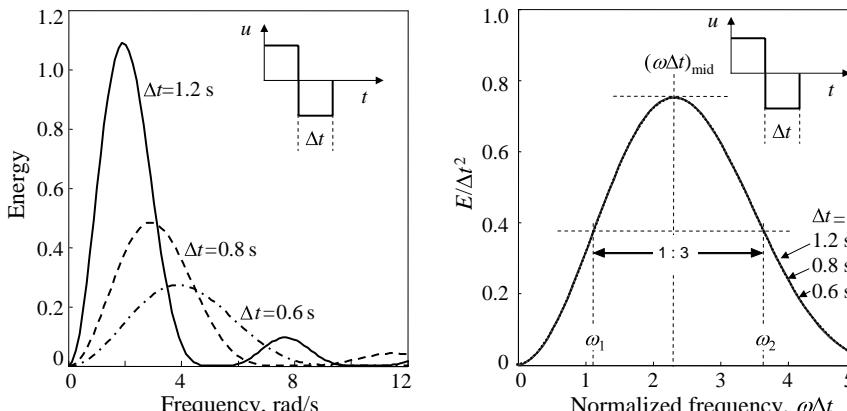


Fig. 2.6 Energy spectra of doublet inputs.

$\Delta t = 0.6, 0.8$, and 1.2 s, which are computed and plotted using the program “/FVSysID/chapter02/EnergyDoublet.m.” The drop in the energy content around the peak is relatively fast. As observed on the right-hand side of Fig. 2.6, the normalized frequency $\omega\Delta t$ for the peak is roughly 2.3, that is,

$$\omega_n \Delta t_{DBLT} \approx 2.3 \quad (2.9)$$

where ω_n is the short period frequency. Equation (2.9) can be expressed in terms of the period of oscillation, yielding an estimate of the time step for the doublet given by:

$$\Delta t_{DBLT} \approx \frac{2.3}{\omega_n} \approx \frac{2\pi}{2.7\omega_n} \approx \frac{1}{2.7} \cdot \text{period of oscillation} \quad (2.10)$$

Although Eq. (2.10) shows a factor of $1/2.7$, in practice, a simpler factor of $1/2$ is often used, which corresponds to choosing the total length of the doublet ($=2\Delta t$) equal to the period of the eigenmotion. Through this very simple rule of thumb, the corresponding mode based on the a priori model can be easily and adequately excited. It is observed from Fig. 2.6 that the doublet input has a bandwidth of roughly 1:3, which is the ratio of the normalized frequencies ω_2 (~ 3.63) to ω_1 (~ 1.1 , cf. Fig. 2.6). In this range the corresponding spectrum is at least half its maximum value. Owing to its simplicity the doublet input is widely used for estimation of stability and control derivatives. A combination of doublet and pulse inputs is also quite useful in some cases; a somewhat smaller step size than given by Eq. (2.10) leads to a slightly higher central frequency of the doublet, which is followed by a pulse to cover the lower frequency range.

Extending the logic that progressing from single-step input to two-step input (doublet) leads to a spread of the power spectrum, a much broader band signal

can be achieved through a multistep input. Without going into exact details, a multistep input can be designed by minimizing the deviation of the computed spectrum for a design from the desired spectrum suitably defined from the results of frequency response magnitude plots (Fig. 2.3). In such cases it is desirable to minimize the variations in the energy contents in the range of frequencies covered, with the drop on the two sides being smooth and rapid, and ideally zero or very small energy at zero frequency. The parameters to be adjusted during the minimization are the number of time steps N in Eq. (2.8), and amplitudes at each step. Design of such a multistep input was performed by Koehler, resulting in the widely known 3–2–1–1 input.¹² The 3–2–1–1 input is $7\Delta t$ long and consists of alternating positive and negative, equal-amplitude, steps of relative duration 3, 2, 1, and 1. The energy spectrum of such a 3–2–1–1 input, computed from Eq. (2.8) using the program “/FVSSysID/chapter02/EnergyImpDbtS3211M3211.m,” is shown in Fig. 2.7.

The time step Δt for the 3–2–1–1 input is usually chosen such that the natural frequency of the mode being excited lies in the center or in the upper third of the input spectrum. As seen from Fig. 2.7, the normalized frequencies $\omega\Delta t$ for these two conditions are roughly $(\omega\Delta t)_{\text{mid}} = 1.6$ and $(\omega\Delta t)_{2/3} = 2.1$ respectively. This leads to an estimate for Δt_{3211} given by:

$$\Delta t_{3211} \approx \frac{1.6}{\omega_n} \approx \frac{\pi}{2\omega_n} \approx \frac{1}{4} \cdot \text{period of oscillation} \quad (2.11)$$

or

$$\Delta t_{3211} \approx \frac{2.1}{\omega_n} \approx \frac{1}{3} \cdot \text{period of oscillation} \quad (2.12)$$

Thus, the time step Δt_{3211} is somewhat smaller than Δt_{DBLT} . As a rule of thumb, a time step Δt_{3211} given by $\Delta t \approx 0.3/f_c$ is quite effective in several aircraft cases,

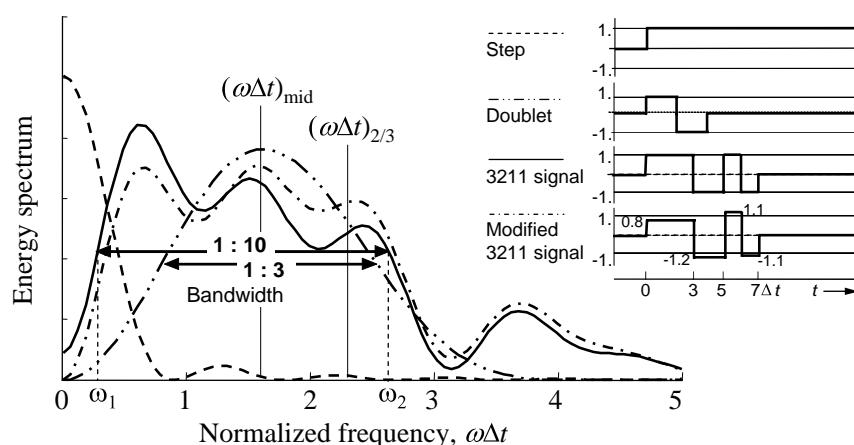


Fig. 2.7 Frequency domain comparison of standard inputs.

where f_C in Hertz is the frequency of the mode to be excited. This matches well with the range 0.25–0.35 based on an empirical rule and numerical studies.³¹

The spectra for various signals are computed with suitable time steps using the program “/FVsysID/chapter02/EnergyImpDbtS3211M3211.m”; for comparison purposes they are plotted in Fig. 2.7 as a function of normalized frequency of the 3–2–1–1 signal. Assuming a time step of $\Delta t_{3211} = 1$ s for the 3–2–1–1 signal, the equivalent time step for the doublet is computed from Eqs. (2.11) and (2.9), and turns out to be roughly 1.44 s ($\Delta t_{DBLT} \approx 2.3/\omega_n$; $\omega_n \approx 1.6/\Delta t_{3211}$). It is observed from Fig. 2.7 that the 3–2–1–1 input has a bandwidth of roughly 1:10 ($=\omega_2/\omega_1 \approx 2.7/0.3$), having at least half of its maximum energy. Thus, the 3–2–1–1 input has a much wider spectrum compared with that of the impulse or doublet inputs. For reference purposes, another equivalent program based on FFT, “PSDImpDbtS3211M3211.m,” is provided to compute the various spectra.

Since the natural frequency of the mode is flight condition-dependent, strictly speaking the optimal Δt varies from flight condition to flight condition. However, the time variations and step reversals in the 3–2–1–1 signal result in a broad frequency content, and in spite of some deviations from the optimal value of Δt , the input is adequate to excite a band around the natural frequency. On the other hand, tuning of Δt for a doublet input is a little more critical, because it has a smaller spread of frequency content around the central frequency. It is primarily for this reason that the 3–2–1–1 signal is superior to the doublet input.

The main advantage of the 3–2–1–1 input lies in its simplicity and the ability to execute it manually. It can be fairly easily remembered without any external aid. Different procedures have been adopted in practice to apply the 3–2–1–1 input manually, matching the desired Δt or for repeatability. For example, just counting “twenty-one, twenty-two, twenty-three” to keep the stick pushed, then “twenty-one, twenty-two” to keep it pulled, then “twenty-one” to push, and finally once again “twenty-one” in the pulled position, and each time changing the stick position abruptly, has produced remarkably good quality 3–2–1–1 inputs. Another procedure is to provide audio-cuing, consisting of series of beeps. Optical cuing has also been used in a few cases, by providing on the display the multistep input form which the pilot tries to track.

Two minor aspects of 3–2–1–1 inputs are that 1) they are asymmetric about the trim deflection (with four time steps in one direction and three time steps in the other), and as a consequence, they have nonzero energy at zero frequency, and 2) the first step being of larger duration, namely three units of Δt , they may lead to motions far from the initial trim condition before the following steps are applied. These undesirable effects can be minimized by modifying the input amplitudes of the 3–2–1–1 input; such a modified 3–2–1–1 input is shown in Fig. 2.7. It has much reduced energy at zero frequency and the variations in the spectrum are much less. Precise manual application of the modified 3–2–1–1 is, however, difficult and calls for onboard computer implementation. Another way is to apply 3–2–1–1 twice, and first starting by pulling up the aircraft and the second time by pushing the stick.

Another variation of the 3–2–1–1 input is the 1–1–2–3 input obtained by time twisting the steps. The 1–1–2–3 input first excites the higher frequencies

at the trimmed condition and prevents the vehicle from going far from the trim condition, before the larger duration time steps are applied. Like the 3–2–1–1, the 1–1–2–3 input can be easily applied manually with little practice. The 1–1–2–3 has the same energy spectrum as the 3–2–1–1 input. In some cases a 1–2–1 input, which is a combination of a normal doublet followed by an inverted doublet, is used. By using a time step for the 1–2–1 signal that is half of that of the doublet, it becomes possible to shift the spectrum to higher frequencies. This can be easily verified using the sample program “/FVSysID/chapter02/EnergyDbt121.m.” Much more complex multistep maneuvers have also been designed in some cases, which of course cannot be flown manually.^{15,17} In general, for more complex multistep inputs, the time step has to be chosen carefully, otherwise there are pronounced variations with valleys in the input spectrum.

Following a similar procedure based on analyzing the frequency response magnitude plots and approximations given in Eq. (2.10) or (2.11), we can also design a 3–2–1–1 input for Dutch roll excitation. However, the Dutch roll is lightly damped compared with the short period, and hence a doublet input is found adequate to excite the Dutch roll adequately. Furthermore, although the time step of a doublet can be optimized based on a priori knowledge, mostly the duration as well as the exact shape is of secondary importance. This is evident from the typical Dutch roll responses shown in Fig. 2.8 for clean configuration at four different speeds. The durations of the oscillations as measured from the recorded data and the actual time steps of the inputs are also provided in the same figure. Based on the very good results obtained for several aircraft cases, we recommend a doublet input on the rudder to excite the Dutch roll motion and 3–2–1–1 or 1–1–2–3 on the elevator to excite the short period motion.

A detailed investigation based on flight data,³² comparing five different types of inputs, has shown that the simpler 3–2–1–1 or doublet inputs yield comparable and in many cases even better estimates compared with the optimal harmonic inputs such as Mehra or DUT elaborated in Sec. III.A. This is mainly because the multistep inputs are richer in frequency contents than the harmonic inputs.

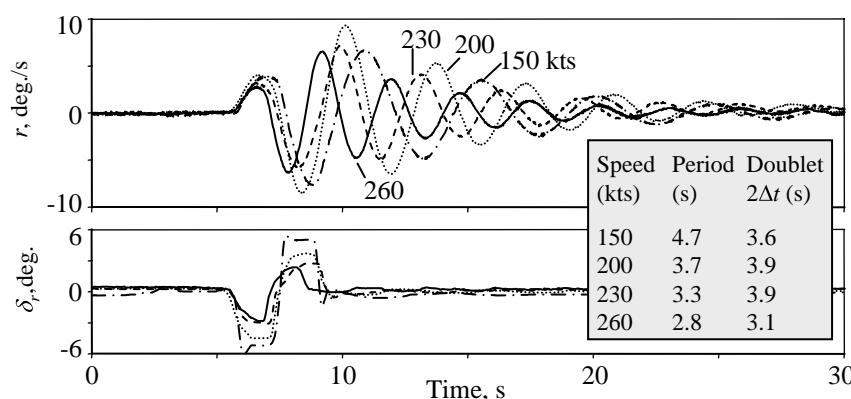


Fig. 2.8 Typical Dutch roll responses.

For the multistep inputs, if applied automatically using an onboard computer, it becomes possible to change the input levels within one sampling time. Such abrupt changes may lead to rate saturation in the electro hydraulic actuators or result in much higher load factors at off-center-of-gravity locations, for example in the cockpit, than at the center-of-gravity. They may also lead to excitation of much higher frequencies and of aeroelastic modes. The optimal harmonic inputs are limited in frequency range and tuned to the desired rigid-body frequencies. This is particularly important when the rigid-body and structural frequencies are close to each other, as tends to be the case for very large aircraft. To avoid such undesirable excitations from multistep inputs, the slopes of the input changes should be limited. On the other hand, if they are applied manually, the inputs do not have extreme sharp edges because the pilot acts as a filter. This does not affect the estimation of the stability and control derivatives, because, as already mentioned, the exact shape and time step are not critical.

Thus, based on our discussion in Secs. II.B and III.B, a variety of maneuvers is usually necessary to excite dynamic motions about different axes. We recommend independent control inputs applied one at a time, and prefer those which can be flown manually by the pilot. The excitation level should be adjusted to the flight condition being tested; at lower dynamic pressures larger amplitudes are necessary, whereas at higher dynamic pressures lower amplitudes can be applied. We have performed a large number of flight tests, first with a small amplitude input followed by a larger amplitude input, but still within the range of the linear model. The large amplitude maneuvers, in which nonlinear aerodynamic effects are encountered, are performed separately. Figure 2.9 summarizes schematically the most important system identification maneuvers.

The various multistep inputs elaborated above belong to the class of binary signals, which consist of a series of optimally spaced pulses, yielding desired

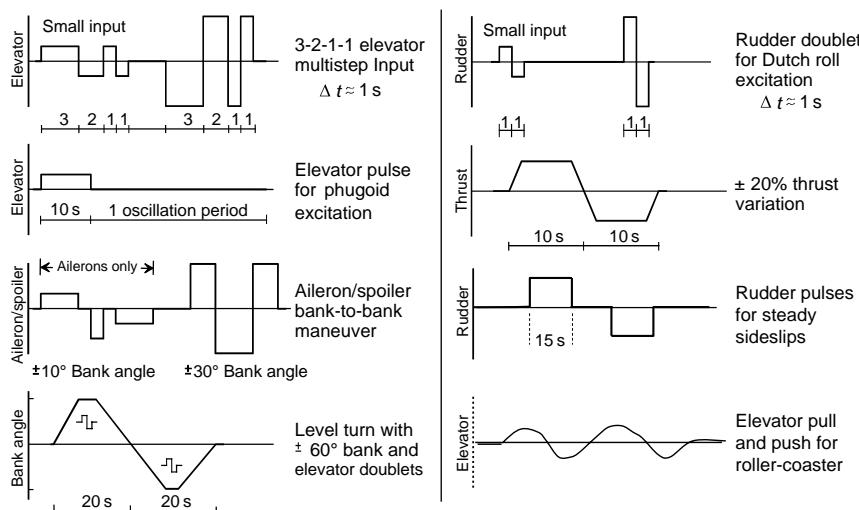


Fig. 2.9 Representative manual control inputs.

frequency spectrum. It is also possible to realize an arbitrary power spectral density spread across a specified bandwidth through superposition of phased harmonic signals.^{33–36} Such inputs belong to the class of multisine signals. One such input that is used in some applications is called the Schroeder-phased multisine signal. By introducing orthogonalization, simultaneous multiaxis optimal excitation signals can be generated. Such input sequences are preferred in modal estimation of structural systems and for estimation of aerodynamic derivatives of hypersonic flight vehicles, because the total test time is usually very short and the flight condition (Mach number) changes rapidly. Furthermore, such input signals have a lot of energy, keeping the input amplitude levels low. In the case of helicopters too, they provide a broadband excitation with improved low-frequency contents.³⁷ As already mentioned in Sec. III.A, an onboard computer implementation is necessary to apply such multisine inputs. Other signals in the class of frequency sweeps, such as Chirp or Fresnel chirp input signal, are also possible.

Although the control inputs such as impulse, doublet, and 3–2–1–1 are sometimes used for ASE applications and for flight flutter investigation, it is more common to adopt the frequency sweep testing in this case.³⁸ In general, the frequency sweeps may not be time-optimal, but they have a broader bandwidth. Frequency sweep testing has been applied, for example, to X-29, EAP aircraft^{39,40} and to identify several structural modes of an XV-15 tilt-rotor aircraft.⁴¹

IV. Scope of Flight Testing

Flight testing is costly, hence it is necessary to limit the number of flight tests to a minimum, and also to optimize the test procedure and maneuver sequence within each flight. As pointed out in Chapter 1, Sec. IV, the goals of system identification could be different, each posing different requirements on the range of validity and fidelity of the model to be identified. The scope of the flight testing for system identification depends mainly on these requirements, that is, on the specific goal for which the model is intended. To some extent, it also depends upon the type of flight vehicle being investigated. The larger the number of configurations to be modeled is, the larger the number of tests to be performed. Thus, the following important aspects need to be considered while organizing a flight test program:

- 1) Define the specific goal of flight testing.
- 2) Select the appropriate configurations to be flown.
- 3) Choose the trim conditions to be tested.
- 4) Define maneuvers to be performed and the input signals to be applied.

Extensive flight testing is required for generating aerodynamic databases for flight simulators meeting the FAA level-D fidelity requirements. Estimation of a so-called global model valid over the complete operational envelope needs a comprehensive flight test program covering all possible configurations. Since the flight vehicle response characteristics depend on flight conditions (angle of attack, sideslip angle, Mach number), dynamic maneuvers will have to be performed in various trim conditions. Furthermore, depending upon the aircraft,

special configurations and effects will have to be investigated, which may call for additional tests. As already mentioned in Sec. II, the data for the simulator certification may have to be gathered separately.

As summarized in Fig. 2.9, a variety of maneuvers is usually necessary to allow collection of data necessary to excite significantly all modes which we will be analyzing. The maneuvers are usually performed in blocks, repeated in different trim conditions defined by Mach number, angle of attack, altitude, and configuration, which will cover flaps, thrust levels (particularly for propeller aircraft), landing gear, center of gravity, and so on. The general procedure is as follows:

- 1) Fly to the desired level and speed.
- 2) Select the configuration and trim the aircraft at a reference flight condition.
- 3) Apply the sequence of inputs; choose the amplitudes such that operational limits are not exceeded.
- 4) Repeat the sequence multiple times at the same trim condition.

Between maneuvers the aircraft is trimmed to horizontal level flight. The same input maneuver is repeated a number of times to enable comparison of estimates, to minimize the influence of external disturbances, and to determine the amount of variation in the measurements. If possible, the same maneuver should be performed three or more times. If the maneuver is performed twice, we face the common dilemma of experimental investigations, namely if the estimates from the two repeat runs deviate significantly, it becomes difficult to choose or discard one or the other. External disturbances are to be minimized, which means that maneuvers should be performed in calm air. Atmospheric disturbances due to thermal activities are smaller early in the morning. The reference point for the estimates obtained from the dynamic maneuvers is usually the starting trim condition. Another possibility is to treat the reference point as the average of the initial trim and flight condition to which the flight vehicle returns after the dynamic maneuver; ideally both should be the same. In a few cases, averaging over the complete maneuver has also been considered.

The reference flight conditions are defined in terms of altitude and calibrated airspeed. Figure 2.10 shows the flight envelope for the Transall C-160 military transport aircraft. As an example, various test flight conditions are shown spread over four altitude levels and several speeds at each level. The equidistant stapling of trim points over altitude and speed, shown in the figure, may not be optimal, and needs to be adapted to each case appropriately. For example, fewer altitude levels would suffice or more points at lower speeds may provide more data at operationally critical flight conditions. Likewise, trim points at several thrust levels should be part of the testing, although this is not explicitly shown in the figure. Different altitude levels allow separation of the Mach number and dynamic pressure effects in the model, because the same trim speed at two different altitudes results in two different Mach numbers. At each point the sequence of maneuvers to be performed is shown at the bottom of the same figure. Usually, it is efficient to perform all the maneuvers at different trim conditions at one flight level, including the repeat runs. Then the flight level

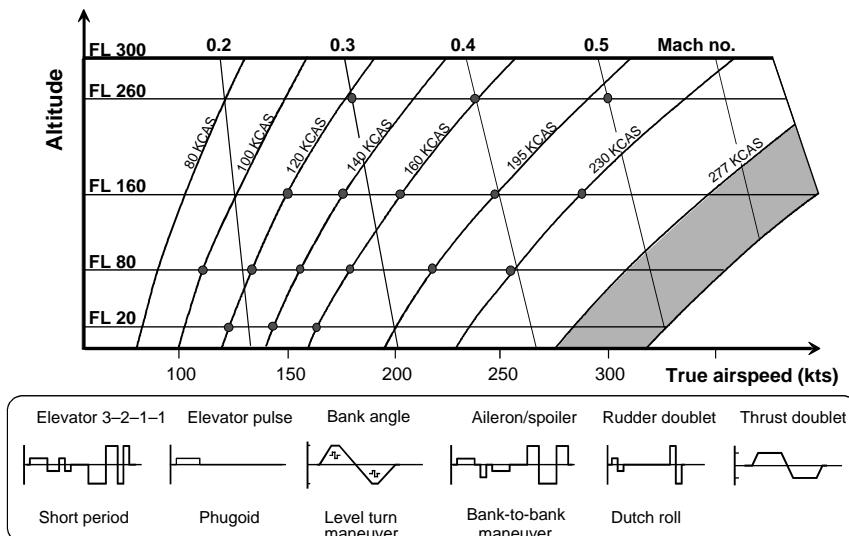


Fig. 2.10 Typical flight test program for system identification.

is changed to follow the same procedure. During the actual flight testing the flight levels may have to be adapted to the prevailing weather conditions.

In some cases, at high Mach numbers it may not be possible to fly level; it may then be necessary to perform the tests in descending flight. Special flight conditions and those pertaining to flight at low altitudes above ground, for example flybys, takeoffs, and landings, are not marked in Fig. 2.10. Similarly, tests at all the altitude levels and speeds are not possible for each configuration. The test points have to be chosen accordingly; for example, dynamic tests with landing gear extended will be possible only up to certain altitudes and speeds. It is also not meaningful to perform such tests with landing gear extended at high altitudes, even if possible, because such test conditions are not realistic.

Development of a global model that is valid over the complete operational envelope requires flight testing for different landing flaps. The common approach is to perform dynamic maneuvers at each landing flap and identify models for each configuration separately. Interpolation over the flaps is incorporated in the final database to cater for the flap change dynamics, which is one of the tests to be covered in validating the models for flight simulators. If the models identified for the individual flap settings are accurate enough, usually the above procedure of interpolation is adequate to meet the fidelity requirements. If not, then some model adjustments are necessary late in the database development, which is a drawback of this approach. On the other hand, the above approach yields more accurate models for each flap configuration, covering significantly larger flight durations than the transition from one flap setting to another. The other approach of identifying global model from simultaneous evaluation of flight maneuvers at different flap settings is not considered efficient, although it may theoretically be possible, given sufficient computing power and

computer memory for data processing. Irrespective of the approach adopted, it is important to note that the input sequence does not include specific dynamic inputs to the landing flaps. Instead, various flaps are chosen as a reference configuration, performing the dynamic tests at each of them.

V. Flight Test Instrumentation and Measurements

Accuracy of parameter estimates is directly dependent on the quality of flight measured data, and hence high-accuracy measurements of control inputs and of motion variables are a prerequisite for successful application of the modern methods of flight vehicle system identification. Classical information on flight test instrumentation for aircraft parameter estimation is provided by Wolowicz⁴² and also found in other references, for example, Ref. 43. During the 1970s the various aspects of flight test instrumentation were investigated in detail.⁴⁴ Dedicated flight test instrumentation systems were developed in several applications of flight vehicle system identification, mainly because 1) several signals from the basic aircraft systems can be tapped, that is, no additional sensors are necessary (for example engine parameters, static pressure, rate gyros, etc.), and 2) control surfaces to be instrumented are flight vehicle-dependent. Although more recently flight instrumentation systems based on commercially available sensors and standard signal processors have been developed to simplify this task, flight instrumentation is still a laborious and time-consuming job. Instead of providing a detailed description of a particular setup, we take a look at the basic requirements in terms of the necessary signals and their importance in the estimation of parameters.

A typical set of measurements required for aerodynamic model extraction from flight data consists of: 1) control surface deflections, 2) linear accelerations, 3) angular rates, 4) attitude angles, 5) air data, 6) static pressure, 7) engine parameters, and 8) pilot forces and inceptor positions. If the scope of modeling is extended to validation of CFD simulations, additional measurements of flow field velocity, other surface variables such as pressures and temperatures, and of loads will become necessary.⁴⁵

Control surface deflections are inputs to the aerodynamic model being identified; inaccuracies in their measurements directly affect estimation results. The deflections are usually measured using potentiometers or LVDT (linear variable differential transducer), mounted directly at the respective hinges. Besides the primary controls (elevator, ailerons, and rudder), other control surfaces like flaps, speed brakes, and spoilers, need to be measured.

Accelerations and angular rates along and about the three axes provide very good information about aerodynamic effects. The inertial navigation system (INS), sometimes called IMU (inertial measuring unit), is an integral unit providing these measurements and those of attitude angles. Some INS incorporate low pass filters, mainly because such systems are used for long-term navigation purposes, where short-term dynamic response is not of particular importance. For aerodynamic modeling purposes, however, dynamic response is of primary importance, hence, it needs to be ascertained that raw data from the INS is available or at least the cut-off frequency of the filter included is high enough. A dedicated tri-axis accelerometer is usually recommended. Accelerations are also

measured at several other locations such as cockpit or the wing tips, horizontal tail, rudder tip, and engines, which are useful for other investigations such as flutter. For rigid-body aerodynamic model identification, a single set of acceleration is sufficient.

The measurements of attitude angles are mostly provided by INS, usually through internal integration of measured angular rates; in rare cases they are obtained using magnetometers. These measurements are primarily useful in the data compatibility check, which we will discuss in Chapter 10, to ascertain quality of the recorded data and to determine biases in the measurements of the angular rates. Usually for aerodynamic model identification, the attitude angles are considered to be unnecessary or of secondary importance, because aerodynamic forces and moments do not depend on the aircraft attitude relative to earth axes. They are not required for the least squares method discussed in Chapter 6, which also includes pre-processing of data to compute aerodynamic coefficients from the measured translational accelerations and angular rates, see Chapter 6, Sec. IX. On the other hand, the attitude angles are useful in the output error method discussed in Chapter 4, as they directly aid the determination of aerodynamic zero terms and of the other derivatives indirectly, because they contain information at lower frequencies; see Fig. 2.11 showing typical plots of time histories and power spectra for attitude angle, angular rate, and angular acceleration related to pitching and rolling motion.

The angular accelerations are generally not measured directly. In many exercises related to system identification, they are derived by numerical differentiation of the measured angular rates. It is our experience that inclusion of angular accelerations, besides the angular rates and attitude angles, significantly improves the accuracy of parameter estimates and overall convergence of the optimization methods. The contributions due to control surface deflections appear directly in the accelerations, and hence they come out much better from the acceleration signals. As seen from Fig. 2.11, these signals contain frequencies higher than those from the corresponding attitude angles and angular rates. Their inclusion leads to a model based on a wider range of frequencies. Based on these considerations, although angular accelerations may not be measured using dedicated sensors, and aerodynamic forces and moments do not depend on the attitude angles, all three variables (angular accelerations, angular rates, and attitude angles) are considered useful in estimating aerodynamic derivatives.

Air data measurements, consisting of angle-of-attack, angle-of-sideslip and airspeed, are also very useful in the parameter estimation. The flow angles are commonly measured using mechanical vanes and the airspeed by stagnation of the flow using a Pitot tube or by propeller. Swiveling probes minimize the alignment errors. These variables are also measured quite often using a five-hole probe, measuring the difference and total pressures. In specific cases, a more advanced flush air data sensing system avoiding an intrusive boom is necessary to measure these parameters. If possible the flow angles should be measured at multiple locations, typically in front of the aircraft or near the pilot station and wing tips. A static pressure measurement is necessary to derive the information on the dynamic pressure, which is the difference between the total and static pressure and also for thrust computations. The static pressure varies along the length of the aircraft, and these measurements usually contain position errors.

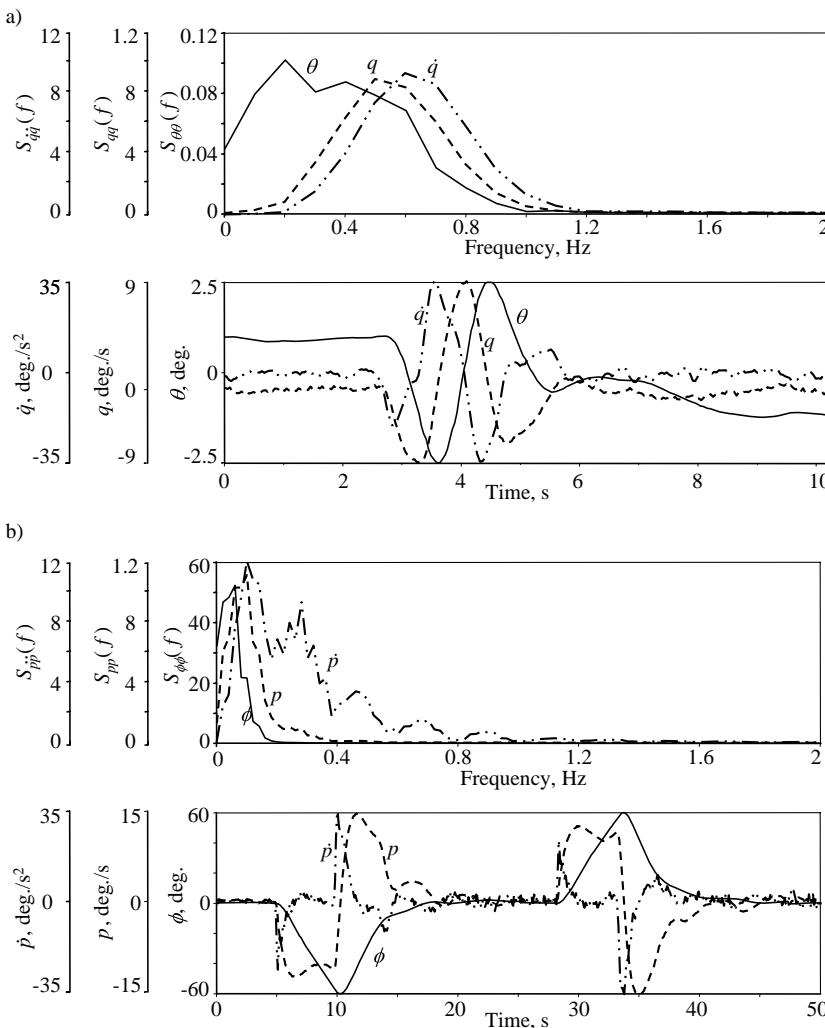


Fig. 2.11 Time histories and power spectra. a) Pitching motion, b) rolling motion.

Availability of additional measurements such as altitude above ground measured using a radio altimeter, the ground speed, and the geodetical positions may be necessary in specific cases.

The engine parameters are necessary to correctly account for the thrust, which is necessary to enable estimation of drag. During aerodynamic model identification, the engine manufacturer-supplied data is used to compute the thrust, which is then treated as known input. In other words, we do not estimate the engine-thrust model. The identified drag model will, of course, depend upon the thrust model. Validating and updating the engine thrust (performance)

model from flight tests is possible, but is not addressed here.^{46,47} The exact set of required engine parameters depends on the aircraft (jet or propeller engines). For example, measurements of power lever and condition lever positions, propeller rpm, low/high pressure rpm, torque, and turbine gas temperature are necessary for propeller engines, whereas those of low pressure engine rpm are necessary for jet engines. The engine-thrust model also needs other variables such as Mach number, airspeed, and static pressure or altitude.

The pilot inputs (stick, wheel, and pedal positions) and pilot forces are necessary only for the modeling of the flight control system, particularly in the case of reversible controls, driven through mechanical rods and linkages. For estimation of rigid-body aerodynamics the measured control surface deflections are treated as inputs to the model.

Besides the foregoing aspect of measurements required for extracting aerodynamic derivatives, other aspects related to data recording also affect data analysis. We simply mention them here without going into any details:

- 1) The minimum sampling rate, that is, the rate at which data is sampled and recorded, is given by the Nyquist frequency, which should be twice the frequency of interest. In general, the data is sampled at a much higher frequency. A sampling frequency of 20–25 Hz is usually sufficient for rigid-body aerodynamic model estimation. For an extended model comprising rigid-body and structural modes, a higher sampling rate is necessary.
- 2) The anti-aliasing filter for all measurements should have the same cut-off frequency, so that all signals will have the same time delays introduced by such filters.
- 3) Recording of raw data is usually preferable.
- 4) Ideally, all data channels should be recorded at the same sampling rate. Some of the data channels tapped from the basic aircraft systems may be available at lower sampling rates, for example slowly varying parameters like altitude. However, they are less critical in the estimation of aerodynamic parameters. More critical measurements like translational accelerations, angular rates, and control surface deflections should be sampled at higher and uniform rates.
- 5) Ideally, all data channels should be time synchronized. This may not be possible due to serial data recording or the onboard system may introduce time delays. It is usually sufficient to know which data channel is recorded fastest; the others can be time-synchronized through data compatibility checking.
- 6) The signal-to-noise ratio of 10:1 is desirable.
- 7) All sensors should be calibrated in laboratory and in situ as far as possible.
- 8) Data reduction should be avoided at the time of recording; if necessary, data reduction can be done while reading the data stored with higher sampling rate by skipping every point or other point during off-line analysis. Such a procedure is better, because once the information is lost due to data compression in the recording, it can never be recovered.

Finally, on-site preliminary verification of recorded data is recommended to ensure the adequacy of the data gathered for off-line analysis.

A. Digital Filter

Since recording of raw data is recommended, in a few cases it may be necessary to suppress high-frequency noise before using the signals. While generating the angular accelerations by numerical differentiation of measured angular rates, it is preferable to filter the measured signals for this specific purpose only. Digital filters based on a linear combination of input values are widely used for this purpose. They have the advantage that, owing to symmetry, they do not introduce any lag in the filtered values. In our applications, we used a low-pass digital filter, developed by Spencer, based on 15 points, the current value, and preceding as well as following seven data points.⁴⁸ It is given by

$$\begin{aligned}y_n = \frac{1}{320} & [-3u_{n-7} - 6u_{n-6} - 5u_{n-5} + 3u_{n-4} \\& + 21u_{n-3} + 46u_{n-2} + 67u_{n-1} + 74u_n \\& + 67u_{n+1} + 46u_{n+2} + 21u_{n+3} \\& + 3u_{n+4} - 5u_{n+5} - 6u_{n+6} - 3u_{n+7}]\end{aligned}\quad (2.13)$$

The above equation can be correctly implemented for a truncated time series for all data points except for the first and the last seven. To smooth the two ends of the time series, a filter based on fewer points is applied. Owing to the lack of sufficient information the first two and the last two data points cannot be filtered accurately. This needs to be kept in mind while using above procedure. For the data points 3–7 from the two ends the following filter using two data points can be incorporated:

$$y_n = \frac{1}{96}[7u_{n-2} + 24u_{n-1} + 34u_n + 24u_{n+1} + 7u_{n+2}] \quad (2.14)$$

Equations (2.13) and (2.14) have been coded for a single time segment in the function “/FVSSysID/chapter02/smooth.m” and for multiple time segments in “smoothMultTS.m,” which is used in the following chapters.

B. Numerical Differentiation

As already pointed out earlier in this section, angular accelerations are obtained by numerical differentiation of the measured angular rates. Time derivatives of other measurements may also be needed in specific cases. The simplest way to obtain such time derivatives is to use the one-sided forward or two-sided central difference formulas. We will deal in more detail with such formulas in Chapter 3, Sec. V.C in connection with approximating the system functions. Such simple formulas are not quite suitable for numerical differentiation of measured time histories, because they tend to amplify the noise. In spite of high-quality sensors, the presence of measurement noise is unavoidable. Better results are obtained using a numerical differentiator incorporating a suitable low-pass digital filter. Without going into mathematical details, the frequency response of a differentiating smoother given by $i\Omega/\Delta t$, where $\Omega = \omega\Delta t$, having no real part, can be

constructed by the following digital filter of arbitrary order given by⁴⁹

$$y(k) = \frac{1}{\Delta t} \sum_{i=1}^N C_i [x(k+i) - x(k-i)] \quad (2.15)$$

where k is the discrete time point, x the measured signal, y the time derivative of the measured signal, N the order of the filter incorporated within a first-order differentiator, and C_i the corresponding coefficients. The coefficients C_i are obtained from the equation

$$AC = b \quad (2.16)$$

where the vector b and the elements of the n -by- n matrix A are given by:

$$b^T = [1 \ 0 \ 0 \ 0 \ \dots]; \quad a_{ij} = (-1)^{i+1} j^{2i-1} \quad (2.17)$$

Equations (2.16) and (2.17) yield the following coefficients for differentiating smoother from first to fourth order respectively: (0.5), (0.66666667, -0.083333333), (0.75, -0.15, 0.016666667) and (0.8, -0.2, 0.038095238, -0.003571428). These coefficients and those for the higher-order smoother differentiator can be generated using the program “/FVSysID/chapter02/numDiffCoeff.m,” and verified on a simple test case using “ndiffVerify.m.” It is fairly obvious that Eq. (2.15) for $N = 1$ corresponds to the central difference formula. We have used eighth and 12th-order smoother differentiators in several applications. We will be using the first-order differentiator with eighth-order filter (function “ndiff_Filter08.m”) in some of the test cases analyzed in this book. Other versions of centrally pivoted algorithms for smoothing and differentiation are possible.⁵⁰

In spite of great care in choosing sensors, their installation and calibration, and in recording the data, instrument errors such as bias and time delays are unavoidable. In some cases verification of scale factors is necessary, particularly for flow angles, because they may be configuration dependent and realistic conditions may not be duplicated in a laboratory. Such a step dealing with data checking, estimating instrument errors and generating refined measurements is part of the overall data gathering process. We defer the discussion of this step, called data compatibility checking, to Chapter 10, because it requires some of the estimation algorithms which we will discuss in the following chapters.

VI. Concluding Remarks

In this chapter we have discussed an important aspect of data gathering for system identification purposes. In terms of the Quad-M basics introduced in the previous chapter, we covered in detail the aspects pertaining to the first M for maneuvers, and a part of the second M for measurements. In essence, optimal input design boils down to two simple basic principles: 1) inputs must be such as to excite significantly a range of frequencies around the eigenmodes which are to be modeled; and 2) independent input on every control is necessary.

Two approaches to designing optimal inputs have been discussed; the one based on the estimation error criterion is more rigorous, and the other is the more practical one based on a simple engineering process of synthesizing the contributions due to each of the parameters appearing in each of the force and moment equations. Since both designs are based on the a priori model involving parameter uncertainties, it is necessary to provide for a broader band input signal and one that can be easily adapted to changing flight conditions.

It is our experience that, from basic a priori knowledge about the modes of motion, which we mostly possess, adequate inputs can be designed for conventional aircraft without detailed investigations by following a few simple guidelines: 1) the multistep 3–2–1–1 input is preferable for short period mode, whereas a simple doublet is quite sufficient for Dutch roll; rolling motion is adequately excited by a bank-to-bank maneuver; 2) the exact shape of the doublet or of the multistep inputs and the optimal time step for these inputs are not critical; and 3) the duration of time steps is obtained from simple empirical rules suggested here. To further improve upon the estimation results, it is preferable to apply one control input at a time during dynamic maneuvers and to minimize excitation of the unmodeled modes. We believe that the gains obtained through a more detailed design of input signals are marginal; furthermore the improvements in the statistical accuracies as predicted by the error covariance matrix are associated with certain limitations with which we will deal in one of the following chapters. Thus, we have followed a simple philosophy in many applications, namely stick to the basics elaborated above, and use a combination of the different system identification maneuvers briefly discussed in this chapter.

In special cases such as hypersonic flight vehicles, however, more advanced experiment design will be required, because the duration of the total test time is usually short and the flight condition (Mach number) changes rapidly. In such cases it is desirable to have input signals with a lot of energy, keeping the input amplitude levels low. Multiaxis orthogonal phase-optimized sweeps with a wide frequency band are more suitable in these applications. Similarly, inputs other than the multistep signals may be more suitable for aeroservoelastic applications.

The scope of the flight testing depends on the desired range of validity and fidelity of the model; hence defining the goal of system identification prior to data gathering is necessary to optimize the scope of flight testing. The larger the number of configurations to be modeled and the range of the desired validity, the larger the number of tests to be performed. A brief summary of various flight maneuvers has been provided to gain insight into experimental techniques. It is necessary to point out here that the primary goal of flight testing is to certify flight worthiness and safety, whereas conforming to the subject of this book we have focused here only a very specific part, namely that which allows more accurate validation and updating of aerodynamic database.

A brief summary of typical measurements required for parameter estimation has also been provided. It has been argued that, besides the control surface deflections, translational accelerations, angular rates, and flow angles that are necessary, the angular accelerations and attitudes angles are useful in extracting aerodynamic derivatives, yielding a model covering a broader frequency range.

References

- ¹Milliken, W. F., Jr., "Dynamic Stability and Control Research," *Proceedings of the 3rd Anglo-American Aeronautical Conference*, Brighton, 1951, pp. 447–524.
- ²Hamel, P. G. and Jategaonkar, R. V., "Evolution of Flight Vehicle System Identification," *Journal of Aircraft*, Vol. 33, No. 1, 1996, pp. 9–28.
- ³Levin, M. J., "Optimal Estimation of Impulse Response in the Presence of Noise," *IRE Transactions on Circuit Theory*, Vol. CT-7, No. 1, 1960, pp. 50–56.
- ⁴Litmann, S. and Huggins, W. H., "Growing Exponentials as a Probing Signal for System Identification," *Proceedings of the IEEE*, Vol. 51, June 1963, pp. 917–923.
- ⁵Levadi, V. S., "Design of Input Signals for Parameter Estimation," *IEEE Transactions on Automatic Control*, Vol. AC-11, No. 2, 1966, pp. 205–211.
- ⁶Gerlach, O. H., "Analysis of a Possible Method for the Measurement of Performance and Stability and Control Characteristics of an Aircraft in Non-Steady Symmetric Flights," Report VTH-117, Dept. of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands, Nov. 1964 (in Dutch with summary in English).
- ⁷Gerlach, O. H., "Determination of Performance, Stability and Control Characteristics from Measurements in Non-Steady Manoeuvres," AGARD CP-17, Sept. 1966, pp. 499–523.
- ⁸Mehra, R. K., "Optimal Inputs for Linear System Identification," *IEEE Transactions on Automatic Control*, Vol. AC-19, No. 3, 1974, pp. 192–200.
- ⁹Stepner, D. E. and Mehra, R. K., "Maximum Likelihood Identification and Optimal Input Design for Identifying Aircraft Stability and Control Derivatives," NASA CR-2200, March 1973.
- ¹⁰Marchand, M., "Untersuchung der Bestimmbarkeit von Flugzeugderivativen aus Flugversuchen," DFVLR IB 154-74/32, Dec. 1974 (in German).
- ¹¹Gupta, N. K. and Hall, W. E., Jr., "Input Design for Identification of Aircraft Stability and Control Derivatives," NASA CR-2493, Feb. 1975.
- ¹²Koehler, R. and Wilhelm, K., "Auslegung von Eingangssignalen für die Kennwertermittlung," DFVLR-IB 154-77/40, Dec. 1977 (in German).
- ¹³Plaetschke, E. and Schulz, G., "Practical Input Signal Design," AGARD LS-104, Paper 3, Nov. 1979.
- ¹⁴Mulder, J. A., Sridhar, J. K., and Breeman, J. H., "Identification of Dynamic Systems—Applications to Aircraft, Part 2: Nonlinear Analysis and Manoeuvre Design," AGARD AG-300, Vol. 3, Part 2, May 1994.
- ¹⁵Proskawetz, K.-O., "Optimierung stufenförmiger Eingangssignale im Frequenzbereich für die Parameter Identifizierung," ZFW, Vol. 9, No. 6, 1985, pp. 362–370 (in German).
- ¹⁶Gates, R. J., Bowers, A. H., and Howard, R. M., "A Comparison of Flight Input Techniques for Parameter Estimation of Highly-Augmented Aircraft," AIAA Paper 96-3363, Aug. 1996.
- ¹⁷Morelli, E. A., "Flight Test of Optimal Inputs and Comparison with Conventional Inputs," *Journal of Aircraft*, Vol. 36, No. 2, 1999, pp. 389–397.
- ¹⁸"Airplane Simulator Qualification," FAA Advisory Circular, AC 120-40C, Draft Version, July 1995.
- ¹⁹"Joint Aviation Requirements—Aeroplane Flight Simulators," JAR-STD 1A, Westward Digital Ltd., Cheltenham, April 1997.
- ²⁰Kimberlin, R. D., *Flight Testing of Fixed-Wing Aircraft*, AIAA, Reston, VA, Sept. 2003.
- ²¹Lee, R. E., Jr., "Handling Qualities," AGARD-AR-300 Vol. 14, Paper No. 15, Sept. 1995.

²²Gallagher, G. L., Higgins, L. B., Khinoo, L. A., and Pierce, P. W., "Fixed Wing Performance," Naval Test Pilot School Flight Test Manual, Report USNTPS-FTM-No. 108, Sept. 1992.

²³Anon, "Safety Recommendation," National Transportation Safety Board, Washington, D.C., No. A-02-01 and -02, 8 Feb. 2002.

²⁴Tischler, M. B., Fletcher, J. W., Diekmann, V. L., Williams, R. A., and Cason, R. W., "Demonstration of Frequency-Sweep Testing Techniques Using a Bell 214-ST Helicopter," NASA TM-89422, April 1987.

²⁵Kolwey, H., "Rotorcraft Frequency Domain Flight Testing," Tech Memo for the American Helicopter Society T&E Committee, RW-51, March 1994.

²⁶Ham, J. A., "Frequency Domain Flight Testing and Analysis of an OH-58D Helicopter," *Journal of the AHS*, Vol. 37, No. 4, Oct. 1992, pp. 16–24.

²⁷Maine, R. E. and Iliff, K. W., "Identification of Dynamic Systems," AGARD AG-300, Vol. 2, Jan. 1985.

²⁸Goodwin, G. C., and Payne, R. L., *Dynamic System Identification: Experiment Design and Data Analysis*, Academic Press, New York, 1977.

²⁹Marchand, M., "Untersuchung der Bestimmbarkeit der flugmechanischen Derivative des CCV-Versuchsträgers F-104 G," DFVLR IB 154-77/12, Mar. 1977 (in German).

³⁰Blackman, R. B. and Tukey, J. W., *The Measurement of Power Spectra*, Dover, New York, 1958.

³¹Paris, A. C. and Bonner, M., "Nonlinear Model Development from Flight-Test Data for F/A-18E Super Hornet," *Journal of Aircraft*, Vol. 41, No. 4, 2004, pp. 692–702.

³²Plaetschke, E., Mulder, J. A., and Breeman, J. H., "Results of Beaver Aircraft Parameter Estimation," DFVLR-FB 83-10, Feb. 1983.

³³Schroeder, M. R., "Synthesis of Low-Peak-Factor signals and Binary Sequences with Low Autocorrelation," *IEEE Transactions on Information Theory*, Vol. 16, No. 1, 1970, pp. 85–89.

³⁴Schoukens, J., Pintelon, R., Van der Ouderaa, E., and Renneboog, J., "Survey of Excitation Signals for FFT Based Signal Analyzers," *IEEE Transactions on Instrumentation and Measurement*, Vol. 37, No. 2, 1988, pp. 342–352.

³⁵Roberto, J. and Arruda, F., "Multisine Multiexcitation in Frequency Response Function Estimation," *AIAA Journal*, Vol. 31, No. 1, 1993, pp. 215–216.

³⁶O'Neill, C. R. and Arena, A. S., Jr., "Time-Domain Training Signals Comparison for Computational Fluid Dynamics Based Aerodynamic Identification," *Journal of Aircraft*, Vol. 42, No. 2, 2005, pp. 421–428.

³⁷Young, P. and Patton, R. J., "Comparison of Test Signals for Aircraft Frequency Domain Identification," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, 1990, pp. 430–438.

³⁸Brenner, M. J., Lind, R. C., and Voracek, D. F., "Overview of Recent Flight Flutter Testing Research at NASA Dryden," NASA TM-4792, April 1997.

³⁹Clarke, R., Burken, J. J., Bosworth, J. T., and Bauer, J. E., "X-29 Flight Control System: Lessons Learned," *International Journal of Control*, Vol. 59, No. 1, Jan. 1994, pp. 199–219.

⁴⁰Caldwell, B. D., "The FCS-Structural Coupling Problem and its Solution," AGARD CP-560, Paper 16, Jan. 1995.

⁴¹Acree, C. W., Jr., and Tischler, M. B., "Identification of XV-15 Aeroelastic Modes Using Frequency Sweeps," *Journal of Aircraft*, Vol. 26, No. 7, 1989, pp. 667–674.

⁴²Wolowicz, C. H., "Considerations in the Determination of Stability and Control Derivatives and Dynamic Characteristics from Flight Data," AGARD AR-549, Part 1, 1966.

⁴³Borek, R. W. and Pool, A. (eds.), "Basic Principles of Flight Test Instrumentation Engineering," AGARD AG-160, Vol. 1 (Issue 2), Mar. 1994.

⁴⁴Breeman, J. H., Woerkom, van, K., Jonkers, H. L., and Mulder, J. A., "Aspects of Flight Test Instrumentation," AGARD LS-104, Paper 4, Nov. 1979.

⁴⁵Marvin, J. G., "Perspective on Computational Fluid Dynamics Validation," *AIAA Journal*, Vol. 33, No. 10, 1995, pp. 1778–1787.

⁴⁶Rooney, E. C. and Wilt, C. E., "Development of In-flight Thrust Measurement Procedures for an Afterburning Turbofan Engine," AIAA-1985-1405, July 1985.

⁴⁷Niewald, P. W., and Parker, S. L., "Flight-Test Techniques Employed to Successfully Verify F/A-18E In-flight Lift and Drag," *Journal of Aircraft*, Vol. 37, No. 2, 2000, pp. 194–200.

⁴⁸Kendall, M., Stuart, A., and Ord, J. K., *The Advanced Theory of Statistics, Volume 3: Design and Analysis, and Time-Series*, Charles Griffin, London, 1983.

⁴⁹Koehler, R., "Experimentelle Aspekte," Lecture Notes: 'Praktische Einführung in die Systemidentifizierung', Carl-Cranz-Gesellschaft, Oberpfaffenhofen, Germany, 11–13 June 1985, Sect. 9 (in German).

⁵⁰Forsythe, W., "Digital Algorithms for Prediction, Differentiation and Integration," *Transactions of the Institute of Measurement and Control*, Vol. 1, No. 1, Jan.–March 1979, pp. 46–56.

Chapter 3

Model Postulates and Simulation

I. Introduction

DYNAMICAL SYSTEMS can be described by differential equations, whose order depends upon the process complexity, coupling between the subsystems, and on the degree of accuracy required for the specific application. The state space representation transforms higher-order differential equations into a set of coupled first-order equations. The two important issues that characterize the postulated model are 1) choice of the state variables and 2) input–output and internal system behavior. The set of states which can be chosen to represent a system is never unique, but will depend upon the pertinent physical characteristics of the system being modeled. Having selected appropriate states, the internal system behavior is then characterized by the system parameters. Such models in terms of states and parameters for real-world processes are mostly nonlinear. Linear system models are simplified representations of nonlinear processes. They are obtained through linearization about a pre-defined operating point, and hence are valid for small variations around the point of linearization.

Having postulated a model, it becomes possible to investigate the time propagation of states through simulation, usually by solving an initial-value problem applying numerical integration procedures. The model responses are composed of different modes, some of which may be fast decaying and others slow, requiring longer oscillation time. For stiff systems, that is, when two modes characterized by the smallest and the largest eigenvalues differ greatly, special integration algorithms are necessary. The fidelity of simulated model responses is determined by comparing them with the measured system outputs. Because measurements are likely to be corrupted by deterministic errors such as scale factor or bias as well as by stochastic noise, additional sensor models may be necessary. Such recorded measurements are usually available at discrete time points, although the actual process evolves in continuous time.

For aircraft parameter estimation, state space models describing the aircraft motion are generally necessary. In the field of flight mechanics, such models are invariably derived from the Newtonian mechanics; they lead to well-formulated kinematic equations pertaining to the equation of aircraft motion with translational and rotational degrees of freedom.^{1,2} The instantaneous aerodynamic forces and moments acting on a vehicle characterize the internal and external behavior. Since the kinematic equations are well defined, the efficacy of

aircraft parameter estimation depends primarily on aerodynamic model postulates. Modeling of aerodynamic forces and moments is much more complex. The kinematic equations and aerodynamic models valid over the entire flight envelope are invariably nonlinear. We often make the assumption of rigid body aircraft to simplify the model formulation. This has mostly been adequate in the past for the purpose of aircraft parameter estimation. For flight mechanic simulations, even for large transport aircraft, the influence of structural deformations can be described in a quasi-steady manner through a dynamic pressure dependent flex factor, provided the rigid-body and structural frequencies are sufficiently separated. Thereby the complex generalized equations of motion are usually avoided.

In this chapter we mainly concentrate on the general formulation of the state space models that are amenable to time-domain methods, which we will discuss in great detail in the following chapters. It is not the goal of this chapter to provide a treatment on aircraft equations of motion; these equations are found in any standard text book on flight mechanics. They are also provided for the examples covered in this book. Specifically, we look here at the various forms including linear and nonlinear models and at the extensions necessary to deal with practical issues such as mixed continuous/discrete system representation, initial conditions, deterministic bias errors in the measurements, and simultaneous evaluation data recorded from multiple experiments. Special emphasis is placed on the treatment of time delays. Those may result from either the instrumentation system or the internal behavior of the system. Finally, commonly applied numerical integration techniques are briefly discussed. Based on typical results, some general recommendations are made for the adequate choice of integration method to estimate aerodynamic derivatives from measured flight data.

II. Model Description

In a general case, the mathematical model of a process in state space is given by^{3–5}

$$\dot{x}(t) = f[x(t), u(t), \beta] + F(\lambda)w(t), \quad x(t_0) = x_0 \quad (3.1)$$

$$y(t) = g[x(t), u(t), \beta] \quad (3.2)$$

where x is the $n_x \times 1$ column vector of state variables, u the $n_u \times 1$ control input vector, y the $n_y \times 1$ system output vector, and β the $n_q \times 1$ vector of system parameters. The n_x and n_y dimensional system functions f and g are general nonlinear real-valued functions. These system functions are assumed to have sufficient differentiability to be able to invoke Taylor series expansion. Besides the deterministic control input u , the system is also excited by stochastic input, called process noise $w(t)$, an $n_w \times 1$ column vector, which is usually nonmeasurable. In our investigations we assume $n_w = n_x$. The process noise is usually assumed to be a zero-mean white Gaussian noise with an identity power spectral density. The matrix F represents the additive process (state) noise distribution matrix.

Because it is not possible to measure the system parameters β , they have to be estimated from the discrete measurements $z(t_k)$ of the model outputs $y(t_k)$. Since measurements are invariably corrupted by noise, the output equation can

be formulated as:

$$z(t_k) = y(t_k) + Gv(t_k) \quad (3.3)$$

where k is the discrete time index and $v(t_k)$ the $n_v \times 1$ measurement noise vector. Throughout our discussion we consider $n_v = n_y$. The measurement noise is assumed to be characterized by a sequence of independent Gaussian random variables with zero mean and identity covariance. The matrix G represents the additive measurement noise distribution matrix.

It is assumed that, besides the system parameters β , the elements λ of process noise distribution matrix F , and the initial conditions x_0 are also unknown. Accordingly, the unknown parameter vector is given by:

$$\Theta = [\beta^T \quad \lambda^T \quad x_0^T]^T \quad (3.4)$$

The measurement noise distribution matrix G is also unknown. However, estimation of G , or alternatively the covariance matrix $R (=GG^T)$, is treated separately. This will be elaborated in more details in Chapters 4 and 5. For the present, it suffices to deal with the unknown parameter vector of Eq. (3.4). In our particular case of flight vehicle system identification the system parameters β correspond to the stability and control derivatives or other parameters modeling aerodynamic forces and moments.

III. Extensions of the Mathematical Models

In the preceding section it was assumed that measurements of control variables u are error free and that the output measurements y are corrupted by Gaussian noise only. In practice, however, the measurements of both these variables contain systematic errors. These measurement errors are unavoidable and we have to account for them in the parameter estimation through the postulated model. Let us denote these so-called zero shifts (also termed measurement biases or offsets) in the control variables u and output variables z as Δu and Δz , respectively. The zero shifts are treated as constant over the period of observation and are usually unknown. They represent, of course, very simple sensor models; in reality the sensor models may be much more complex.

Considering the zero shifts in the control and output variables, the general system representation of Eqs. (3.1)–(3.3) can be rewritten as

$$\dot{x}(t) = f[x(t), u(t) - \Delta u, \beta] + F(\lambda)w(t), \quad x(t_0) = x_0 \quad (3.5)$$

$$y(t) = g[x(t), u(t) - \Delta u, \beta] \quad (3.6)$$

$$z(t_k) = y(t_k) + \Delta z + Gv(t_k) \quad (3.7)$$

The unknown parameter vector in this case is given by:

$$\Theta = [\beta^T \quad \lambda^T \quad x_0^T \quad \Delta u^T \quad \Delta z^T]^T \quad (3.8)$$

Besides the n_q number of unknown system parameters β and λ s, we have $(n_x + n_u + n_y)$ initial conditions and zero shifts. In practice, it may not always be possible to estimate all the initial conditions x_0 and the zero shifts Δu and Δz independently because of high correlation. For nonlinear systems, it is essential to estimate the initial condition x_0 . The linear dependence of zero shifts can be determined from the estimation error correlation matrix, which we will study in Chapter 4, Sec. XVII. Theoretical investigation of parameters identifiability is difficult for nonlinear systems. However, a more detailed treatment of this important issue is presented in Sec. V.A considering linear system models.

Yet another extension of the model represented in Eqs. (3.5)–(3.7) results from the practical limitation of performing experiments. A large number of experiments may have to be carried out separately to enable estimation of a single set of system parameters. A typical example pertains to estimation of derivatives pertaining to the lateral-directional motion. Even when we restrict our attention to a simple model that is valid for a single trim point, the lateral-directional motion consisting of rolling, yawing, and side-slipping requires aileron and rudder control inputs. Although it would be possible to apply aileron as well as rudder inputs simultaneously, it is preferable to carry out these maneuvers separately. In general, to aid identifiability of parameters, in many cases only one control input is varied, keeping the other controls fixed, and if it is not possible to keep the other controls fixed, at least the variations in them are minimized. Furthermore, many of the stability and control derivatives are functions of angle-of-attack or angle-of-sideslip. To estimate such dependencies and other nonlinear effects, it is necessary to perform flight experiments at different trim points and excite the dynamic motion about each axis separately.

Thus, a capability to process simultaneously multiple experiments (flight maneuvers) containing different information to estimate a single set of parameters is a necessity that arises out of practice. The initial conditions on the state variables, that is, trim conditions, vary from maneuver to maneuver. Theoretically, the zero shifts Δu and Δz should be independent of the experiments, because they represent systematic errors in the sensors; as long as the same sensors and recording system are used, these errors should be constant. However, in reality, owing to temperature effects, vibrations, and differences in laboratories and flight calibrations, they vary from experiment to experiment, albeit the variations may be small. Therefore, in a general case, we have to account for the initial conditions and systematic errors separately for each experiment. In this case the system representation of Eqs. (3.5)–(3.8) can be extended as follows:

$$\dot{x}(t) = f[x(t), u(t) - \Delta u(\gamma_l), \beta] \quad x(t_0) = x_0(\alpha_l) \quad (3.9)$$

$$l = 1, 2, \dots, n_E$$

$$y(t) = g[x(t), u(t) - \Delta u(\gamma_l), \beta] \quad (3.10)$$

$$z(t_k) = y(t_k) + \Delta z(\delta_l) + Gv(t_k) \quad (3.11)$$

where α_l , γ_l , and δ_l represent respectively the unknown components of x_0 , Δu and Δz for the l th experiment, and n_E is the number of experiments (time segments) to be analyzed simultaneously.

The complete unknown parameter vector in this case is given by

$$\Theta = [\beta^T \quad \alpha_1^T \quad \alpha_2^T \quad \dots \quad \alpha_{nE}^T \quad \gamma_1^T \quad \gamma_2^T \quad \dots \quad \gamma_{nE}^T \quad \delta_1^T \quad \delta_2^T \quad \dots \quad \delta_{nE}^T]^T \quad (3.12)$$

We notice here that the process noise matrix $F(\lambda)$ has been dropped from the system representation in Eq. (3.9). Theoretically, it would be possible to include λ s separately for each of the n_E experiments. However, treatment of separate distribution matrices for multiple experiments leads to some algorithmic difficulties which we will address in more detail in Chapter 5, dealing with estimation accounting for process noise. If we assume that the process noise distribution matrix F remains the same, then the elements λ can be easily included in the vector of unknown parameters of Eq. (3.12), just like the system parameters β . This is considered a viable approach, because in many of the cases, we analyze multiple experiments carried out under similar atmospheric conditions. Moreover, the estimates of system parameters are relatively insensitive to exact values of the process noise distribution matrix.

IV. Retarded Systems

Modeling of physical phenomena in many cases requires time delays to be accounted for. These may result either from measurement and recording systems or in our specific case of aerodynamic modeling as transit time effects. First, let us consider a case of calibrating a five-hole probe for flow angles mounted on a nose boom, in particular the differential pressure for the angle of attack, although any arbitrary variable would have served the purpose here of explaining model formulation. For the recorded variable corresponding to the differential pressure, $p_{d\alpha}$, let us consider a typical measurement equation of the form

$$p_{d\alpha m}(t) = K_\alpha p_{dyn}(t)\alpha_{nb}(t) + \Delta p_{d\alpha} \quad (3.13)$$

where the subscript m refers to the measurements, that is, $p_{d\alpha m}$ is the measured differential pressure, α_{nb} the computed angle of attack at the nose boom (model output), p_{dyn} the computed dynamic pressure, K_α the unknown scale factor, and $\Delta p_{d\alpha}$ the unknown bias in the measurement. Note that the variables α_{nb} and p_{dyn} appearing on the right-hand side of Eq. (3.13) are obtained from the model (state) variables estimated through flight path reconstruction technique, which will be discussed in Chapter 10.

To account for the time delays, we reformulate Eq. (3.13) as follows:

$$p_{d\alpha m}(t) = p_{d\alpha C}(t - \tau_\alpha) + \Delta p_{d\alpha} \quad (3.14)$$

where τ_α denotes the time delay in the measured variable, and $p_{d\alpha C}$ is the computed quantity, which is basically the same as that given by the first term on the right-hand side of Eq. (3.13), namely $p_{d\alpha C}(t) = K_\alpha p_{dyn}(t)\alpha_{nb}(t)$. Equation (3.14) implies that current measurement $p_{d\alpha m}$ (on the left-hand side) at time t is a function of the past variable $p_{d\alpha C}$ at time $t - \tau_\alpha$. In other words, the computed variable is to be time delayed to match the measured data, which is the same as saying

that the measurements of $p_{d\alpha}$ contain time delay. As postulated in Eq. (3.14), the time delays τ always come out positive, which is consistent with the physical interpretation of the phenomenon. For causal systems, negative time delays are not physically meaningful, because that would imply anticipating the future system characteristics ahead of time.

The second example that we consider pertains to modeling of transit time effect resulting from internal system behavior. As a specific case pertaining to aircraft motion, multi-point aerodynamic models considering, for example, the wing–body combination and horizontal tail separately include such transit time effects to model the time required for flow variations generated at the wing to reach the tail.^{1,6} Such a transit time is a function of the forward speed, and hence variable. This aerodynamic phenomenon is commonly termed the downwash lag effect. From flight mechanics we know that the angle of attack at the horizontal tail α_H can be modeled as^{1,6,7}

$$\alpha_H = \alpha + i_H - \varepsilon_H + \alpha_{dyn} \quad (3.15)$$

where α is the angle of attack at the wing, i_H the horizontal-tail trim angle, ε_H the downwash angle at the tail, and α_{dyn} the dynamic angle of attack. The downwash and the lag effect can be modeled as:

$$\varepsilon_H = \frac{\partial \varepsilon_H}{\partial \alpha} \alpha(t - \tau) + \frac{\partial \varepsilon_H}{\partial C_T} C_T(t - \tau) \quad (3.16)$$

where $\partial \varepsilon_H / \partial \alpha$ and $\partial \varepsilon_H / \partial C_T$ denote the unknown downwash parameters, C_T the thrust coefficient and $\tau = r_H/V$ the transit time, where r_H is the tail length (i.e. the horizontal distance between the neutral points of wing and horizontal tail). Equation (3.16) models the downwash generated due to angle-of-attack variations and also due to the thrust changes for wing-mounted engines. Other wing mounted control devices such as direct-lift-control flaps or speed brakes also contribute to downwash and may have to be included in the model of Eq. (3.16). An example of accounting for transit time delay in the estimation from flight data using such a model is presented in Chapter 12.

Having brought out the need to account for time delays, we now turn our attention to possible ways to account for them in the estimation procedure. In general, there are three different ways: 1) data preprocessing, 2) first-order lag or Padé approximation, and 3) delay array. The first option, called data preprocessing, is a fairly simple and commonly applied procedure, which is a part of the flight data reading process. Being a data preprocessing step, no changes to the estimation algorithm are necessary; the only program changes that may be necessary are for the part that reads the flight data, see Fig. 3.1. It enables time shifting variables through a prespecified fixed value in seconds, usually multiples of the sampling time, although through linear interpolation even time shifts of partial sampling time can be done. Time shifting the data in both directions, that is, for positive and negatives values, is theoretically feasible. If the recorded length is long enough, selection of a portion of time segment for further analysis after time shifting does not pose any problem. If recorded data are short in length,

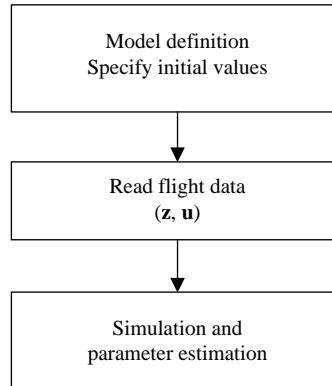


Fig. 3.1 General flow chart of data processing.

for example just the maneuver part, then at respective ends time shifting may not be possible; in such case the time segment for data analysis will have to be reduced by a few data points. However, this is not a major limitation, because usually longer records are available. The approach is necessarily limited to directly measured variables, and can be used to time shift the measurements of the specified output, control, and state variables prior to estimation. The time delays in various channels may be determined from the flight path reconstruction techniques discussed in Chapter 10 or through elaborative data recording systems with an option to time stamp each channel. In any case, a time reference is needed to perform the time synchronization, and also for estimation using other methods. This is provided by the channel that is recorded fastest with a minimum of time delay. All other channels are shifted relative to the time frame of this best available signal, which in many cases happens to be linear accelerations measured by dedicated accelerometers and stored directly.

The second option to account for the time delays is through an approximation by a first-order lag.⁸ To illustrate the approach, let us consider the following equation for forward speed u :

$$\dot{u} = X_0 + X_u u + X_w w - qw + rv - g \sin \theta + \frac{F_e}{m} \cos \sigma_T \quad (3.17)$$

To introduce now time delay in the state variable u , we introduce an additional state variable u_ℓ and model the same as follows:

$$\dot{u}_\ell = (u - u_\ell)/\tau_u \quad (3.18)$$

It is apparent that Eq. (3.18) models a first-order lag effect. Thus, the new state variable u_ℓ is time delayed through τ_u . Having extended the state model, we now compare the time delayed u_ℓ obtained through integration of the state equation, and not the u , with the measured variable of forward speed.

The above approach is simple and does not require any sophisticated estimation program capabilities. Although this approach is adopted by some analysts, it is not the most efficient one, because 1) an additional first-order differential equation of the form of Eq. (3.18) is required for each variable to be time-shifted, leading to larger computational overhead (computational time is directly proportional to the number of state variables which have to be integrated), and 2) some dynamic effects, albeit small, are introduced due to the approximation. For a complex aerodynamic model, the increase in computational overhead due to additional state equations for time delays may not be too large, because, as will be demonstrated in Chapter 4, Sec. XVIII, the major burden is usually to compute the right-hand sides of the state equations containing forces and moments.

The third option uses an array to generate time delay during the estimation procedure in any specified variable. It is much more complex, but provides a more accurate representation of the time-shift phenomenon. Separate delay arrays are required for each variable to be time-shifted. Since it is part of the computational procedure, time delays can be treated as unknown parameters and estimated. It is much more flexible than the other two options, since it enables identification of time delays in any arbitrary variable (directly measured or computed in the model). However, this option requires an estimation program capable of handling general nonlinear systems and a special procedure (subroutine) for generating the delay.^{9,10} The system model of Eqs. (3.9)–(3.11) can be modified to account for time delays as follows:

$$\dot{x}(t) = f[x(t), X(t, \tau), u(t) - \Delta u(\gamma_l), U(t, \tau), \beta] \quad x(t_0) = x_0(\alpha_l) \quad (3.19)$$

$$l = 1, 2, \dots, n_E$$

$$y(t) = g[x(t), X(t, \tau), u(t) - \Delta u(\gamma_l), U(t, \tau), \beta] \quad (3.20)$$

$$z(t_k) = y(t_k) + \Delta z(\delta_l) + v(t_k) \quad (3.21)$$

where x , u , y , and z are as before the state, input, output and measured variables, β the unknown system parameters, and Δu and Δz the zero shifts. The terms $X(t, \tau)$ and $U(t, \tau)$ denote the time delayed state and input variables given by:

$$[X(t, \tau)]_{ij} = x_i(t - \tau_j) \quad \text{and} \quad [U(t, \tau)]_{ij} = u_i(t - \tau_j) - \Delta u(\gamma_l) \quad (3.22)$$

The complete unknown parameter vector in this case is given by:

$$\Theta = [\beta^T \quad \tau^T \quad \alpha_1^T \quad \alpha_2^T \quad \dots \quad \alpha_{n_z}^T \quad \gamma_1^T \quad \gamma_2^T \quad \dots \quad \gamma_{n_z}^T \quad \delta_1^T \quad \delta_2^T \quad \dots \quad \delta_{n_z}^T]^T \quad (3.23)$$

where τ denotes the vector of time delays. It is obvious that only few selected variables may be associated with time delays. The presence of time delays affects the simulation and consequently parameter estimation as well. They must be accounted for to obtain accurate and reliable estimates.^{9,11}

V. Linearized Models

We are primarily concerned in this book with general nonlinear systems as represented in the preceding sections. In a few specific cases, however, we may consider simplified linear system representations. As already pointed out in Sec. I, such linear models are valid over small variations about the operating point. The system equations (3.1) and (3.2) can be linearized about some suitable operating point, leading to

$$\dot{x}(t) = Ax(t) + Bu(t) + Fw(t), \quad x(t_0) = x_0 \quad (3.24)$$

$$y(t) = Cx(t) + Du(t) \quad (3.25)$$

where the system matrices A , B , C , and D are respectively denoted by

$$\begin{aligned} A &= \frac{\partial f[x, u, \beta]}{\partial x}, & B &= \frac{\partial f[x, u, \beta]}{\partial u}, \\ C &= \frac{\partial g[x, u, \beta]}{\partial x}, & D &= \frac{\partial g[x, u, \beta]}{\partial u} \end{aligned} \quad (3.26)$$

The unknown system parameters β appearing in Eqs. (3.1) and (3.2) now appear as elements of these systems matrices, either directly or as their linear effects. The linearization point is usually chosen to be the trim point or estimated initial conditions. The measurement equation (3.3) is the same in this case, and hence is not repeated here. The state, input, and output variables appearing in Eqs. (3.24) and (3.25) are the perturbations around the trim conditions.

The formulation of Eqs. (3.5) and (3.6) accounting for the measurement biases, Δu and Δz , leads in the present case to:

$$\dot{x}(t) = Ax(t) + B[u(t) - \Delta u] + Fw(t), \quad x(t_0) = x_0 \quad (3.27)$$

$$y(t) = Cx(t) + D[u(t) - \Delta u] + \Delta z \quad (3.28)$$

For convenience, we have now equivalently included the measurement bias in Eq. (3.28) for y instead of that for z . Once again, the system of Eqs. (3.27)–(3.28) contains $(n_x + n_u + n_y)$ constant parameters consisting of biases and initial conditions. In contrast to the nonlinear systems discussed in Sec. III, linear model postulates are directly amenable to theoretical analysis using linear system theory.

A. Identifiability of Aerodynamic Derivatives and Constant Parameters

The identifiability of parameters is closely related to the concepts of observability and controllability.¹² In our case, we have to consider this issue separately for the system parameters (aerodynamic derivatives) and for the constant terms (initial conditions and biases). The identifiability of aerodynamic derivatives is determined by the information content in the data being analyzed. A unique component for each derivative must be available in the response data to enable its accurate estimation. We covered in Chapter 2 some of the important aspects of

performing flight maneuvers for this purpose. Concatenation of different flight maneuvers, as discussed therein and in Sec. III, may be necessary in most of the cases. Thus, the first part of this general issue is directly linked to the experiment.

With regard to the identifiability of constant parameters comprising measurement biases in the output and input variables, and initial conditions on the state variables, we treat linear and nonlinear models differently. In Sec. III, it has already been pointed out that, in the case of nonlinear models, such as those given by Eqs. (3.5)–(3.8), initial conditions have to be estimated; we will also address this issue in Sec. VII once again. Regarding the other bias parameters, because of correlation it may not be possible to estimate all of them, and the theoretical analysis being difficult we choose to make use of the estimation error correlation matrix to decide upon the appropriate choice of biases that can be estimated. On the other hand, linear models can be analyzed readily by applying the conventional observability analysis to determine the exact number of constant parameters that can be theoretically estimated. In Appendix B, it has been demonstrated that, for a linear model represented in Eqs. (3.27) and (3.28), a maximum of $(n_x + n_y)$ terms can be determined independently and not $(n_x + n_u + n_y)$ as ideally desired.

B. Models with Lumped Bias Parameters

Besides the just discussed limitation of being able to estimate only a subset of constant zero shifts and initial conditions, the linear model extension of Eqs. (3.27) and (3.28) is plagued with yet another undesirable effect. It leads to a model containing terms $B \cdot \Delta u$ and $D \cdot \Delta u$. Thus, the model is still linear in the state and control variables, but not in the parameters being estimated. This affects the convergence of the parameter estimation through optimization of some suitably defined cost function.

To overcome both these difficulties, we reformulate Eqs. (3.27) and (3.28) using the transformation $x^* = x - x_0$. It can be easily verified that the substitution of $x = x^* + x_0$ in these equations followed by simple manipulation and agreeing for convenience to still label x^* , the transformed x , as x , leads to an equivalent system representation of

$$\dot{x}(t) = Ax(t) + Bu(t) + b_x + Fw(t), \quad x(t_0) = 0 \quad (3.29)$$

$$y(t) = Cx(t) + Du(t) + b_y \quad (3.30)$$

The constant terms x_0 , Δu , and Δz now appear equivalently as lumped parameters $b_x (=Ax_0 - B\Delta u)$ and $b_y (=Cx_0 - D\Delta u + \Delta z)$. The transformed system is now linear in all the parameters. The initial conditions for simulation reduce to zero and, hence, are no longer unknown and as such not to be estimated. We now have exactly $(n_x + n_y)$ constant bias terms. Other transformations, such as $x^* = x - A^{-1}B\Delta u$, are also possible, but the one elaborated above is more efficient. Based on the aspects presented in the previous and this section, whenever we wish to estimate parameters of a linear system, we will formulate our equations according to Eqs. (3.29) and (3.30).

C. Numerical Approximation of System Matrices

Computing the derivatives of a function analytically is the most accurate approach, but not very convenient, particularly when dealing with complex and nonlinear systems. Numerical approximations are more convenient in such applications and found to work quite satisfactorily, provided we pay attention to the limitations of these approaches. Such procedures to numerically approximate functional derivatives or response gradients are commonly used to derive system matrices or in optimization algorithms.

The elements of the system state and control input matrices, A and B , are approximated using the central difference formula given by:

$$A_{ij} \approx \frac{f_i[x + \delta x_j e^j, u, \beta] - f_i[x - \delta x_j e^j, u, \beta]}{2\delta x_j}, \quad j = 1, 2, \dots, n_x \quad (3.31)$$

and

$$B_{ij} \approx \frac{f_i[x, u + \delta u_j e^j, \beta] - f_i[x, u - \delta u_j e^j, \beta]}{2\delta u_j}, \quad j = 1, 2, \dots, n_u \quad (3.32)$$

where e^j is a column vector with one in the j th row and zeros elsewhere and δx_j and δu_j are small perturbations in each of the n_x states and n_u control variables. The matrices C and D are similarly approximated from the observation function g . The truncation error of the centered formulas of Eqs. (3.31) and (3.32) is of the order of $O(h^2)$, where h is the perturbation. A proper choice of step size is critical to obtain valid and accurate approximations, because on one hand these increments have to be small enough to give valid approximation to the derivative, while on the other if they are too small then round-off errors may adversely affect the approximation. Some trade-off between round-off and truncation error is usually necessary. We will address this aspect in more detail in Chapter 4, Sec. VIII.

The higher-order centered formula having truncation error of the order of $O(h^4)$ is given by¹³

$$A_{ij} \approx \frac{-f_i[x + 2\delta x_j e^j] + 8f_i[x + \delta x_j e^j] - 8f_i[x - \delta x_j e^j] + f_i[x - 2\delta x_j e^j]}{12\delta x_j} \quad (3.33)$$

For notational simplicity, we have dropped the arguments u and β in Eq. (3.33). The advantage of the above fourth-order formula is that the truncation error goes to zero faster than that of the second-order formula. Hence, larger step sizes are possible. The main disadvantage is that it requires four function evaluations. For a detailed error analysis of the numerical approximation, the reader is referred to any standard text book on numerical computations.¹³

VI. Pseudo-control Inputs

During the 1970s a good number of estimation programs were developed based on the maximum likelihood estimation that we will address in the next

two chapters. Although these programs provided capabilities to estimate linear stability and control derivatives, analysis of flight data at high angles of attack and at extreme flight conditions required capabilities to handle nonlinear terms such as w^2 , V^2 , α^2 , $\alpha\delta_e$, qu , or $|\delta_a|$. The approach that was adopted then was that of the pseudo-control inputs.^{14–16} In this case, the control input vector u was augmented by additional terms, namely with those corresponding to the nonlinear terms, but now computed prior to estimation using the measured variables. These computed nonlinear terms were then treated as additional pseudo-inputs, and led to a system that was linear in these derived inputs.

The approach of pseudo-control inputs can also be used to reduce the model size, and yet account for the effects of other motion variables which are not treated as state variables. For example, in the decoupled model for longitudinal motion with the forward speed, vertical speed, pitch rate, pitch attitude, and pressure altitude (u, w, q, θ, h) as state variables, and elevator deflection, δ_e , as input, we can treat the variables pertaining to the lateral-directional motion as pseudo inputs. The converse is, of course, true whereby we can analyze the lateral-directional motion considering the typical variables of longitudinal motion, say angle of attack, as pseudo inputs. The approach of pseudo-control inputs, also applicable to linear models, relaxes the demands on performing flight maneuvers with minimum variations of the other mode of motion.

To illustrate the basic concept of pseudo-control inputs, we consider the following state equation:

$$\dot{u} = X_0 + X_u u + X_w w + X_{w^2} w^2 - qw + rv - g \sin \theta + \frac{F_e}{m} \cos \sigma_T \quad (3.34)$$

For the purpose at hand, without explaining each variable, it would suffice to make a note that Eq. (3.34) pertains to forward speed u in terms of the dimensional derivatives $X_{(.)}$, and involves nonlinear terms w^2, qw, rv and $g \sin \theta$. If we compute these nonlinear terms prior to estimation using the measured values of w, q, r, v , and θ , and denote them as $\sigma_1, \sigma_2, \sigma_3$, and σ_4 , respectively, then Eq. (3.34) can be rewritten as

$$\dot{u} = X_0 + X_u u + X_w w + X_{w^2} \sigma_1 - \sigma_2 + \sigma_3 - \sigma_4 + \frac{F_e}{m} \cos \sigma_T \quad (3.35)$$

Equation (3.35) is clearly linear in all the variables.

In this approach the basic linear representation is retained, therefore the estimation programs capable of handling linear models can be applied without any modifications. However, the measurement errors in the variables from which the nonlinear terms are computed propagate during estimation. The approach, therefore, yields biased estimates in the presence of noise and measurement errors. Moreover, it is limited to nonlinearities in variables for which the measurements are available. We will not pursue this approach any further, because we will arrive at estimation techniques which can handle general nonlinear systems more efficiently.

An alternative approach based on the so-called state vector augmentation is also possible to transform a nonlinear model into an equivalent linear model to

which an estimation program for linear systems could be applied.¹⁷ It is applicable when the nonlinearities are in terms of the state variables or when measured state variables contain significant noise and modeling errors, making the use of pseudo-controls questionable. In this approach the state vector is augmented through the nonlinear terms. Although the approach is more justified, it often involves modifications of the estimation program. These changes are required each time to incorporate the new sensitivity equations resulting from different nonlinear terms considered. The approach has rarely been used in practice, mainly because it is not flexible enough to investigate different nonlinear model postulates.

VII. Treatment of Initial Conditions

It has been clearly pointed out in Sec. V that, for linear systems, the model representation in terms of lumped bias parameters, Eqs. (3.29) and (3.30), is preferable, which reduces the initial conditions to be used in the simulation to zero. On the other hand, as mentioned in Sec. III, for nonlinear models it is necessary to specify and/or estimate the initial conditions explicitly. Accordingly, they are treated as part of the unknown parameter vector. Generally, the convergence of x_0 has poor asymptotic properties, because the information content is concentrated only at the beginning of the maneuver.² Loosely speaking, increasing the length of the segment does not necessarily add to the information useful for estimation of the initial conditions, at least for fast responding modes such as short period and Dutch roll. For slow motion modes, such as phugoid, longer duration records will be necessary.

For large models, for example six degrees of freedom aircraft motion with 10 states or combined rigid-body and dynamic models for reversible flight controls,¹⁸ requiring concatenation of multiple maneuvers leads to a large number of initial conditions, namely $(n_E \cdot n_x)$. This may far exceed the number of aerodynamic derivatives in which we are primarily interested. Although the identifiability of this large number of initial conditions is principally no problem, because each maneuver is treated separately and provides unique uncorrelated information, it increases the computational time disproportionately. To reduce the computational burden, a pragmatic approach would be to define the initial conditions appropriately and keep them fixed. In practice, provided measurements of adequate variables are available, the initial conditions are set to the first data point or average of first few data points to reduce the effects of noise and other errors. Keeping appropriately defined initial conditions fixed during the first few iterations of the optimization procedure and estimating the same in the last few iterations helps to speed up the process; the reduction in computational time can be significant for large-scale systems.

VIII. Simulation

As defined in Chapter 1, Sec. I, simulation is the process of reproducing (in technical terminology computing) numerically the system response to a given input for a prespecified system model. At this stage it is irrelevant to consider the model quality. It is always possible to carry out a simulation of a physically

existing process, or even of an unreal (anticipated) process. The model updates, fidelity and adequacy are addressed in the subsequent chapters on parameter estimation and model validation. Computation of system responses calls for integration of the state equations, which is usually performed by applying suitable numerical procedures. An adequate choice of the integration method is important for efficient and reliable parameter estimation.

In many of the engineering and scientific problems it is convenient to solve the initial-value problem through numerical approximations, because the exact solution may be theoretically possible but very complex, or the explicit solution may not exist. Therefore, we focus in this section on numerically integrating first-order ordinary differential equations. Starting from the specified initial conditions and neglecting the process noise, numerical integration over one sampling period yields state variables at each successive discrete point given by

$$x(t + \Delta t) = x(t) + \int_t^{t+\Delta t} f[x(t), u(t), \beta] dt \quad (3.36)$$

In our case, for stable aircraft a solution to Eq. (3.36) always exists. For unstable aircraft, integration of open loop plant may numerically diverge. How to deal with numerical divergence will be considered in Chapter 9, dealing specifically with unstable aircraft. Equation (3.36) applies to deterministic systems only. For a stochastic system of Eq. (3.1) or (3.5), process noise $w(t)$ being nonmeasurable we need a state estimator consisting of a prediction step which is the same as Eq. (3.36) and a correction step to improve the predicted states based on the process noise distribution matrix F and measurements z . We will address this problem in Chapters 5 and 7, and study different approaches of steady-state and time-varying filters. The state estimator includes in the first step of prediction the numerical integration methods that we apply to the deterministic system, and which we will study in this section. It is also relevant to make a specific mention that, for the purpose of parameter estimation, here we are concerned only with off-line simulation techniques. Real-time simulation, as necessary, for example, in the flight simulators, calls for different implementation, although the basic algorithms might be similar.

A. Numerical Integration Methods

The numerical integration methods can be classified into four general categories:^{13,19,20} 1) Taylor series methods, 2) Runge–Kutta Methods, 3) multistep methods, and 4) extrapolation methods. The multistep Adams–Bashforth–Moulton method is a two-step predictor–corrector procedure for higher accuracy requirements, but not suitable for our application since it is not self-starting. The extrapolation methods are particularly suitable for integrating over a large interval (time step) and when high precision is necessary. In the exercise that we are mainly concerned with, namely estimation of aerodynamic model from flight data, the sampling times are typically 20–100 ms, corresponding to sampling frequencies of 50 and 10 Hz. The accuracy requirements are moderate. A comparative study has demonstrated that in such cases the advantages of the extrapolation methods are not apparent, besides them being slower than the conventional

Table 3.1 Commonly applied numerical integration formulas

Method	Function evaluations	Solution
1 Euler method, first order	$f_1 = f(x, u, \beta)$	$x(k+1) = x(k) + f_1 \Delta t$
2 Heun method, second order	$f_1 = f(x, u, \beta)$ $f_2 = f(x + f_1 \Delta t, \bar{u}, \beta)$	$x(k+1) = x(k) + (f_1 + f_2) \Delta t / 2$
3 Runge–Kutta, second order	$f_1 = f(x, u, \beta)$ $f_2 = f[x + f_1(\Delta t/2), \bar{u}, \beta]$	$x(k+1) = x(k) + f_2 \Delta t$
4 Runge–Kutta, third order	$f_1 = f(x, u, \beta)$ $f_2 = f[x + f_1(\Delta t/3), \bar{u}, \beta]$ $f_3 = f[x + 2f_2(\Delta t/3), \bar{u}, \beta]$	$x(k+1) = x(k) + [f_1 + 3f_3] \Delta t / 4$
5 Runge–Kutta, fourth order	$f_1 = f(x, u, \beta)$ $f_2 = f[x + f_1(\Delta t/2), \bar{u}, \beta]$ $f_3 = f[x + f_2(\Delta t/2), \bar{u}, \beta]$ $f_4 = f(x + f_3 \Delta t, \bar{u}, \beta)$	$x(k+1) = x(k) + [f_1 + 2f_2 + 2f_3 + f_4] (\Delta t / 6)$

Runge–Kutta methods.²¹ Accordingly, we will briefly cover the simplest of the Taylor series method, namely Euler's method, to understand the principle of numerical integration and then most commonly applied Runge–Kutta formulas.

The numerical techniques provide a large number of integration formulas. Table 3.1 gives those which have found application on a routine basis in the aircraft parameter estimation. The five algorithms provide the solution to the initial value problem for systems with first-order ordinary differential equations, including retarded systems, that is, those with time delays. Since integration is across two data points, different procedures have been adopted while using control input u for function evaluations. The simplest is to use the same control input corresponding to the start of or the end of the interval for all the function evaluations, that is, one at the beginning, between, and at the end of the interval. Another approach is to use an average of the inputs at the two discrete points, as denoted by \bar{u} in Table 3.1. To achieve more accurate numerical results, it is also possible to use $u(t_k)$ and $u(t_{k+1})$ for function evaluations at the two ends of the interval and appropriately interpolated values (half, one-third or two-thirds) between $u(t_k)$ and $u(t_{k+1})$ for the intermediate steps. We will follow this procedure of interpolating the control inputs in the software that will be developed in Chapter 4.

The five methods listed in Table 3.1 are characterized by the number of function calls required to compute the solution at $x(t_{k+1})$ starting from the known (given) solution $x(t_k)$, where k is the discrete time point index. The Euler integration based on forward step requires one evaluation of the state derivative function f for each time point, whereas the Heun and Runge–Kutta second-, third- and fourth-order formulas require two, two, three and four evaluations, respectively. These methods differ not only in the number of function calls, but in the accuracy of the solution. The errors at the end of the interval for these methods is of the order of $O(h^2)$, $O(h^3)$, $O(h^3)$, $O(h^4)$, and $O(h^5)$, respectively, where $h (= \Delta t)$ is the integration interval.

The Euler method is the simplest one based on the gradient information at a single point (at the beginning of the integration interval), see Table 3.1. At each point just one function evaluation is necessary. It has limited accuracy and usage in parameter estimation, because the errors are propagated as the process progresses in time. The error accumulation also depends upon the number of data points and on the integration interval; the smaller the sampling time is, the slower the error propagation and vice-versa. Euler integration is rarely used, particularly for longer duration flight maneuvers and for sampling times greater than 20 ms because of adverse error propagation.

The Runge–Kutta formulas with constant step size provided in Table 3.1 are most widely applied in the offline simulation. They are adequate for moderate accuracy requirements, typical of rigid-body parameter estimation. Higher-order methods, possibly with step size control, will be useful for high bandwidth models, such as those for helicopter rigid-body mode extended with rotor degrees of freedom. It requires five or more function evaluations per time point depending upon the accuracy required. The computational load is directly proportional to the number of function calls, and, hence, it is obvious that the Euler integration is the fastest and fourth- or higher-order formula the slowest. On the other hand, higher-order integration formulas are more accurate. The higher the order of the formula, the more accurate the approximation is, but it requires more function evaluations and computational time. Depending upon the complexity of the problem, we may have to make a trade-off between speed and accuracy.

The choice of the integration method depends on the type of the system being investigated. Although the fourth-order Runge–Kutta method is generally recommended for rigid-body aircraft dynamics, this choice may be conservative.²² As a typical example, estimation of derivatives pertaining to the longitudinal motion from a two-point aerodynamic model accounting for wing–body and tail separately is considered. As seen from Table 3.2, the second- and third-order methods yield equally acceptable results compared with the more

Table 3.2 Estimates of aerodynamic derivatives applying different integration methods

Parameter	Euler	Second-order Runge–Kutta	Third-order Runge–Kutta	Fourth-order Runge–Kutta	Fifth-order Runge–Kutta–Fehlberg
C_{L0}	0.0682	0.0770	0.0775	0.0775	0.0775
C_{D0}	0.0232	0.0296	0.0297	0.0297	0.0297
C_{m0}	-0.1732	-0.1731	-0.1732	-0.1732	-0.1732
$C_{L\alpha}$	5.536	5.443	5.437	5.438	5.437
e	0.646	1.038	1.049	1.048	1.048
$C_{L\delta e}$	1.436	1.427	1.433	1.434	1.433
C_{m0WB}	10.362	11.39	11.23	11.23	11.23
$\text{Det}(R)^a$	2.32×10^{-9}	1.25×10^{-9}	1.25×10^{-9}	1.25×10^{-9}	1.25×10^{-9}
Iterations	7	5	5	5	5

^aCost function defined as determinant of the residual covariance matrix R .

time consuming fourth- and fifth-order Runge–Kutta formulas. The Euler integration, although it appears to converge to almost the same minimum, which may just be a coincidence, does show some changes in the numerical values, the convergence was affected as evident from the increase in the number of iterations. Moreover, the performance deteriorated for sampling times greater than 20 ms and propagation of errors was unacceptable, particularly when the flight maneuver was of longer duration. For systems characterized by higher-order dynamics, integration methods with step size control will be necessary to ensure that the errors are low and that they do not propagate adversely, which may affect the accuracy of the estimates and convergence of the optimization method.

For stiff systems, that is, those where all eigenvalues of the linearized system matrix have a negative real part and the largest and the smallest differ a lot, special integration methods are necessary because the system is characterized by a fast part (corresponding to a large eigenvalue) and also by a slowly decaying part (corresponding to a small eigenvalue). Typically, we encounter such systems, for example, in the modeling of elasto-viscoplastic deformations of metallic material.²² The Runge–Kutta algorithms in such cases encounter numerical problems and lead to unreliable results. Several algorithms which overcome these problems are available; Gear’s method, also called backward differentiation formula, is one of the most widely used.^{23,24} The primary limitation of such methods is that they are not applicable to systems with time delays. In general, special algorithms such as Gear’s method for stiff systems work inefficiently when applied to nonstiff systems.

For estimation of rigid-body aerodynamic models, the second-order Runge–Kutta algorithm is usually adequate and hence recommended during the initial iterations of the iterative estimation algorithms, switching over to the fourth-order Runge–Kutta only during final iterations. Such sophistications lead to speeding up of the overall estimation procedure for large-scale systems without affecting the final results. Here, we restrict ourselves to the use of a single algorithm throughout the iterative estimation procedure.

B. Integration of Linear Systems

In the case of linear systems, as represented in Eq. (3.29), there are two approaches to integrate the state equation. They are: 1) discretize the state equation and solve for x using the state transition matrix; and 2) apply one of the numerical integration methods covered in Sec. VIII.A. Using the discrete-time theory of linear dynamic systems, it can be shown that discretization of Eq. (3.29) leads to

$$x(t_{k+1}) = \Phi x(t_k) + \Psi B \bar{u}(t_k) + \Psi b_x \quad (3.37)$$

where $\Phi = e^{A\Delta t}$ is the state transition matrix (also called matrix exponential), and its integral will then be given by $\Psi = \int_0^{\Delta t} e^{A\tau} d\tau$, where $\Delta t (= t_k - t_{k-1})$ is the sampling time interval. For numerical computational purpose, the exponential of a matrix can be computed in many ways.²⁵ Here, we approximate Φ

through the Taylor series expansion of $e^{A\Delta t}$ given by

$$\Phi = e^{A\Delta t} \approx I + A\Delta t + A^2 \frac{\Delta t^2}{2!} + \dots \quad (3.38)$$

and the integral of Eq. (3.38) is given by:

$$\Psi = \int_0^{\Delta t} e^{A\tau} d\tau \approx I\Delta t + A \frac{\Delta t^2}{2!} + A^2 \frac{\Delta t^3}{3!} + \dots \quad (3.39)$$

Typically 8–10 terms of the Taylor series in Eqs. (3.38) and (3.39) are found to be adequate for accurate computational purposes.

Although the well-established state transition matrix approach was widely used in parameter estimation programs capable of handling linear systems only, during the last few decades more sophisticated programs have been developed capable of handling nonlinear models. Such programs invariably integrate the state equations using numerical integration methods. Since the same estimation program is applied to linear models as well, they will then be integrated using the chosen numerical integration method. Detailed investigation performed to evaluate the differences between the two approaches clearly showed that they are equivalent and yield the same estimates within the numerical accuracy of number representation. In general, it is preferable to have a single estimation program capable of handling both types of models, and from this view point there is little choice but to go for the methods covered in Sec. VIII.A.

IX. Concluding Remarks

In this chapter we have postulated general state space models and discussed several extensions to account for practical requirements of catering to 1) multiple experiments, 2) systematic errors in the measurements of control inputs and system outputs, and 3) time delays either in the direct output measurements or in the internal variables of the system. Issues related to identifiability of initial conditions and constant bias terms have been elucidated. The two approaches which were adopted in the past to account for nonlinear terms within the framework of linear estimation have been brought out, emphasizing that the advanced estimation techniques which have evolved over the last few decades, and as will be covered in this book, are more flexible and capable of handling the nonlinear systems directly. The equations of aircraft motion and aerodynamic model can be put within the framework of models postulated in this chapter. Hence, while developing the estimation methods in the next few chapters, model postulates as described in this chapter will form the basis for describing the flight vehicle, or dynamic system in general. Based on a brief discussion, it has been pointed out that multiple experiment analysis may involve estimation of a large number of initial conditions, far exceeding the actual number of aerodynamic derivatives in which we are mainly interested, and a pragmatic approach has been suggested to reduce the computational burden. Finally, different approaches to integrate both linear and nonlinear system have been presented.

In general, the classical Runge–Kutta fourth-order formula is commonly used for parameter estimation from flight data. However, substantiated by typical results obtained by applying the output error method, which will be studied in the next chapter, it is argued that even second- or third-order formulas may be sufficient in many cases. It leads to speeding up of the overall process, which is particularly useful while analyzing large scale systems. Aspects related to integration of stiff systems and limitations of such algorithms have been discussed briefly.

References

- ¹Etkin, B., *Dynamics of Atmospheric Flight*, John Wiley & Sons, New York, 1972.
- ²Maine, R. E. and Iliff, K. W., “Identification of Dynamic Systems—Applications to Aircraft Part 1: The Output Error Approach,” AGARD AG-300 Vol. 3, Part 1, Dec. 1986.
- ³Jategaonkar, R. V. and Plaetschke, E., “Maximum Likelihood Parameter Estimation from Flight Data for General Nonlinear Systems,” DFVLR-FB 83-14, March 1983.
- ⁴Maine, R. E. and Iliff, K. W., “Identification of Dynamic Systems,” AGARD AG-300, Vol. 2, Jan. 1985.
- ⁵Jategaonkar, R. V. and Plaetschke, E., “Algorithms for Aircraft Parameter Estimation Accounting for Process and Measurement Noise,” *Journal of Aircraft*, Vol. 26, No. 4, 1989, pp. 360–372.
- ⁶Mönlich, W., “Ein 2-Punkt-Aerodynamikmodell für die Identifizierung,” *Proceedings of the Symposium on “Systemidentifikation in der Fahrzeugdynamik”*, DFVLR-Mitt. 87-22, Paper No. 3.1, Nov. 1987 (in German).
- ⁷Jategaonkar, R. V., “Identification of the Aerodynamic Model of the DLR Research Aircraft ATTAS from Flight Test Data,” DFVLR-FB 90-40, July 1990.
- ⁸Mackie, D. B., “A Comparison of Parameter Estimation Results from Flight Test Data Using Linear and Nonlinear Maximum Likelihood Methods,” DFVLR-FB 84-06, Dec. 1983.
- ⁹Plaetschke, E., “Ein FORTRAN-Programm zur Maximum-Likelihood-Parameterschätzung in nichtlinearen retardierten Systemen der Flugmechanik—Benutzeranleitung,” DFVLR-Mitt. 86-08, Feb. 1986 (in German).
- ¹⁰Weiß, S., “NLHP1L: Ein Programm zur Maximum-Likelihood-Parameter-schätzung für nichtlineare Systeme—Benutzeranleitung,” DFVLR IB 111-87/29, July 1987 (in German).
- ¹¹Blackwell, J. and Feik, R. A., “Identification of Time Delays in Flight Measurement,” *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 1, 1991, pp. 132–139.
- ¹²Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.
- ¹³Mathews, J. H., *Numerical Methods for Mathematics, Science, and Engineering*, 2nd ed., Prentice Hall, Englewood Cliffs, NJ, 1992.
- ¹⁴McBrinn, D. E. and Brassell, B. B., “Aerodynamic Parameter Estimation for the A-7 Airplane at High Angles of Attack,” *Proceedings of the 3rd Atmospheric Flight Mechanics Testing Conference*, 1976, pp. 108–117.
- ¹⁵Maine, R. E., “Aerodynamic Derivatives for an Oblique Wing Aircraft Estimated From Flight Data by Using Maximum Likelihood Technique,” NASA TP 1336, Oct. 1978.
- ¹⁶Ross, A. J., “Identification Experience in Extreme Flight,” AGARD LS-104, Paper No. 8, Nov. 1979.
- ¹⁷Iliff, K. W., “Maximum Likelihood Estimation of Lift and Drag from Dynamic Aircraft Maneuvers,” *Journal of Aircraft*, Vol. 14, No. 12, 1977, pp. 1175–1181.

¹⁸Jategaonkar, R. V. and Mönnich, W., "Identification of Do-328 Aerodynamic Database for a Level D Flight Simulator," AIAA Paper 97-3729, Aug. 1997.

¹⁹Forsythe, G. E., Malcolm, M. A., and Moler, C. B., *Computer Methods for Mathematical Computations*, Prentice Hall, Englewood Cliffs, NJ, 1977.

²⁰Hairer, E., Nørsett, S. P., and Wanner, G., *Solving Ordinary Differential Equations I—Nonstiff Problems*, Springer, Berlin, 1987.

²¹Jategaonkar, R. V. and Plaetschke, E., "Non-Linear Parameter Estimation from Flight Test Data Using Minimum Search Methods," DFVLR-FB 83-15, March 1983.

²²Jategaonkar, R. V. and Thielecke, F., "Aircraft Parameter Estimation—A Tool for Development of Aerodynamic Databases," *Sadhana*, Indian Academy of Sciences, Vol. 25, Part 2, April 2000, pp. 119–135.

²³Gear, C. W., *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1971.

²⁴Radhakrishnan, K. and Hindmarsh, A. C., "Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations," NASA RP 1327, Dec. 1993.

²⁵Moler, C. and van Loan, C., "Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later," *SIAM Review*, Vol. 45, No. 1, 2003, pp. 3–49.

Chapter 4

Output Error Method

I. Introduction

HAVING DEALT in the previous chapter with the formulation of nonlinear state-space model postulates to define the cause–effect relationship purported to underlie the physical phenomenon, and having elaborated on the simulation (i.e., computation of system responses) using numerical procedures, we now turn our attention to the first of the two central methods of aircraft parameter estimation, namely the output error method. The other method, called the filter error method, will be discussed in the next chapter. Both of these methods belong to a general class of output error, also called response curve fitting, methods. The class of estimation methods called the equation error methods will be considered separately. In this class of output error methods, model parameters are adjusted iteratively to minimize the error between the measured variables (system output) and the estimated (model predicted) responses. The method, however, leads to a nonlinear optimization problem, in which the computational burden is relatively high. The method of weighted least squares, the simplest among this class, accounts for measurement noise. However, it assumes a priori specification of the weighting matrix. Based on probability theory, a more profound formulation called the maximum-likelihood principle was provided by Fisher.^{1–3} It can handle both process and measurement noise, and has several desirable statistical properties of a “good estimator.” This chapter considers the case where we assume that the process noise is negligible and that the measurements are corrupted by additive measurement noise only. It leads to the “output error method” (OEM). The output error as well as filter error methods represent a natural formulation for dynamic systems, be it a linear or nonlinear. In the majority of practical applications, the dynamic process being investigated evolves continuously in time, whereas the observations are recorded at discrete time points. Accordingly, we consider here the mixed continuous/discrete time formulation.

Since its introduction in the 1960s, the output error method is the most widely applied time-domain method to estimate aircraft parameters from flight data. This is evident from the numerous applications reported in the literature; Refs. 4–9 provide a good survey of the flight vehicle applications. Initially, during the 1960s to 1970s the scope of the output error method was limited to linear systems. The limitations were not related to the applicability of the maximum

likelihood principle to nonlinear systems; rather they were practical difficulties related to handling general nonlinear model structures and computing the sensitivity matrix required in the optimization of the cost function. In the 1980s, these difficulties were solved through a numerical approach, which has paved the way for analysis of general nonlinear systems of arbitrary complexity.⁶

In this chapter, we begin with the basic formulation of the maximum likelihood principle, the assumptions that are made and study briefly its statistical properties. This is followed by a detailed study of the cost function and various optimization methods. For each aspect, we indicate the pros and cons, and attempt to make recommendations based on our experience of modeling different types of flight vehicles.^{6,7} We study the performance of the output error method on two sample examples, using estimation software that we also supply with the book to enable the reader to follow the discussion and gain more insight into the algorithmic steps and numerical aspects. The data analyzed is also supplied as sample data to check the results and for possible adaptation of the software to individual needs.

II. The Principle of Maximum Likelihood Estimation

In the general theory of estimation founded by Fisher,^{1–3} it is assumed as a working hypothesis that the parent population has a known determinate mathematical form but for the numerical value of some parameter vector Θ (a column vector). The parent population is a set of all possible measurements as the number of samples N goes to ∞ . It is required to deduce the unknown value of the parameter vector Θ using a set of observations (z_1, z_2, \dots, z_N) .

Considering the given samples of N random observations (z_1, z_2, \dots, z_N) to be independent and taken from the same population, the “likelihood function” can be defined as:

$$\begin{aligned} p(z|\Theta) &= p(z_1|\Theta) \cdot p(z_2|\Theta) \cdots p(z_N|\Theta) \\ &= \prod_{k=1}^N p(z_k|\Theta) \end{aligned} \quad (4.1)$$

where $p(z|\Theta)$ is the probability of z given Θ . The maximum likelihood (ML) method, introduced by Fisher as a general estimation procedure, amounts to selecting that value of Θ within the admissible range such that $p(z|\Theta)$ is maximized. The likelihood function represents probability density, but the probability density of observed variables and not of the parameters. The unknown parameters are assumed not to depend on chance.^{10,11}

Because of the exponential nature of many density functions,¹² the logarithm of the likelihood function, which has the same optimal solution, is generally preferred. The maximum likelihood estimate is thus obtained as

$$\hat{\Theta}_{ML} = \arg \left\{ \max_{\Theta} \ln p(z|\Theta) \right\} \quad (4.2)$$

Assuming that $p(z|\Theta)$ is a twice differentiable function, practical application of the maximum likelihood method requires solution of the likelihood equation:

$$\frac{\partial \ln p(z|\Theta)}{\partial \Theta} = 0 \quad (4.3)$$

Equation (4.3), a column vector, is a set of nonlinear equations which can be solved only by successive approximation. Linear expansion of the likelihood equation about a first approximation Θ_0 of Θ yields:

$$\frac{\partial \ln p(z|\Theta_1)}{\partial \Theta} \approx \frac{\partial \ln p(z|\Theta_0)}{\partial \Theta} + \frac{\partial^2 \ln p(z|\Theta_0)}{\partial \Theta^2} \Delta \Theta \quad (4.4)$$

where $\Theta_1 = \Theta_0 + \Delta \Theta$ is the improved approximation to Θ . Equating Eq. (4.4) to zero yields a linear system of equations:

$$\frac{\partial^2 \ln p(z|\Theta_0)}{\partial \Theta^2} \Delta \Theta = -\frac{\partial \ln p(z|\Theta_0)}{\partial \Theta} \quad (4.5)$$

which can be solved by any procedure of linear algebra to yield the parameter improvement vector $\Delta \Theta$. The expected value of the matrix of second gradient $[\partial^2 \ln p(z|\Theta)/\partial \Theta^2]$ is called the Fisher information matrix.

In order to apply the maximum likelihood function to the dynamical system that we have considered in Chapter 3, it is now required to write the expression for $p(z|\Theta)$, that is, the conditional probability density function. Although the method is applicable to any form of the density function distribution, for mathematical tractability we consider the Gaussian (normal) distribution, which is completely determined by the first and second moments, that is, the mean and the covariance matrix. It is the most widely used assumption in practical cases. For such a case, and further assuming the error $v(t_k) = z(t_k) - y(t_k)$ at different time points t_k to be statistically independent, that is,

$$E\{v(t_k)v^T(t_\ell)\} = R\delta_{k\ell} \quad (4.6)$$

where $\delta_{k\ell}$ is the Kronecker delta symbol, ($\delta_{k\ell} = 1$ for $k = \ell$ and $= 0$ for $k \neq \ell$), it is shown in Appendix C that the likelihood function $p[z(t_1), \dots, z(t_N)|\Theta, R]$ of the n_y -dimensional measurement vector at N discrete time points for a given parameter vector Θ and given measurement error

covariance matrix R is^{13,14}

$$\begin{aligned}
 p[z(t_1), \dots, z(t_N) | \Theta, R] &= \prod_{k=1}^N p[z(t_k) | \Theta, R] \\
 &= \{(2\pi)^{n_y} |R|\}^{-N/2} \exp \left[-\frac{1}{2} \sum_{k=1}^N [z(t_k) - y(t_k)]^\top \right. \\
 &\quad \times \left. R^{-1} [z(t_k) - y(t_k)] \right]
 \end{aligned} \tag{4.7}$$

The reader may note that we have implicitly extended here the basic definition of Eq. (4.1) to include the conditioning on R , the covariance matrix of the residuals. For notational simplicity we denote:

$$p(z | \Theta, R) = p[z(t_1), \dots, z(t_N) | \Theta, R] \tag{4.8}$$

We pay specific attention here to the commonly misunderstood concept and interpretation of the exact meaning of the maximum likelihood function. The likelihood function $p(z | \Theta, R)$ does not represent the probability distribution of the unknown parameters Θ , but of the measurements z . Although looking at the observations some parameters may appear more plausible and some less, the unknown parameters are not random variables and do not have probability density. Maximum likelihood estimation means that the Θ -vector is searched, which maximizes the function $p(z | \Theta, R)$. Such a vector is “the most plausible,” because it gives the highest probability to the measurements.

From Eq. (4.7) we can immediately compute the maximum likelihood estimates of the unknown parameters Θ and of the measurement noise covariance matrix R by setting the first derivatives with respect to each of them to zero. However, an equivalent and more practicable solution to the optimization problem is the minimization of $L(z | \Theta, R)$, the negative logarithm of the likelihood function:

$$\begin{aligned}
 L(z | \Theta, R) &= \frac{1}{2} \sum_{k=1}^N [z(t_k) - y(t_k)]^\top R^{-1} [z(t_k) - y(t_k)] \\
 &\quad + \frac{N}{2} \ln[\det(R)] + \frac{N n_y}{2} \ln(2\pi)
 \end{aligned} \tag{4.9}$$

We prefer to deal with Eq. (4.9) instead of Eq. (4.7), because, as already mentioned, the natural logarithm of the normal density function is a simpler function than the density function itself. The differentiation result $\partial(\ln p)/\partial\Theta = 1/p \cdot \partial p/\partial\Theta$ guarantees that the derivative of log of p is zero, when $\partial p/\partial\Theta$ is zero, and thus yields the same results.

III. Properties of Maximum Likelihood Estimates

Solving the likelihood equation yields parameter estimates $\hat{\Theta}_{ML}$ that lend the largest probability to the observations (z_1, z_2, \dots, z_N) , which are assumed to be statistically independent. Although the maximum-likelihood principle is intuitively appealing, we will study here briefly the statistical properties of the estimates. For a more detailed treatment of these statistical properties, the reader may refer to Appendix D or any one of the standard books on statistics.^{10–17}

The concepts of efficiency and consistency of the maximum likelihood estimates were introduced by Fisher.³ Based on the assumption of independent observations, Wald established the asymptotic consistency and unbiasedness of the maximum likelihood estimates.¹⁵ The asymptotic normality was proved by Cramér.¹⁶ We recapture here only the essentials of a “good” estimator.¹⁸

- 1) The maximum likelihood estimates are asymptotically unbiased, that is,

$$\lim_{N \rightarrow \infty} E(\hat{\Theta}_{ML}) = \Theta$$

where Θ represents the true values of the parameters.

2) The maximum likelihood estimates $\hat{\Theta}_{ML}$ are asymptotically consistent, that is, $\hat{\Theta}_{ML}$ converges in probability to the true value Θ .

3) The maximum likelihood estimates $\hat{\Theta}_{ML}$ obtained from different sets of data samples are asymptotically normally distributed around the true value Θ , that is,

$$\sqrt{N} \cdot (\hat{\Theta}_{ML} - \Theta) \longrightarrow r_1 \sim \mathcal{N}(0, \bar{\mathcal{F}}^{-1})$$

where r_1 is the random variable, $\bar{\mathcal{F}}$ the average Fisher information matrix per sample, and $\mathcal{N}(0, \bar{\mathcal{F}}^{-1})$ the normal (Gaussian) distribution with zero mean and variance $\bar{\mathcal{F}}^{-1}$.

4) The maximum likelihood estimates $\hat{\Theta}_{ML}$ are asymptotically efficient in the sense that they attain the Cramér–Rao lower bounds, which is a measure of achievable statistical accuracy.

The property of consistency implies that the maximum likelihood estimates converge in probability to the true values. The property of asymptotic normality implies that the estimates obtained from different sets of data samples corresponding to different experiments are clustered around the true value with a normal distribution. The property of asymptotic efficiency is of practical significance. It implies that the maximum likelihood estimator makes efficient use of the available data. The Cramér–Rao lower bound indicates the theoretically maximum achievable accuracy of the estimates; see Appendix D.

IV. The Maximum Likelihood Function for Estimation of Parameters in Dynamic Systems

The mathematical model of the dynamic system whose parameters are to be estimated is assumed to be described by the following general nonlinear system representation:

$$\dot{x}(t) = f[x(t), u(t), \beta], \quad x(t_0) = x_0 \quad (4.10)$$

$$y(t) = g[x(t), u(t), \beta] \quad (4.11)$$

$$z(t_k) = y(t_k) + Gv(t_k) \quad (4.12)$$

This is the same set of equations that we had postulated in Chapter 3 to represent a dynamic system in state space. All the variables have already been defined there. To recollect, the nonlinear functional relationships f and g are assumed to be real-valued. They may be nonlinear in the state variables x , and/or in the control variables u , as well as in the parameters to be estimated.^{18–21} This class of nonlinear system description is in general adequate for various estimation problems encountered in flight vehicle modeling from flight data.^{4–6}

In order to apply the likelihood function to estimate the parameters of a dynamic system represented in Eqs. (4.10)–(4.12), we make the following assumptions:

- 1) The input sequence $[u(t_k), k = 1, 2, \dots, N]$ is exogenous, that is, generated independent of the system output.
- 2) The measurement errors $v(t_k) = z(t_k) - y(t_k)$ at different discrete time points are statistically independent. They are assumed to be distributed with zero mean and covariance matrix R , that is,

$$E\{v(t_k)\} = 0,$$

$$E\{v(t_k)v^T(t_\ell)\} = R\delta_{k\ell}$$

- 3) The system is corrupted by measurement noise only.
- 4) Control inputs $u(t_k)$ are sufficiently and adequately (i.e., in magnitude and frequency) varied to excite directly or indirectly the various modes of the dynamical system being analyzed.

By definition, exogenous inputs are those that affect the system, but are not affected by the system itself. It is the key to success of parameter estimation, and also helps to ensure that the residuals are independent at different time points. For stable aircraft, it is usually possible to apply control inputs about all axes independent of the aircraft response. However, in the case of unstable aircraft, owing to feedback this assumption may not be met and hence special treatment is necessary. We will discuss problems and solutions for unstable aircraft in a separate chapter later.

The second assumption allows the likelihood function that we have discussed to be applied directly. The third assumption implies that we treat in this chapter the dynamic system as deterministic. Note that we have dropped the process noise w and the process noise distribution matrix F in Eq. (4.10). For such cases, the state variables are obtained through simple integration of the state equations. The case of process noise will be dealt with in the next chapter. Through the fourth assumption, we ensure identifiability of the parameters, because the basic rule “If it is not in the data, it can not be identified” applies to all exercises that attempt parameter estimation from experimental data. This is true irrespective of the type of flight vehicle we may attempt to model.

In the discussion of Secs. II and III, the likelihood function $p(z|\Theta, R)$ included the arguments z , Θ , and R . Strictly speaking, mathematically it would be more appropriate to write the likelihood function for the given data as $p(z|\Theta, R, u)$, where we have now included the argument u for the system inputs. However, the argument u is dropped in the further discussions without loss of generality, because 1) identical control inputs are used to excite the system as well as the postulated model, see Fig. 4.1, and 2) the control inputs are assumed to be exogenous, noise-free and known *a priori*.

From Sec. II, we know that maximum likelihood estimates are obtained by minimization of Eq. (4.9), the negative logarithm, $L(z|\Theta, R)$, of the likelihood function. For convenience we rename $L(z|\Theta, R)$ as J , the cost function to be minimized:

$$\begin{aligned} J(\Theta, R) = L(z|\Theta, R) &= \frac{1}{2} \sum_{k=1}^N [z(t_k) - y(t_k)]^T R^{-1} [z(t_k) - y(t_k)] \\ &\quad + \frac{N}{2} \ln[\det(R)] + \frac{Nn_y}{2} \ln(2\pi) \end{aligned} \quad (4.13)$$

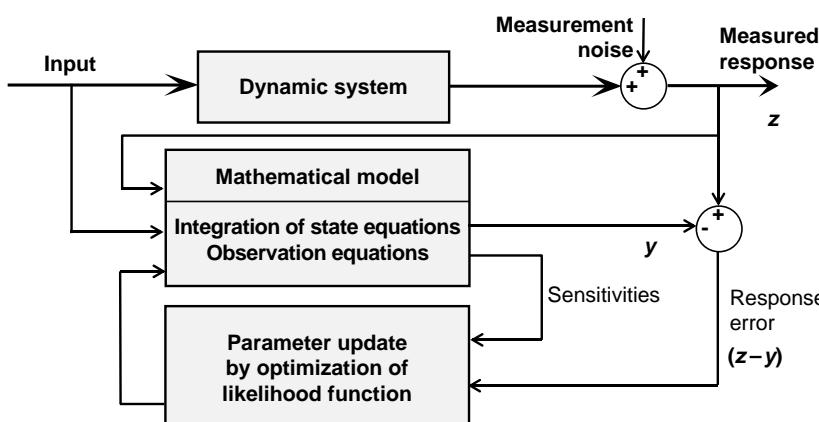


Fig. 4.1 Block schematic of output error method.

subject to the system model postulated in Eqs. (4.10)–(4.12). Figure 4.1 shows a block schematic of the output error method. The details of each of the blocks in this figure will be discussed hereafter. Note that the system responses y are functions of the parameters Θ , and hence the exact notation of the above equation should include $y(\Theta)$. However, for the sake of brevity, this dependence is not shown explicitly in any of the equations that we deal with.

V. Basics of Cost Function Optimization

Before we turn to actual methods of optimization, we further analyze the likelihood cost function $J(\Theta, R)$ to gain more insight.^{22,23} For a postulated model and data set being analyzed, the number of observation variables n_y and the number of data points N are fixed. Thus, the last term in Eq. (4.13) is a constant and hence neglected in the optimization without affecting the results. For the first two terms on the right hand side of Eq. (4.13), there are two cases which are of interest depending upon whether the measurement error covariance matrix R is known or not.

A. Known Measurement Noise Covariance Matrix

In the first case, we assume that the measurement noise covariance matrix R is known; at this stage it is immaterial as to how we might arrive at the knowledge of R . In such a case, since R is assumed to be known, the second term in Eq. (4.13), $N \ln[\det(R)]/2$, is constant. The cost function then reduces to

$$J(\Theta) = \frac{1}{2} \sum_{k=1}^N [z(t_k) - y(t_k)]^T R^{-1} [z(t_k) - y(t_k)] \quad (4.14)$$

To understand the nature of Eq. (4.14), let us consider a simpler case of having just two observation variables and assume R to be a diagonal matrix. In such a case, multiplying out the terms in the summation on the right hand side leads to $J(\Theta) = \sum [r_{11}e_1^2(t_k) + r_{22}e_2^2(t_k)]$, where e_1 and e_2 denote the response errors in the two observation variables, and r_{11} and r_{22} the diagonal elements of R representing the variances. Thus, the cost function $J(\Theta)$ in Eq. (4.14) is nothing but the weighted sum of squares of the response error and is quadratic in nature. We can apply any one of the optimization methods we will discuss in the following sections.

B. Unknown Measurement Noise Covariance Matrix

In the other case of unknown R , optimization of Eq. (4.13) is a little more involved. The brute-force method would be to include the elements of the covariance matrix in the unknown parameters vector Θ , and then apply one of the optimization methods. This approach is, however, never practiced, because there is no closed form solution to this minimization problem. During the optimization, the estimates of the system parameters depend on the measurement noise covariance matrix and vice versa. Any attempt in this direction is bound to fail. To overcome this difficulty, we use the relaxation strategy in which optimization of the likelihood function, Eq. (4.13), is carried out in two steps.

OUTPUT ERROR METHOD

87

In the first step, it can be shown that, for any given value of the parameter vector Θ , the maximum likelihood estimate of R is given by

$$R = \frac{1}{N} \sum_{k=1}^N [z(t_k) - y(t_k)][z(t_k) - y(t_k)]^T \quad (4.15)$$

Equation (4.15) is obtained by partial differentiation of Eq. (4.13) with respect to R and setting it to zero. A detailed derivation of the above expression for R is provided in Appendix E.

Having obtained a maximum likelihood estimate of R , substitution of Eq. (4.15) in (4.13) yields

$$J(\Theta) = \frac{1}{2} n_y N + \frac{N}{2} \ln[\det(R)] + \frac{Nn_y}{2} \ln(2\pi) \quad (4.16)$$

Once again recall that n_y and N are fixed for a postulated model and a data set being analyzed. Hence, in addition to the last term, the first term on the right hand side of Eq. (4.16) also turns out to be a constant and can be neglected without affecting the minimization results. Thus, the cost function reduces to:

$$J(\Theta) = \det(R) \quad (4.17)$$

Determination of the parameter vector Θ which minimizes $\det(R)$, or equivalently the function $L(z|\Theta)$, is an optimization problem which can be solved by applying different methods.

Thus, the relaxation procedure can be summarized as follows:

- 1) Choose suitable initial values for Θ .
- 2) Compute system outputs y and the residuals $(z - y)$; estimate the measurement noise covariance matrix R .
- 3) Minimize $J(\Theta)$ with respect to Θ by applying one of the nonlinear optimization methods.
- 4) Iterate on step 2 and check for convergence.

Mathematical proof of a global convergence of the relaxation procedures is difficult, but it is convenient to use and mostly works well in practice. From the abundant examples pertaining to flight vehicles of diversified complexity reported in the literature, it is apparent that this is the only pragmatic approach to nonlinear optimization.

In general, the covariance matrix of the output errors is unknown and, hence, has to be estimated. In such a case, as shown above, the cost function to be minimized reduces to the determinant of the covariance matrix of the output errors. Since the covariance matrix is usually assumed to be diagonal, a simplification that has found widespread use in practice, the cost function then represents nothing but the product of the variances. Any optimization method, direct search or gradient based, can be applied to obtain the estimates of Θ . In any case, it is necessary to compute the system states x and responses y to obtain the cost function. Furthermore, depending upon the optimization algorithm, response gradients may be required.

VI. Gauss–Newton Algorithm

The necessary condition for minimization of the likelihood function with respect to the unknown parameters is given by

$$\frac{\partial J(\Theta)}{\partial \Theta} = 0 \quad (4.18)$$

The Taylor series expansion of $\partial J / \partial \Theta$ about the i th value of the parameter vector Θ , truncated after two terms, is given by

$$\left(\frac{\partial J}{\partial \Theta} \right)_{i+1} \approx \left(\frac{\partial J}{\partial \Theta} \right)_i + \left(\frac{\partial^2 J}{\partial \Theta^2} \right)_i \Delta \Theta \quad (4.19)$$

where $\Delta \Theta = \Theta_{i+1} - \Theta_i$ is the parameter change and $(\partial^2 J / \partial \Theta^2)_i$ the second gradient of the cost function with respect to Θ at the i th iteration. In optimization theory, the matrix of second gradients is termed Hessian. Now, using the necessary condition of Eq. (4.18) for the minimum, we set the right-hand side of Eq. (4.19) to zero and solve for $\Delta \Theta$:

$$\Delta \Theta = - \left[\left(\frac{\partial^2 J}{\partial \Theta^2} \right)_i \right]^{-1} \left(\frac{\partial J}{\partial \Theta} \right)_i \quad (4.20)$$

The above change in Θ on the $i+1$ th iteration makes local $(\partial J / \partial \Theta)_{i+1}$ approximately zero. Thus, starting from an initial guess value Θ_0 , $\Theta_{i+1} = \Theta_i + \Delta \Theta$ provides an iterative solution to find the minimum of a function. Such an algorithm is commonly known as the Newton–Raphson method.^{24,25}

To demonstrate the above optimization procedure, we consider a simple one-dimensional case. First, we apply the above procedure to find θ that minimizes a quadratic cost function $J(\theta) = 8\theta^2$. Let the starting value of θ be 4. The first and second gradients, $\partial J / \partial \theta$ and $\partial^2 J / \partial \theta^2$, of this cost function evaluated at $\theta_0 = 4$ are given by $16\theta_0 = 16 \times 4$ and 16 respectively. Equation (4.20) then yields an incremental value of $\Delta \theta = -(16)^{-1} \times 16 \times 4 = -4$. Thus, after the first iteration the iterative update leads to $\theta_1 = \theta_0 + \Delta \theta = 4 - 4 = 0$. For the quadratic cost function, this solution happens also to be the minimum, and does not require any further iteration. This is always the case for any arbitrary quadratic function, because the second gradient $\partial^2 J / \partial \theta^2$ is constant, which allows, as illustrated schematically in Fig. 4.2a, determination of the minimum in a single step irrespective of the starting value θ_0 . This simple test case can be run using the program “/FVSysID/chapter04/ExQuadFun.m.”

In the second example we consider optimization of a nonquadratic function $J(\theta) = 4\theta^2 + 0.4\theta^3$. The first and the second gradients are given by $\partial J / \partial \theta = 8\theta + 1.2\theta^2$ and $\partial^2 J / \partial \theta^2 = 8 + 2.4\theta$, respectively. The second gradient in this case is not a constant, but is a function of θ . The optimization procedure yields step by step an improved solution given by Eq. (4.20) based on the local quadratic approximation of the nonquadratic cost function. For example, starting from $\theta_0 = 4$, the first iteration yields the increment

OUTPUT ERROR METHOD

89

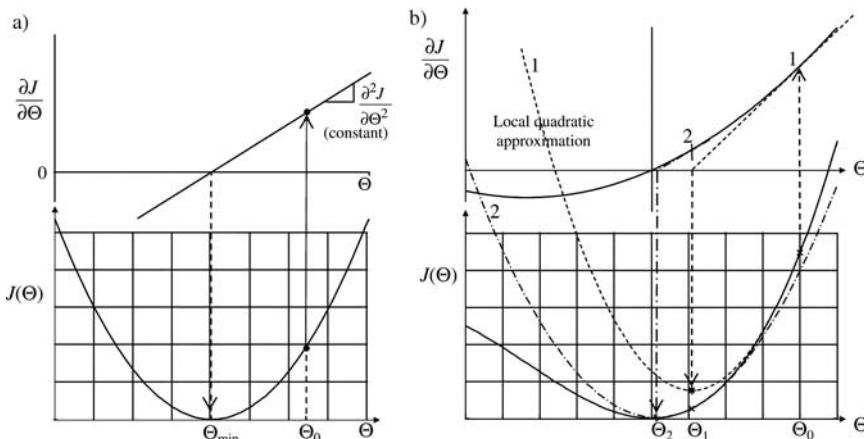


Fig. 4.2 Schematic of optimization procedure. a) Quadratic cost function, b) nonquadratic cost function.

$\Delta\theta_1 = -(8 + 2.4 \times 4)^{-1}(8 \times 4 + 1.2 \times 4^2) = -2.9091$, and the updated parameter $\theta_1 = \theta_0 + \Delta\theta_1 = 4 - 2.9091 = 1.0909$. Now, starting from the updated value θ_1 the second iteration leads to $\Delta\theta_2 = -(8 + 2.4 \times 1.0909)^{-1}(8 \times 1.0909 + 1.2 \times 1.0909^2) = -0.9564$, yielding the updated parameter value of $\theta_2 = \theta_1 + \Delta\theta_2 = 1.0909 - 0.9564 = 0.1345$. The iterative procedure continues until the minimum is achieved after the fourth iteration. As already pointed out, in each iteration the first and second gradients change and have to be computed for the local value of θ . The optimization procedure is schematically illustrated in Fig. 4.2b. This test case can be run using the program “/FVSysID/chapter04/ExNonQuadFun.m.” In general, away from the minimum of the cost function, the algorithm takes larger incremental steps corresponding to larger values of the gradient, whereas they become smaller and smaller as the minimum is approached; the number of steps required to find the minimum in a specific case will depend upon the shape of the cost function. If the cost function has multiple minima, depending upon the initial starting values the search procedure may find a local extremum.

In our case of parameter estimation from flight data, the partial differentiation of the likelihood function, Eq. (4.13) results in

$$\frac{\partial J}{\partial \Theta} = -\sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} [z(t_k) - y(t_k)] \quad (4.21)$$

and the partial differentiation of $\partial J / \partial \Theta$ with respect to Θ yields

$$\frac{\partial^2 J}{\partial \Theta^2} = \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \frac{\partial y(t_k)}{\partial \Theta} + \sum_{k=1}^N \left[\frac{\partial^2 y(t_k)}{\partial \Theta^2} \right]^T R^{-1} [z(t_k) - y(t_k)] \quad (4.22)$$

The computation of the first gradient $\partial J / \partial \Theta$ from Eq. (4.21) is relatively straightforward. It requires just the measured data z , the corresponding computed variables y , and the response gradient $\partial y / \partial \Theta$. Recall that the estimate of the weighting matrix R , given by Eq. (4.15), is already obtained for the current Θ .

Computation of the second gradient $\partial^2 J / \partial \Theta^2$ given by Eq. (4.22) is more complex and time-consuming, as it requires the second gradient of the response $\partial^2 y / \partial \Theta^2$. However, the second term on the right-hand side of Eq. (4.22) includes a term $[z(t_k) - y(t_k)]$, and it is found that this contribution goes to zero as the process converges. In an ideal case, the residuals $[z(t_k) - y(t_k)]$ should just be the random measurement error at each time point. Based on our assumption of zero mean and independent noise, the second term tends to cancel out when summed over a sufficient length of data points. Based on this practical consideration, we neglect the second term in Eq. (4.22) and approximate the second gradient $\partial^2 J / \partial \Theta^2$ as:

$$\frac{\partial^2 J}{\partial \Theta^2} \approx \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \left[\frac{\partial y(t_k)}{\partial \Theta} \right] \quad (4.23)$$

This simplification was suggested by Balakrishnan²⁶ and leads to a more tractable algorithm called the modified Newton–Raphson method, which works well in most of the practical applications. In the literature this algorithm is also sometimes called the Newton–Balakrishnan algorithm or the Gauss–Newton method, the terminology that we will follow in this book.

VII. Method of Quasi-linearization

In this section we follow a slightly different approach to derive an expression to compute the parameter update. We already know from Eq. (4.18) that $\partial J / \partial \Theta$ must vanish at the minimum. Accordingly, setting Eq. (4.21) to zero yields

$$\frac{\partial J}{\partial \Theta} = - \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} [z(t_k) - y(t_k)] = 0 \quad (4.24)$$

Now, applying the quasi-linearization,^{27,28} that is, a first-order approximation, to the system responses around some nominal value Θ_0 , we obtain:

$$y(\Theta) = y(\Theta_0 + \Delta\Theta) \approx y(\Theta_0) + \frac{\partial y}{\partial \Theta} \Delta\Theta \quad (4.25)$$

Substituting the linearized $y(\Theta)$ in Eq. (4.24) leads to:

$$\frac{\partial J}{\partial \Theta} = - \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \left[z(t_k) - \left\{ y(t_k) + \frac{\partial y(t_k)}{\partial \Theta} \Delta\Theta \right\} \right] = 0 \quad (4.26)$$

Note that for convenience, we have dropped the argument Θ_0 from y in the above equation. Equation (4.26) can be rewritten as:

$$\sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} [z(t_k) - y(t_k)] - \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \left[\frac{\partial y(t_k)}{\partial \Theta} \right] \Delta \Theta = 0 \quad (4.27)$$

which leads to a system of linear equations:

$$\sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \left[\frac{\partial y(t_k)}{\partial \Theta} \right] \Delta \Theta = \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} [z(t_k) - y(t_k)] \quad (4.28)$$

Equation (4.28) derived applying quasi-linearization is exactly the same as Eqs. (4.21) and (4.23) derived for the Gauss–Newton method. Thus, the Gauss–Newton and quasi-linearization methods are equivalent.

The Gauss–Newton and quasi-linearization methods lead to a system of linear equations, which can be represented in a general form as

$$\Theta_{i+1} = \Theta_i + \Delta \Theta, \quad \text{and} \quad \mathcal{F} \Delta \Theta = -\mathcal{G} \quad (4.29)$$

where i is the iteration index and \mathcal{F} and \mathcal{G} are given by

$$\mathcal{F} = \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \left[\frac{\partial y(t_k)}{\partial \Theta} \right]; \quad \mathcal{G} = - \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} [z(t_k) - y(t_k)] \quad (4.30)$$

\mathcal{F} is the information matrix (also called Hessian), \mathcal{G} is the gradient vector and $\Delta \Theta$ the parameter change vector. Once the gradient vector \mathcal{G} and the matrix of second gradients \mathcal{F} are computed, the parameter updates are given by Eq. (4.29). Thus, we now have both the steps of the relaxation algorithm which can be carried out successively, namely compute the measurement noise covariance matrix using Eq. (4.15), and then apply the Gauss–Newton method given by Eqs. (4.29) and (4.30) to update the parameters. Equation (4.29), $\mathcal{F} \Delta \Theta = -\mathcal{G}$, can be solved for $\Delta \Theta$ by any standard algorithm from linear algebra. Owing to inaccurate numerical results, direct matrix inversion methods are rarely used. It is obvious from Eq. (4.30) that the information matrix \mathcal{F} is symmetric, positive-definite. For such cases the Cholesky factorization is an efficient procedure. Singular value decomposition is yet another approach that is also efficient. Details of these techniques can be found in any standard text book on numerical methods.

VIII. System Response and Sensitivity Coefficients

The iterative update of the parameter vector Θ by Gauss–Newton or quasi-linearization method using Eqs. (4.29) and (4.30) requires 1) computation of the system responses $y(t_k)$, and 2) computation of the response gradients $\partial y(t_k)/\partial \Theta$. Efficient implementation of these computational aspects, together

with flexibility to handle conveniently different model structures, is important for parameter estimation in nonlinear systems.^{20,21,29}

The model predicted response $y(t)$, Eq. (4.11), is a function of the parameters being estimated and the state variables $x(t)$. According to our assumption of the system being corrupted by measurement noise only and that the measured control inputs contain no errors and noise, we are dealing in this chapter with deterministic systems. For such a case we can simply integrate the state equations, Eq. (4.10), using one of the numerical integration procedures elaborated in Chapter 3, Sec. VIII.

Computation of the response gradients $\partial y(t_k)/\partial \Theta$, also called the sensitivity matrix, involves some critical issues that directly affects the capability to handle different nonlinear models. In the conventional approach, the sensitivity matrix $\partial y/\partial \Theta$ is obtained by solving the sensitivity equations, which are obtained analytically by partial differentiation of the system equations (4.10) and (4.11):

$$\frac{\partial \dot{x}}{\partial \Theta} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial \Theta} + \frac{\partial f}{\partial \Theta} \quad (4.31)$$

$$\frac{\partial y}{\partial \Theta} = \frac{\partial g}{\partial x} \frac{\partial x}{\partial \Theta} + \frac{\partial g}{\partial \Theta} \quad (4.32)$$

In the case of linear systems, the method of transition matrices for the solution of Eq. (4.31) and simple matrix multiplications provide an elegant procedure. However, if the system model is nonlinear, as in Eqs. (4.10) and (4.11), any changes in the model structure entail re-derivation of the sensitivity equations of the form given in Eqs. (4.31) and (4.32). As a consequence, changes in the parameter estimation software are necessary. This additional programming effort becomes cumbersome, especially when alternative model structures are to be tried out to fit the experimental data. Furthermore, in case of systems with discontinuous nonlinearities, numerical problems may be encountered owing to the difficulty of defining the derivatives analytically. These are the practical difficulties arising out of the nonlinear model postulate and the need for flexibility of handling different model structures conveniently without software changes.²⁹

The need for explicit derivation of the sensitivity coefficients, Eqs. (4.31) and (4.32), can be eliminated by approximating the sensitivity coefficients by numerical differences. The forward difference approximation yields each element of the response gradient matrix required:

$$\left[\frac{\partial y(t_k)}{\partial \Theta} \right]_{ij} \approx \frac{y_i^p(t_k) - y_i(t_k)}{\delta \Theta_j}; \quad i = 1, \dots, n_y; j = 1, \dots, n_q$$

$$\approx \frac{g_i[x^p(t_k), u(t_k), \Theta + \delta \Theta_j e^j] - g_i[x(t_k), u(t_k), \Theta]}{\delta \Theta_j} \quad (4.33)$$

where n_y is the number of output variables and n_q the total number of unknown parameters; superscript p denotes perturbed variables, $\delta \Theta_j$ a small perturbation in the j th component of Θ , e^j a column vector with one in the j th row and zeros elsewhere, $y^p(t_k)$ the perturbed response variables corresponding to the perturbation

$(\Theta + \delta\Theta_j e^j)$, and $y(t_k)$ the unperturbed responses corresponding to Θ . The perturbed responses $y^p(t_k)$ at each time point are obtained from the perturbed system equations. These equations have a form similar to those of Eqs. (4.10) and (4.11), except that each time the parameter vector Θ is replaced through the perturbed parameter vector Θ_p . The reader can easily write these equations. The perturbed responses $y^p(t_k)$ are also functions of the perturbed states $x^p(t_k)$, which are once again obtained by numerically integrating the perturbed state equations, which are basically the same state equations, except for a different set of parameter values. Thus, we use the same model and thereby avoid the changes to the estimation software.

Thus, the finite difference approximation of the response gradients provides a more flexible approach to handling nonlinear models. The sensitivity coefficients in Eq. (4.33) were approximated by forward differences. A more accurate method would be based on the two-sided, central difference formula. However, it requires two evaluations of the perturbed states and of the response variables at each time point, and needs twice the computational time. From our experience, the one-sided forward difference approximation works fairly well in most cases. For large-scale problems (i.e., large numbers of outputs n_y , parameters, n_q , and data points N), near the minimum of the cost function, a better performance may be obtained through central differencing.³⁰ The other alternative that also provides good results in such cases is forward difference approximation in conjunction with line search, which we will discuss in Sec. X.B, to overcome stalling of the optimization, or applying the Levenberg–Marquardt method, which we will address in Sec. XIII.

The choice of the parameter perturbation will affect the accuracy of the approximated sensitivity coefficients. Some care is necessary in choosing the perturbation size; it should be small enough to give a valid approximation of the true gradients, but too small a value may pose difficulties due to machine accuracy and round-off errors. We have already discussed this aspect of trade-off in Chapter 3, Sec. V.C in connection with approximating system matrices. Since the parameters may vary in the order of their magnitudes, a relative parameter perturbation is more appropriate. Typically,

$$\delta\Theta_j = 10^{-6} \Theta_j \quad (4.34)$$

is found to be a reasonable choice. In some cases, larger perturbation may be required, for example when the particular parameter appears as a discrete nonlinear term, or when the effect (changes) in the system response due to that parameter is small. If a particular parameter happens to be zero, either as initially specified or during iteration, then deviating from Eq. (4.34), it is necessary to specify a small perturbation; otherwise Eq. (4.34) would yield zero perturbation and the computed gradient would be zero. A small absolute value, say $\delta\Theta_j = 10^{-6}$, is usually adequate to overcome this numerical peculiarity.

The procedure elaborated hitherto is a classical approach in which the parameters are varied one at a time.^{18,20,21,29} Thus, for n_q unknown parameters, at each time point we have to solve n_q times the perturbed state equations. More recently, in connection with the stochastic optimization a procedure called

“simultaneous perturbations” has been investigated,³¹ requiring just two evaluations at each discrete time point. For large n_q , such a method will be much faster. We have not investigated this concept of simultaneous perturbation for dynamic systems and in the framework of Gauss–Newton method, and hence do not go into any further discussion of it.

IX. Automatic Gradient Computation

As elaborated in Sec. VIII, gradients are usually approximated using either one-sided (forward or backward) or two-sided central differencing. The latter is more accurate, but requires twice the computational time. Basically, in both cases proper choice of step size is critical to obtain valid and accurate approximations. If the step size is too small, the round-off errors (resulting from subtraction of two almost equal numbers) may dominate; on the other hand too large a step may lead to non-negligible truncation errors (resulting from neglected higher-order terms of Taylor series), yielding an erroneous gradient. Some trade-off is usually necessary between these errors. We resorted to the finite-difference approximation for various reasons, for example, 1) analytical differentiation of the postulated model may be very tedious for complex models, 2) flexibility to consider different nonlinear model structures without programming changes each time, and 3) in some cases a program may be available just for computing the desired function.

However, if we can find a technique to do all these jobs automatically and also generate program code equivalent to an analytic derivative without actually going through the steps of deriving the mathematical expressions, we will not only achieve the desired goal, but also improve the accuracy by eliminating the errors introduced by finite differencing. This becomes possible through an upcoming technology in the field of mathematical computations, called automatic differentiation (AD).^{32,33} We cover here the fundamentals of this recent approach to indicate new possibilities.

The concept of automatic differentiation is based on the fact that any function, regardless of its programming language, is executed as a sequence of elementary operations such as additions, multiplications, and intrinsic functions. Such basic operations and functions are limited in number and their analytic derivatives are known. By repeated application of chain and product rules any function can be differentiated. To illustrate the procedure we quote here a simple example from Ref. 33 considering a function $y = (x + x^2)^2$. Figure 4.3a gives a possible way of programming the above function and Fig. 4.3b that for the automatically generated gradient, where the value of dx is arbitrary but not zero.

In simple terms, given a program code to evaluate a function in terms of the dependent and independent variables, the AD processor first synthesizes (breaks down) the code in terms of elementary operations. Based on this code synthesis, a program code is constructed to compute the derivative of the dependent variables with respect to the independent variables. Since the derivatives are computed now using elementary and intrinsic functions, the approximation errors vanish, giving derivatives to machine precision. The approach being analytical, it does not involve perturbations, and hence round-off or truncation errors are eliminated.

a)	Function $y = f(x)$ $z = x^*x$ $w = x + z$ $y = w^*w$ end	b)	Function $(y, dy) = fdot(x, dx)$ $z = x^*x$ $dz = 2*x^*dx$ $w = x + z$ $dw = dx + dz$ $y = w^*w$ $dy = 2*w^*dw$ end
----	---	----	--

Fig. 4.3 Sample example of automatic differentiation by chain rule. a) Function evaluation, b) function and gradient evaluation.

Thus, automatic differentiation provides an option to overcome numerical difficulties that may arise from finite difference approximations. Automatic differentiation will perform better, particularly in those optimization cases where the algorithm is sensitive to gradients. Applications of this new approach have been reported in a few fields based on static models (i.e., systems not incorporating state equations). Preliminary investigations suggest that the approach can be applied to dynamic systems with state and observation equations. Utility in terms of estimating aerodynamic derivatives and catering to special cases where finite-differences posed severe problems will have to be established through applying the new approach to a large number of cases. The only minor issue appears to be using a generated code that looks like a black-box, and thereby tends to lose the engineer's insight into a program code he may use.

X. Step Size Control

The unconstrained Gauss–Newton method, described in Sec. VI, takes a full step of parameter update at each iteration. Generally, the Gauss–Newton method performs very well. However, as presented in Secs. VI and VII, the update is based on the assumption of local linearity and a quadratic cost function. Furthermore, an approximation to the second gradient is incorporated to reduce the numerical burden, and the response gradients are approximated through numerical approximations. Owing to these limitations, the Gauss–Newton method may perform poorly, for example due to initial values far from optimum or numerical errors introduced through the finite difference approximation of gradients near the optimum, and it may show intermediate local divergence or stalling. In such cases it is necessary to control the step size carefully once the direction of the correction vector has been established. Three options are possible to overcome these typical numerical difficulties: 1) heuristic approach of parameter halving, 2) line search, and 3) dominant directions approach.

A. Heuristic Approach

In the first approach based on heuristic considerations, if the full step of the Gauss–Newton algorithm, Eq. (4.29), diverges at any iteration, the parameter update $\Delta\Theta$ is successively reduced by halving each time until reduction in the

cost function compared with the prior iteration is achieved. It amounts to a successive reduction of the full step through the factors $(2, 4, 8, \dots)$. During these halving steps just the cost-function evaluation is necessary, without computing the gradient. Halving of the parameter increments is generally performed a maximum of 10 times; which amounts to a factor of $(2^{10} = 1024)$ and thereby at the end of 10 halving steps we practically reach the parameters of the previous iteration. This implies that, under given conditions, the current iteration does not lead to any further improvement in the optimization. In that case, the optimization procedure is terminated. Usually a few halving steps are sufficient to overcome the intermediate divergence. This heuristic procedure is quite simple and in many cases found to be adequate.

B. Line Search

A more systematic approach and one that guarantees efficient performance is based on introducing a line search into the Gauss–Newton method. The update equation, Eq. (4.29), in such a case becomes

$$\Theta_{i+1} = \Theta_i + \alpha_i \Delta\Theta \quad \text{with} \quad \Delta\Theta = -\mathcal{F}^{-1}\mathcal{G} \quad (4.35)$$

where α_i is determined by a line search algorithm in each iteration. Line searches, also called one-dimensional searches, are the basic optimization procedures of finding a minimum of a cost function along the specified search direction. In the present case it amounts to reducing or increasing $\Delta\Theta$ of the basic Gauss–Newton method optimally to obtain maximum reduction in the cost function at each iteration.

Since just a few function evaluations are necessary for line search, the additional computational overhead is marginal. Such damping strategies help to widen the convergence region and to overcome other numerical problems.^{18,34,35} Any standard line search algorithm may be applied here; a quadratic line search procedure has shown good performance in many practical problems. The line search algorithm is also necessary for the bounded-variable Gauss–Newton method, which will be discussed in the next section.

C. Dominant Directions

The third approach to overcome intermediate divergence or stalling is based on evaluating the eigenvalues of the information matrix \mathcal{F} and determining the dominant search directions by checking the smallest and the largest eigenvalues. To understand the basic philosophy behind this approach, we begin by analyzing the various contributions to the parameter step $\Delta\Theta$. Recall that the parameter step is given by:

$$\Delta\Theta = -\mathcal{F}^{-1}\mathcal{G} \quad (4.36)$$

where \mathcal{F} is the information (Hessian) matrix and \mathcal{G} is the gradient vector. If \mathcal{F} is ill-conditioned, the Gauss–Newton method may result in an extremely large step and divergence. We now attempt to detect the ill-conditioning through singular

value decomposition (SVD). We know that \mathcal{F} is an $(n_q \times n_q)$ real, symmetric matrix. In this case we can write \mathcal{F} as follows:

$$\mathcal{F} = T \Lambda T^{-1} \quad (4.37)$$

where Λ is the diagonal matrix of eigenvalues of \mathcal{F} and T is the matrix of eigenvectors. By inverting the above equation, and using the matrix algebra results $(AB)^{-1} = B^{-1}A^{-1}$ and $T^{-1} = T^T$ for orthogonal matrices, we obtain

$$\mathcal{F}^{-1} = (T \Lambda T^{-1})^{-1} = (T^{-1})^{-1} \Lambda^{-1} T^{-1} = T \Lambda^{-1} T^T \quad (4.38)$$

Thus the parameter improvement $\Delta\Theta$ is given by

$$\Delta\Theta = -T \Lambda^{-1} T^T \mathcal{G} \quad (4.39)$$

Equation (4.38) can be rewritten in SVD form as

$$\mathcal{F}^{-1} = \frac{1}{\lambda_1} t_1 t_1^T + \frac{1}{\lambda_2} t_2 t_2^T + \cdots + \frac{1}{\lambda_q} t_q t_q^T \quad (4.40)$$

where the matrix of eigenvectors is partitioned into n_q column vectors as $T = [t_1, t_2, \dots, t_q]$ corresponding to the $(\lambda_1, \lambda_2, \dots, \lambda_q)$ eigenvalues.

Substituting Eq. (4.40) into Eq. (4.36) leads to

$$\Delta\Theta = -\frac{t_1^T \mathcal{G}}{\lambda_1} t_1 - \frac{t_2^T \mathcal{G}}{\lambda_2} t_2 - \cdots - \frac{t_q^T \mathcal{G}}{\lambda_q} t_q \quad (4.41)$$

which is a well-known result in the optimization theory, saying that the parameter step $\Delta\Theta$ is a sum of n_q steps of magnitude $[t_i^T \mathcal{G}/\lambda_i]$ in the directions of the eigenvectors $t_i, i = 1, 2, \dots, n_q$, of the information matrix.

For an ill-conditioned information matrix \mathcal{F} , at least one of its eigenvalues is very small. Let us assume that there are r such eigenvalues that may be considered very small. Since the step sizes in Eq. (4.41) are inversely proportional to the value of the eigenvalues, it may lead to very large step sizes along these r eigenvectors, and in the worst cases to divergence. The intuitive idea is to neglect these so-called bad directions and consider only the remaining $(n_q - r)$ dominant search directions. Mathematically, Λ and T are partitioned as

$$\Lambda = \begin{bmatrix} \Lambda_d & | & 0 \\ - & - & - \\ 0 & | & \Lambda_r \end{bmatrix} \quad \text{and} \quad T = [T_d \ T_r] \quad (4.42)$$

such that the dominant eigenvalues, that is, those greater than ε (a small positive number), are grouped together as Λ_d and the corresponding eigenvectors in the left partition T_d . Substituting Eq. (4.42) in Eq. (4.39), and simple manipulation

leads to

$$\Delta\Theta = -T_d\Lambda_d^{-1}T_d^T\mathcal{G} - T_r\Lambda_r^{-1}T_r^T\mathcal{G} \quad (4.43)$$

and neglecting the r small eigenvalues gives the step in the dominant search direction as:

$$\Delta\Theta \approx -T_d\Lambda_d^{-1}T_d^T\mathcal{G} \quad (4.44)$$

Let us discuss some practical issues and the performance of the above modified form of the Gauss–Newton method. It requires ε to be specified appropriately. This may depend upon the model size, modeling errors, and the noise in the data analyzed. In several practical cases of flight vehicle parameter estimation, it turned out that the above modification worked only in a third of the cases of an ill-conditioned matrix. This is not totally surprising, because neglecting large $[t_i^T\mathcal{G}/\lambda_i]$ steps does not necessarily mean that new directions are used in the optimization. All the remaining so-called dominant directions, in many cases a major number of them, are still the Gauss–Newton directions. A better way could be to generate a new set of search directions, as suggested in Ref. 35, possibly interpolating between Gauss–Newton and the Newton direction. The approach is again based on SVD and dominant directions. This is left to the reader to pursue, but with a remark that the best strategy would be one that is simple and straightforward, like the heuristic approach or line search presented earlier.

XI. Bounded-variable Gauss–Newton Method

The Gauss–Newton method discussed in Sec. VI provides an unconstrained solution to a nonlinear optimization problem. Parameter estimation subject to simple bounds may, however, be relevant in some cases. Typical applications related to flight-vehicle system identification are the following:³⁶

- 1) Parameters that describe the physical effects, in the present case aerodynamic effects, are often constrained to lie in a certain range, for example the Oswald's factor³⁷ characterizing the increase in drag over ideal condition caused by nonelliptical lift distribution and interference is typically limited to less than 1.
- 2) Time delays in the measurement variables are by definition always positive and hence greater than zero.
- 3) Estimation of highly nonlinear model parameters such as friction may lead to numerical difficulties due to reasons like poor guesses of initial values.³⁸
- 4) An attempt to estimate parameters with almost zero sensitivity resulting either from the use of inappropriate maneuvers or observation variables in the estimation procedure may lead to divergence.

In this section we address the issues pertaining to extending the Gauss–Newton method to account for simple bounds.

The linearly constrained optimization problem in which the constraints are simple bounds on the variables is formulated as:

$$\min_{\Theta} J(\Theta) \quad \text{subject to} \quad \Theta_{\min} \leq \Theta \leq \Theta_{\max} \quad (4.45)$$

where Θ_{\min} and Θ_{\max} are the lower and upper bounds on the parameter respectively.

Optimization theory provides several approaches to solve this problem, such as 1) transformation techniques, 2) penalty function methods, 3) barrier function or Lagrangian approach, and 4) active set strategy. The transformation technique is based on transforming the independent variables such that the constraints are satisfied automatically. Once this is done, any standard unconstrained optimization method can be applied.³⁹ Although the concept is simple to understand, it is not always possible to transform all constraints, and the form of the transformation equation may not be simple to find. The penalty function and Lagrangian approaches are more general and suitable for complex nonlinear inequality constraints. They are also more complex. In the present case we are considering a simpler problem of linear constraints. Several algorithms and software programs, for example, limited memory BFGS,⁴⁰ extrem,⁴¹ quasi-Newton,⁴² and bounded-variable least squares,⁴³ provide solutions to this problem. However, as already pointed out, the Gauss–Newton method is preferred here. The active set strategy is conceptually very appealing and can be readily extended to the Gauss–Newton method.³⁶

Starting from the initially specified parameter values Θ_0 , an active set IA containing the indices of the variables hitting the bounds is formed and updated every iteration. A variable is called a free variable if it is within the permissible bounds, and hence not in the active set. The Gauss–Newton search directions for the free variables are computed as follows:

$$\Delta\Theta_{\text{free}} = -\mathcal{F}_{\text{free}}^{-1} \mathcal{G}_{\text{free}} \quad (4.46)$$

where the information matrix $\mathcal{F}_{\text{free}}$ and the gradient vector $\mathcal{G}_{\text{free}}$ are computed using Eq. (4.30) for the free variables. The parameter updates resulting from Eq. (4.46) are checked for the specified bounds, and any violation of the constraints leads to modification of the active set IA. For such parameters the values are set to the respective bounds and the search directions of Eq. (4.46) to zero. For the remaining free parameters a new set of values is computed using a line search, which is necessary because changes in the number of free parameters change the search directions. As already addressed briefly in Sec. X.B, we have used the quadratic line search procedure in our applications. It is interesting to point out that, since the bounded-variable Gauss–Newton method includes the line search as an integral part of the algorithm, it automatically eliminates the problems of poor performance due to intermediate divergence.

An important aspect of the active set strategy is to develop a procedure of finding variables leaving or entering the feasible region, and accordingly to alter the active set IA in each iteration as the optimization progresses.

The active set is changed whenever a free variable hits its bounds during iteration. Furthermore, if the Kuhn–Tucker optimality conditions³⁹

$$\begin{aligned} \mathcal{G}_i < 0, \quad &\text{for } \Theta_i = \Theta_{i,\max} \quad \text{or} \\ \mathcal{G}_i > 0, \quad &\text{for } \Theta_i = \Theta_{i,\min} \end{aligned} \quad (4.47)$$

are not satisfied for any of the variables in the active set, then those variables are dropped from the active set and made free; \mathcal{G}_i , Θ_i , $\Theta_{i,\min}$ and $\Theta_{i,\max}$ are respectively the components of the gradient vector given by Eq. (4.30), the current parameter value, and its lower and upper bounds. In other words, conditions checked in Eq. (4.47) guarantee that the gradients for the variables hitting the bounds are such that they point outwards from the feasible region, implying that any further minimization of the cost function would be possible only when the particular parameters were not constrained within the specified limits.

The computational overhead to implement the active set strategy in an existing unconstrained Gauss–Newton method is minor; it is just necessary to check for the variables that hit the bounds and for the optimality conditions of Eq. (4.47) to enter or drop parameters from the active set. The advantages of this particular approach are twofold: 1) it provides an efficient solution retaining the desirable properties of the Gauss–Newton method, namely the quadratic convergence property, and 2) statistical information regarding the accuracy of the estimates, which we will address in Sec. XVII, is readily available.

XII. Constrained Gauss–Newton Method Using the Interior-point Algorithm

Yet another approach to constrained optimization that has attracted quite some attention in the area of nonlinear programming is based on the interior-point algorithm.^{44–46} Without going into the historical background of the development of such techniques, we straightaway address the applicability of such an algorithm within the framework of the maximum likelihood estimation that we have so far considered. To restate the problem, the maximum likelihood estimates of the unknown parameters and of the unknown noise covariance matrix are obtained by minimizing the cost function:

$$J(\Theta, R) = \frac{1}{2} \sum_{k=1}^N [\bar{z}(t_k) - y(t_k)]^T R^{-1} [\bar{z}(t_k) - y(t_k)] + \frac{N}{2} \ln |R| \quad (4.48)$$

where Θ is the n_q -dimensional vector of unknown parameters. Minimization of Eq. (4.48) yields the unconstrained estimates of Θ . In the present case, the parameters to be estimated are subjected to inequality constraints:

$$\Theta_{\min} \leq c(\Theta) \leq \Theta_{\max} \quad (4.49)$$

where $c(\Theta)$ are the inequality constraints. Through simple manipulation, the two-sided constraints of Eq. (4.49) can be transformed into the form

$$\begin{aligned} c(\Theta) - \Theta_{\max} &\leq 0 \\ \Theta_{\min} - c(\Theta) &\leq 0 \end{aligned} \quad (4.50)$$

For simplicity of notation and for derivation of the algorithm, we express the constraints in Eq. (4.50) in the general form given by

$$c_i(\Theta) \leq 0, \quad i = 1, \dots, p \quad (4.51)$$

where p is the total number of constraints and c can be of any form, but is assumed to be twice continuously differentiable. Thus, it is necessary to optimize the cost function of Eq. (4.48) subject to the constraints specified in Eq. (4.51). To apply the interior-point approach, we first reformulate the problem using the slack variables s_i to each of the constraints of Eq. (4.51).

$$\underset{\Theta, R}{\text{minimize}} \quad J(\Theta, R) \quad \text{subject to} \quad c(\Theta) + s = 0, \quad \text{and} \quad s \geq 0 \quad (4.52)$$

where c and s are p -dimensional vectors. The inequality constraints in Eq. (4.52) are eliminated by introducing a barrier parameter $\mu (> 0)$, which results in the following problem:

$$\underset{\Theta, R}{\text{minimize}} \quad \left\{ J(\Theta, R) - \mu \sum_{i=1}^p \log s_i \right\} \quad \text{subject to} \quad c(\Theta) + s = 0 \quad (4.53)$$

where $J(\Theta, R)$ is given by Eq. (4.48). Adopting the Lagrangian approach, we can rewrite Eq. (4.53) as:

$$\begin{aligned} \tilde{J}(\Theta, R, s, \lambda, \mu) &= \frac{1}{2} \sum_{k=1}^N [z(t_k) - y(t_k)]^T R^{-1} [z(t_k) - y(t_k)] \\ &\quad + \frac{N}{2} \ln |R| - \mu \sum_{i=1}^p \log s_i + \lambda^T [c(\Theta) + s] \end{aligned} \quad (4.54)$$

where λ is the vector of Lagrange multipliers.

The first-order condition for a minimum is that the first derivatives of \tilde{J} with respect to the respective variables, namely Θ, R, s , and λ , vanish. As in the case of the classical unconstrained output error method described in Sec. V, we apply the two-step relaxation strategy, estimating R first and then the other parameters. Following the development presented in Sec. V.B and Appendix E, it can be

shown that the estimate of the covariance matrix of residuals R is given by

$$R = \frac{1}{N} \sum_{k=1}^N [z(t_k) - y(t_k)][z(t_k) - y(t_k)]^T \quad (4.55)$$

which is the same expression as given by Eq. (4.15).

For a known R given by Eq. (4.55) during the particular iteration, equating $\partial\tilde{J}/\partial\Theta$, the partial differentiation of Eq. (4.54) with respect to Θ , to zero yields

$$-\sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} [z(t_k) - y(t_k)] + \left(\frac{\partial c}{\partial \Theta} \right)^T \lambda = 0 \quad (4.56)$$

Similarly, partial differentiation of Eq. (4.54) with respect to s and λ and equating to zero leads respectively to

$$-\mu e + S\lambda = 0 \quad (4.57)$$

and

$$c(\Theta) + s = 0 \quad (4.58)$$

where S is a diagonal matrix with elements s_i and e is a vector of ones. It can be shown that Eqs. (4.56)–(4.58) represent the so-called Karush–Kuhn–Tucker conditions for the constrained minimum.

As in the case of the standard unconstrained Gauss–Newton method, we start from a valid (interior) point Θ_0 , s_0 , λ_0 . Without going into the detailed mathematical derivation of individual steps, we state here the final expression for $\Delta\Theta$, Δs , $\Delta\lambda$ updates. For a pre-specified μ , the Gauss–Newton step is computed according to

$$\begin{aligned} & \left\{ \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \left[\frac{\partial y(t_k)}{\partial \Theta} \right] + \left(\frac{\partial c}{\partial \Theta} \right)^T S^{-1} \Lambda \left(\frac{\partial c}{\partial \Theta} \right) \right\} \Delta \Theta \\ &= \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} [z(t_k) - y(t_k)] \\ &\quad - \left(\frac{\partial c}{\partial \Theta} \right)^T S^{-1} \Lambda [s + c(\Theta) + \mu \Lambda^{-1} e] \end{aligned} \quad (4.59)$$

$$\Delta s = - \left(\frac{\partial c}{\partial \Theta} \right) \Delta \Theta - c(\Theta) - s \quad (4.60)$$

$$\Delta \lambda = S^{-1} [\mu e - S\lambda - \Lambda \Delta s] \quad (4.61)$$

where Λ is a diagonal matrix with elements λ_i .

The matrix on the left-hand side of Eq. (4.59) is symmetric and positive definite. Any standard numerical procedure like Cholesky factorization can be used to solve Eq. (4.59) for the parameter update $\Delta\Theta$. The first term in the flower bracket on the left-hand side of Eq. (4.59) is the same as that appearing in Eq. (4.28) or (4.30) applying the unconstrained Gauss–Newton method. The full Gauss–Newton step, resulting from Eqs. (4.59)–(4.61), does not guarantee that the updated parameters satisfy all the constraints. Therefore, a suitable damping strategy needs to be applied. The common approach is

$$\Theta_{i+1} = \Theta_i + \alpha_P \Delta\Theta_i \quad (4.62)$$

$$s_{i+1} = s_i + \alpha_P \Delta s_i \quad (4.63)$$

$$\lambda_{i+1} = \lambda_i + \alpha_D \Delta\lambda_i \quad (4.64)$$

where i is the iteration index and (α_P , α_D) are the damping coefficients. These coefficients guarantee that the non-negative variables remain non-negative and prevent the estimates from approaching the boundaries too closely.^{44,45} They are chosen based on the standard ratio test:

$$\alpha_P = \min \left\{ 0.9995 \min_{(j: \Delta s_j < 0)} \{-s_j / \Delta s_j\}, 1 \right\} \quad (4.65)$$

$$\alpha_D = \min \left\{ 0.9995 \min_{(j: \Delta \lambda_j < 0)} \{-\Lambda_j / \Delta \lambda_j\}, 1 \right\} \quad (4.66)$$

The last aspect of the interior point algorithm pertains to the choice of the barrier parameter μ . In general, the choice is model and data-dependent. However, the following procedure was also found to be adequate for a number of examples of flight data analysis:^{45,46}

$$\mu = \frac{\lambda^T s}{(n_q + p)^2} \quad (4.67)$$

It is fairly obvious that the interior point algorithm with μ equal to zero reduces to the unconstrained method. For $\mu > 0$, the algorithm ensures that the estimated parameters remain within the specified constraints.

For linear constraints specified as simple upper and lower bounds, the two approaches to account for the limits, namely the bounded-variable Gauss–Newton method described in Sec. XI and the interior-point algorithm described in this section, were found to be comparable in terms of convergence, numerical values and accuracies. The more sophisticated formulation based on the interior point will be efficient when more complex constraints have to be satisfied.

XIII. Levenberg–Marquardt Method

The Levenberg–Marquardt method is yet another approach to overcoming the numerical problems discussed in the previous sections.^{47,48} It combines the best features of the standard unconstrained Gauss–Newton method and the

steepest-descent technique that moves in the gradient direction. Owing to optimal combination of two search directions, the Levenberg–Marquardt method has a wider convergence region. The update formula in such a case is given by

$$\Theta_{i+1} = \Theta_i + \Delta\Theta \quad \text{with} \quad (\mathcal{F} + \lambda I)\Delta\Theta = -\mathcal{G} \quad (4.68)$$

where λ is the Levenberg–Marquardt (LM) parameter that can be smoothly varied to control whether the update search direction is more like steepest-descent or along the Gauss–Newton direction. It is easily seen from Eq. (4.68) that, for $\lambda \rightarrow \infty$, the method becomes steepest-descent and, as $\lambda \rightarrow 0$, it becomes like Gauss–Newton.

The computation of \mathcal{F} and \mathcal{G} is exactly the same as in the previous case of the Gauss–Newton method, given by Eq. (4.30). It now remains to find a proper value for λ that performs optimum interpolation between the two methods. Many methods exist for computing the LM parameter λ . The control strategy described in Ref. 47 is simple and works well in practical cases:

- 1) Choose a reduction factor $\nu > 1$, typically $\nu = 10$; choose $\lambda_0 = 0.001$ as an initial starting value.
- 2) Perform one step of the Gauss–Newton method, and increment the iteration counter i by one; let λ^{i-1} denote the value from the previous iteration.
- 3) Solve the system of linear equations, Eq. (4.68), for two values of the LM parameter, λ^{i-1} and (λ^{i-1}/ν) .
- 4) Now, for the two solutions of $\Delta\Theta$ from step 3, perform the parameter update and compute the respective cost functions $L_i = L[\Theta(\lambda^{i-1})]$ and $L_i^\nu = L[\Theta(\lambda^{i-1}/\nu)]$.
- 5) Comparison of L_i and L_i^ν leads to the following three possible cases:
 - a) If $L_i^{(v)} \leq L_{i-1}$, then set $\lambda^{(i)} = \lambda^{(i-1)}/\nu$ and $\Delta\Theta = \Delta\Theta(\lambda^{i-1}/\nu)$ (i.e. smaller λ results in reduced cost function, hence accept the new point and reduce the LM parameter for the next iteration).
 - b) If $L_i^{(v)} > L_{i-1}$ and $L_i \leq L_{i-1}$, then set $\lambda^{(i)} = \lambda^{(i-1)}$ and $\Delta\Theta = \Delta\Theta(\lambda^{i-1})$ (i.e. accept the new point, but retain the LM parameter from the previous iteration).
 - c) Otherwise, $L_i^{(v)} > L_{i-1}$ and $L_i > L_{i-1}$, then set $\lambda^{(i-1)} = \lambda^{(i-1)}\nu$ (i.e., increase λ) and repeat steps 2–5, until an acceptable new point is found.

In simple words, the control-strategy elaborated above evaluates the cost function values (among themselves and compared with the previous lowest cost) for two LM parameters. The one corresponding to the greatest reduction is accepted and the program proceeds to the next iteration. Where both the new updates have costs which are higher than the cost in the previous iteration, the factor λ is increased and the cycle repeated. Adaptation of the LM parameter generally guarantees convergence to the optimum.

The choice of the reduction factor ν is arbitrary, and $\nu = 10$ is a good choice for many test cases pertaining to aircraft parameter estimation. Since

the properties of the gradient optimization methods are not scale-invariant, the starting value for the LM parameter λ depends on the order of magnitude of the system (information) matrix. A scaling, widely used in linear least squares problems, is given by

$$\mathcal{F}^* = (f_{ij}^*) = \left(\frac{f_{ij}}{\sqrt{f_{ii}}\sqrt{f_{jj}}} \right) \quad (4.69)$$

$$\mathcal{G}^* = (g_i^*) = \left(\frac{g_i}{\sqrt{f_{ii}}} \right) \quad (4.70)$$

This leads to a scaled system of linear equations:

$$\mathcal{F}^* \Delta \Theta^* = -\mathcal{G}^* \quad (4.71)$$

which can be solved by any standard procedures of linear algebra. The solution for the transformed $\Delta \Theta^*$ is then scaled back as

$$\Delta \Theta_i = \frac{\Delta \Theta_i^*}{\sqrt{f_{ii}}} \quad (4.72)$$

The transformed matrix \mathcal{F}^* is a simple correlation matrix, whose diagonal elements are one. For such a scaling, a starting value of $\lambda_0 = 0.001$ is recommended, which proved adequate for several test cases investigated.^{30,49}

The Levenberg–Marquardt method requires Eq. (4.68) to be solved twice per iteration compared with once in the Gauss–Newton method. However, this overhead in just a minor part of the total overhead, as elaborated in Sec. XVIII, a large portion being spent on computing the information matrix, that is, the second gradient of the cost function, which is done only once in both the cases. The adaptation of the LM parameter ensures that the convergence is monotonic and guarantees convergence from iteration to iteration, and as such the procedures described in Sec. X to overcome intermediate divergence are not relevant here. The Levenberg–Marquardt method works well for starting points far from the optimum. It is also useful to overcome the numerical problems owing to one-sided approximation of gradients near the optimum.

XIV. Direct Search Methods

Optimization methods that require just function evaluations and not gradients belong to the class of direct search methods. They are, in general, much slower compared with gradient-based methods,^{29,50,51} but are somewhat more robust, particularly starting from a point which is far from the optimum. Moreover, standard program codes are readily available. Since the literature on optimization theory is abundant with the details of these methods, only a very brief verbal description is provided here.

A. Simplex and Subplex Methods

The downhill simplex method developed by Nelder and Mead and the subspace searching method developed by Rowan are based on the principles of reflection, expansion, contraction, and shrinkage (massive contraction) of an initially specified q -dimensional convex hull, a geometrical figure with $n_q + 1$ vertices called simplex.^{52,53}

In the Nelder and Mead simplex method, a simplex moves through the q -dimensional space of the cost function, changing shape and size, and automatically shrinks when the minimum is enveloped by the simplex.⁵² The simplex method is widely used in the optimization of cost functions in several applications, but rarely for flight vehicle parameter estimation from flight data. Nevertheless, owing to its robustness with respect to the initial guess of the unknown parameters, it could be useful in a few cases, for example estimation of parameters with discrete nonlinearities, friction parameters, and so on.

The subplex method is a generalization of the simplex method of Nelder and Mead. It decomposes the higher-dimensional problem into smaller-dimensional subspaces in which the simplex method can search efficiently.⁵³ The task is twofold; first, we determine the improved set of subspaces and second we apply the simplex method. The concept of orthogonality used for the subspaces is similar to the concept of primary and secondary search directions used in Jacob's heuristic minimum search method called "Extrem," see Sec. XIV.C.

When the number of parameters to be estimated is large, the classical simplex method of Nelder and Mead rapidly becomes inefficient. A restart procedure helps in some cases to improve the performance somewhat. On the other hand, the computational overhead for the subspace searching method increases roughly linearly with the problem dimension. From this viewpoint, it is considered the most promising of the simplex-based methods.

B. Powell's Method

Powell's method is a widely used direct search method.⁵⁴ It can be shown to be a method of conjugate directions that minimizes the cost function in a finite number of cost function evaluations. The method is an iterative procedure and involves computing the search directions at each iteration and finding the minimum along these search directions by applying a line search method.

C. Jacob's Method

Jacob's method is based on the choice of optimal search directions, determination of the optimum along a line and defining the step size for each search.⁴¹ The Gram–Schmidt orthogonalization procedure is used to select the orthogonal search directions, in contrast to the conjugate directions used in Powell's method. The search directions are adapted to the local properties of the parameter space being searched. Basically, the procedure is very similar to the Gauss–Newton method. A program called "Extrem," which is available in the literature, caters for constrained optimization. For example, simple bounds on the parameters being estimated can be accounted for.

Having covered in Secs. IV–XIV several methods to optimize the cost function, the choice of a particular method may appear to be a difficult one. However, based on comparative studies performed in the past,^{29,50,51} it can be concluded that the direct search methods covered in Sec. XIV are extremely slow compared with the gradient-based Gauss–Newton, its variation accounting for bounds or the Levenberg–Marquardt method. The last mentioned methods are far superior in performance, mainly because they make use of the first and second gradients to predict the minimum and directly step in the best local direction. Any one of these will perform equally well on the chosen problem. We have widely used the Gauss–Newton and bounded-variable Gauss–Newton method, incorporating either line search or the heuristic approach of parameter halving to overcome any intermediate divergence or stalling of the optimization procedure. Nevertheless, in certain rare cases, like estimation of friction parameters, the simplex or subplex may give better starting values for the parameters. It is also possible to switch from one method to another as the optimization progresses.

XV. Regression Startup Procedure

In Sec. X it has been pointed out that the Gauss–Newton method may perform poorly for initial values far away from optimum, and we have discussed some techniques to overcome this problem. Yet another way would be to generate better starting values using some other technique which does not need initial guess values. This is possible using a procedure belonging to the broad class of least squares methods which we call regression startup method. It represents a mixed least squares and output error approach. To illustrate the method, first consider the linearized system representation:

$$\dot{x}(t) = A(\beta)x(t) + B(\beta)u(t) \quad (4.73)$$

$$y(t) = C(\beta)x(t) + D(\beta)u(t) \quad (4.74)$$

Now, assuming that measurements of all the state variables are available, and denoting them x_m , the system equations are reformulated as:

$$\dot{x}(t) = A_Fx(t) + [B(\beta); \quad A_E(\beta)] \begin{bmatrix} u(t) \\ x_m(t) \end{bmatrix} \quad (4.75)$$

$$y(t) = C_Fx(t) + [D(\beta); \quad C_E(\beta)] \begin{bmatrix} u(t) \\ x_m(t) \end{bmatrix} \quad (4.76)$$

The matrices A and C are each resolved into two, namely (A_F, A_E) and (C_F, C_E). The matrices with the subscript “F” contain the possibly fixed coefficients, if any, and those with the index “E” contain the parameters to be estimated.

In general, the fixed parameters appearing in matrices A_F and C_F correspond to those resulting from kinematic terms or derivatives fixed using the a priori values, whereas those appearing in A_E and C_E correspond to stability and

control derivatives which are unknown. This reformulation makes the state variables, x , and thereby the observation variables, y , and hence the error $z(t_k) - y(t_k)$ linear in the parameters being estimated. For such cases, the cost function being quadratic, the least squares method will yield estimates in a single iteration. A detailed treatment of the least squares technique will be dealt with in Chapter 6. The above formulation has the advantage that the initial values of the unknown parameters are not required. The optimization procedure can be started with zero values of all the parameters. Note that the above formulation retains state equations which have to be integrated, but it incorporates measured states as control inputs in conjunction with unknown parameters, thus leading to a combined output error and least squares approach. The optimization of the cost function is carried out using the same iterative algorithm that we have studied earlier (Sec. VII).

Applying regression analysis to the nonlinear model postulated in Eqs. (4.10)–(4.12) is not straightforward. If the model is linear in parameters the optimization algorithm is a one-shot procedure, whereas for a model nonlinear in parameters the iterative optimization is necessary. In such a case some reasonable start values may be required. Treatment of unknown initial conditions is a little trickier. However, based on practical considerations an acceptable solution becomes feasible. Assuming that the measurements of the state variables, x_m , are available, Eqs. (4.10)–(4.12) are reformulated as follows:

$$\dot{x}(t) = f[x(t), x_m(t), u(t) - \Delta u(b_{u,l}), (\beta_{xF}, \beta_{xE})] \quad x(t_{0,l}) = x_m(t_{0,l}) \quad (4.77)$$

$$y(t) = g[x(t), x_m(t), u(t) - \Delta u(b_{u,l}), (\beta_{xF}, \beta_{xE}), (\beta_{yC}, \beta_{yE})] + \Delta z(b_{y,l}) \quad (4.78)$$

where the unknown parameters β_x and β_y are subdivided into two components (β_{xF}, β_{xE}) and (β_{yF}, β_{yE}) corresponding to those parameters which are constants and those which are to be estimated. The parameters to be estimated are multiplied by the measured states, x_m , or by measured control inputs u , whereas the constant parameters are multiplied by the integrated states. The initial values $x(t_0)$ are set to the measured values at time t_0 and held fixed. Furthermore, the zero shifts in the control variables, Δu , which represent the systematic errors, are assumed to be small and are, hence, either set to zero or kept fixed at the value specified. Estimation of the zero shifts in the observation variables, Δz , poses no difficulties because the model is linear in them.

Thus, associating the unknown parameters with the measured states leads to a mixed output error and least squares approach, which can be used to generate good initial guess values for the unknown parameters, both for linear and nonlinear model postulates. As will be discussed in Chapter 9, Sec. V, such a mixed least squares and output error formulation is also useful in estimating parameters of unstable aircraft without facing the numerical problems otherwise encountered in applying the classical output error method. In a few cases the above general approach has been used without subdividing the matrices, and

using the measured states throughout. Such a procedure is much simpler to implement and found to work equally well.

XVI. Estimation Accounting for a Priori Information

In some cases a priori information about certain aerodynamic derivatives may be available from wind-tunnel tests or from estimation results of other flight test investigations. The Bayesian estimator enables a priori information to be accounted for in a probabilistic way by treating the parameter vector $\Theta(n_q \times 1)$ as a random vector.⁵⁵ In such a case, a priori knowledge about the probability density functions of the measurements and of the parameters is necessary. The Bayesian estimator represents the most general type of parameter estimation methodology, but it also needs the maximum amount of a priori information.

The conditional probability density function $p(\Theta|z)$, which represents the a posteriori probability density function of the parameters Θ , given the measurements z , can be expressed using Bayes' rule as⁵⁵

$$p(\Theta|z) = \frac{p(\Theta)p(z|\Theta)}{p(z)} \quad (4.79)$$

For convenience, both $p(\Theta)$ and $p(z|\Theta)$ are assumed to be Gaussian. In this case, they can be expressed as

$$p(z|\Theta) = \{(2\pi)^{n_y} |R|\}^{-N/2} \exp\left(-\frac{1}{2} \sum_{k=1}^N [z(t_k) - y(t_k)]^T R^{-1} [z(t_k) - y(t_k)]\right) \quad (4.80)$$

and

$$p(\Theta) = \{(2\pi)^{n_q} |R_2|\}^{-1/2} \exp\left(-\frac{1}{2} (\Theta - \Theta^*)^T R_2^{-1} (\Theta - \Theta^*)\right) \quad (4.81)$$

where Θ^* is the vector of a priori values of the derivatives, R_2 is the error covariance matrix of the a priori values and R is the measurement noise covariance matrix. It follows that the a priori values and the associated uncertainty are represented as:

$$E\{\Theta\} = \Theta^* \quad (4.82)$$

and

$$E\{(\Theta - \Theta^*)(\Theta - \Theta^*)^T\} = R_2 \quad (4.83)$$

In the estimation procedure, for a given set of measured data, $p(z)$ can be treated as a constant. Thus, maximization of $p(\Theta|z)$ in Eq. (4.79) is equivalent

to maximization of J_{ap} given by:

$$J_{\text{ap}} = (2\pi)^{-(n_q+n_yN)/2} |R|^{-N/2} |R_2|^{-1/2} \times \exp \left(-\frac{1}{2} \sum_{k=1}^N [z(t_k) - y(t_k)]^T R^{-1} [z(t_k) - y(t_k)] - \frac{1}{2} (\Theta - \Theta^*)^T R_2^{-1} (\Theta - \Theta^*) \right) \quad (4.84)$$

Setting the gradient with respect to Θ to zero and solving yields:

$$\begin{aligned} & \left\{ \sum_k \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \left[\frac{\partial y(t_k)}{\partial \Theta} \right] + R_2^{-1} \right\} \Delta \Theta \\ &= \sum_k \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} [z(t_k) - y(t_k)] - R_2^{-1} (\Theta - \Theta^*) \end{aligned} \quad (4.85)$$

Although the Bayesian estimation represents a more general approach, the a priori information about the probability density function $p(\Theta)$ is often difficult to obtain. Relaxing the assumption of parameters being random variables leads to the maximum likelihood estimation that we had already discussed in some depth in Secs. II–V. To recall once again, the maximum likelihood method assumes that Θ does not depend on chance and therefore requires a priori knowledge only about $p(z|\Theta)$. In such a case Eq. (4.85) simplifies to Eq. (4.28).

Since it is difficult in practice to specify accurately the error covariance matrix of the a priori values, a weighting factor W is additionally introduced in conjunction with R_2 . In such a case, it can be easily shown that Eq. (4.85) leads to

$$\begin{aligned} & \left\{ \sum_k \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \left[\frac{\partial y(t_k)}{\partial \Theta} \right] + WR_2^{-1} \right\} \Delta \Theta \\ &= \sum_k \left[\left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} [z(t_k) - y(t_k)] \right] - WR_2^{-1} (\Theta - \Theta^*) \end{aligned} \quad (4.86)$$

Introduction of the weighting factor W provides a convenient means to study the effect of a priori values on the estimates. It is obvious from Eq. (4.86) that $W = 0$ implies suppression of the a priori values and $W \rightarrow \infty$ implies suppression of the measurements. In the latter case, the algorithm will neglect the measurements completely, yielding the a priori values as the estimates. This is a hypothetical case, because if we are totally sure that the a priori values are perfect, then we do not have to worry at all about the estimation from experimental data. This is in contradiction to our hypothesis. We have gone through the whole exercise of statistically estimating the parameters, because we want to improve upon the accuracy of the estimated parameters and the model structure from the

experimental data. This approach could, however, help to avoid convergence problems encountered when the parameters are near linearly dependent.

In practice, if the a priori information is to be included in the estimation, it is usually done for system parameters (i.e., aerodynamic derivatives) only. When this option is invoked, it is necessary to specify the a priori values and corresponding error covariances along with the values for parameters appearing in the state equations. In practice, specifying an appropriate error covariance matrix R_2 for the a priori values and of the weighting factor W is difficult. It is this aspect that restricts the utility of this option. In general, this option is rarely used in flight vehicle system identification.

XVII. Statistical Accuracy of Parameter Estimates

Having applied the statistical estimation methods, the first and the most natural question that comes to our mind is about the statistical accuracy of the estimates that we have obtained from the experimental data. In other words, we would like to gain some appreciation of the confidence that we may have in the estimates obtained by applying the maximum likelihood or Bayesian estimator. In some sense the issue of accuracy of the estimates, belongs to the category of tests that are required to evaluate the model adequacy and validation. We will deal with this aspect separately in Chapter 11, dealing with various procedures and practical issues of model validation. However, for the sake of completeness we cover here only the mathematical basics of statistical evaluation of the estimates.

From Sec. III and Appendix D, we know that the maximum likelihood estimator is asymptotically efficient in the sense of achieving the Cramér–Rao lower bound. It follows from Eq. (D.35) that the parameter error covariance matrix P is given by the inverse of the Fisher information matrix \mathcal{F} [Eq. (4.30)]:

$$P = \left\{ \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \left[\frac{\partial y(t_k)}{\partial \Theta} \right] \right\}^{-1} \quad (4.87)$$

If a priori information, as discussed in Sec. XVI, is included in the estimation, it is required to use the system matrix, that is, terms within the brace on the left-hand side of Eq. (4.86) while computing the above parameter error covariance matrix. The standard deviations and correlation coefficients of the estimates are given by:

$$\sigma_{\Theta_i} = \sqrt{p_{ii}} \quad (4.88)$$

$$\rho_{\Theta_i \Theta_j} = \frac{p_{ij}}{\sqrt{p_{ii} p_{jj}}} \quad (4.89)$$

The property of asymptotic efficiency is of practical significance. It implies that the ML estimator makes efficient use of the available data and that the Cramér–Rao lower bound indicates the theoretically maximum achievable accuracy of the estimates. Although Eqs. (4.88) and (4.89) provide a simple way to

obtain theoretical information about the accuracy of the estimates and correlation among them, unfortunately, in practical applications this issue is more complex and not without ambiguities. We will discuss these implications separately in Chapter 11, dealing with model validation.

Since the information matrix \mathcal{F} is required in the optimization of the cost function, the parameter error covariance matrix is automatically available without any extra computations. The standard deviations and correlation coefficients can be readily obtained. Thus, information on the statistical accuracy of the estimates is a byproduct of the Gauss–Newton or Levenberg–Marquardt methods. Other optimization methods we have briefly mentioned, namely Powell's, Jacob's, simplex, and subplex, do not provide this information directly, and this is one of the limitations of these methods, besides being very slow in convergence.

XVIII. Algorithmic Implementations

The computational demands for large-scale systems and huge amounts of measured data required for global model identification are significant and can become prohibitive; the job can run to hours of CPU time.⁵⁶ Therefore, special emphasis is placed on computational details to achieve reasonable turn-around time in such cases. It makes use of the special structure of the information matrix resulting from multiple time segments (maneuvers) being processed.

As already elaborated in the previous sections, the iterative unconstrained or bounded-variable Gauss–Newton methods require a linear algebraic equation to be solved once and the Levenberg–Marquardt twice per iteration. Although, depending upon the number of unknown parameters being estimated, the dimension of the information matrix could be large, the main computational burden is not due to solving the algebraic equation required for matrix inversion; see Eq. (4.29). The major portion of the computational time is consumed in computing the $n_y \times n_q$ dimensional sensitivity coefficients $\partial y / \partial \Theta$ and the $n_q \times n_q$ dimensional information matrix \mathcal{F} over N data points; see Eq. (4.30).

As a typical case, we consider an example pertaining to the six DOF equations of motion, with number of states $n_x = 10$, number of inputs $n_u = 12$, number of outputs $n_y = 19$ and the number of parameters to be estimated $n_q = 30$. Throughout the following comparative investigations we use the fourth-order Runge–Kutta integration method. We perform five different evaluations using different numbers of maneuvers analyzed simultaneously each time, namely $n_E = 10, 20, 30, 40$, and 60 , with $N = 13,424, 25,453, 37,720, 51,144$, and $75,724$ data points, respectively. In each case the initial conditions are specified appropriately (as an average of the first few points of measured data, see Chapter 3, Sec. VII) and held fixed, and no other bias parameters are considered. Thus, we have in each case the same number of unknown parameters $n_q = 30$, and the information matrix is 30×30 . On a workstation with 3.4 GHz CPU, on average 9.1, 17.4, 25.7, 34.8 and 52 s per iteration are required to compute the information matrix \mathcal{F} and gradient vector \mathcal{G} (time linearly increasing with the number of data points), whereas the matrix inversion took negligible time (less than 0.01 s).

Now, let us consider estimating the initial conditions for each of the n_E time segments analyzed simultaneously. Then, the total number of parameters n_q is

given by $30 + n_x * n_E$, where in the present case the number of states is $n_x = 10$. Thus, in the five cases it amounts to estimating 130, 230, 330, 430, and 630 parameters, from which each time 30 are the aerodynamic derivatives. On the same 3.4 GHz workstation roughly 20, 78, 214, 485, and 1518 s were required per iteration to compute the information matrix \mathcal{F} and gradients \mathcal{G} . The information matrix \mathcal{F} being symmetrical, we compute only the lower triangular part, and then fill it up before inverting the same. Apart from this single consideration, no other special treatment is performed, implying that the build-up of \mathcal{F} and \mathcal{G} is done in a straightforward fashion using Eq. (4.30) for all n_q parameters. The matrix inversion requires from 0.01 to 3 s, once again a negligible part of the overall computations even for the last case with 630×630 size matrix.

Thus, having substantiated our statement made earlier that the major portion of the computational time is consumed in computing the $n_q \times n_q$ information matrix, we now look for the ways to optimize computations to reduce the overall CPU time. If only aerodynamic parameters are estimated, keeping initial conditions and other bias parameters fixed, then no optimization is possible, irrespective of how many n_E maneuvers are analyzed simultaneously. As demonstrated above, in such a case for a given model postulate the CPU time is linearly proportional to the number of data points. However, if initial conditions or bias parameters are also estimated, then it turns out that, through a judicious implementation of the computations in the \mathcal{F} matrix built up pertaining to n_E time segments, significant reduction in the computational burden can be achieved. This becomes possible because concatenation of several time-slices leads to a special block structure for the information matrix with respect to the initial conditions and other bias parameters. For illustration purposes, just three time segments and for each time segment the initial conditions and bias parameters of the observation equations are considered. The resulting matrix structure is shown schematically in Fig. 4.4. The nonzero elements appear as gray blocks with bold borders. The sparse matrix structure can be used advantageously to reduce the numerical computations while computing the $n_q \times n_q$ dimensional information matrix.⁵⁷ Furthermore, part of the computation is done judiciously just once, namely $[\partial y / \partial \Theta]^T R^{-1}$, which is common to both \mathcal{F} and \mathcal{G} .

The use of this optimization led to CPU time requirements of 12, 23, 35, 48, and 70 s per iteration for the five cases. Compared with the CPU times of 20, 78, 214, 485, and 1518 s using a straightforward approach the reduction is significant. The larger the number of time segments is, the larger the ratio of time saving. Since it is usually preferable to estimate the initial conditions and bias parameters, and the use of a large number of time segments is not uncommon, as well as 10 or more iterations typically being needed for the convergence, a significant amount of reduction in the computational overhead can be achieved by making optimum use of the special block structure for multiple maneuver evaluations. This issue is not so relevant for small problems, but it can become critical to ensure reasonable turnaround time during the model development process for large problems encountered during global model identification, as addressed in Chapter 12, Sec. VIII.

Yet another special implementation feature pertains to avoiding large matrices through judicious implementation of computational sequence. Several estimation programs are based on computing system responses, and those for the perturbed

		0	0		0	0			
	0		0	0		0			
	0	0		0	0				
		0	0		0	0			
	0		0	0		0			
	0	0		0	0				
Block for system parameters		1	2	3	1	2	3	Time segment	Time segment
Block for initial conditions			Block for bias parameters of observation equation						

Fig. 4.4 Schematic of block structure of the information matrix for three time slices analyzed simultaneously.

parameters required for approximating gradients, in a sequential way. In this direct approach, one parameter is perturbed at a time, and simulation performed at a stretch over the complete set of N data points. The process is repeated n_q times, storing the responses in separate data arrays each time. This so-called sequential processing, shown on the left-hand side of Fig. 4.5, needs $(n_q + 1)$

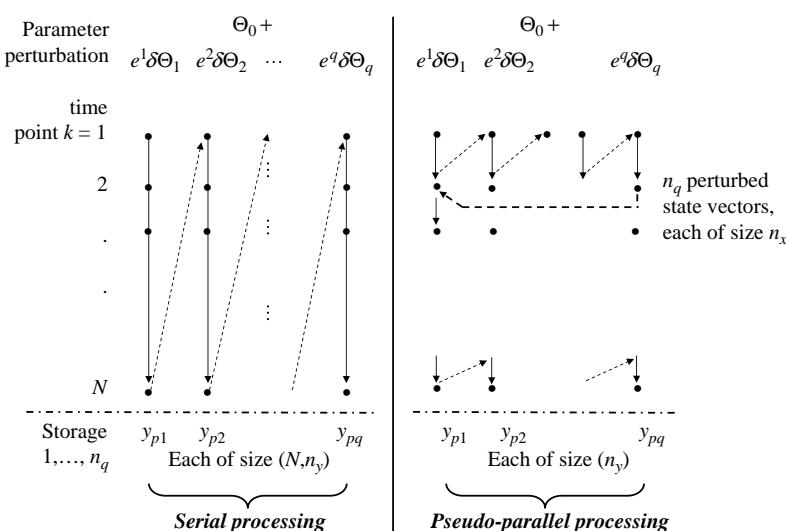


Fig. 4.5 Two different computational sequences for perturbed simulations.

arrays, each of size $(N \times n_y)$. At the end of such $(n_q + 1)$ simulations the gradients are computed from these responses using Eq. (4.33) and summed over N data point according to Eq. (4.30). Thus, depending upon the number of data points N , the program may need very large size arrays; program capable of analyzing up to 80,000 data points and 60 observations, 30 states, 60 inputs, 1200 parameters has been used in practice; in a rare case up to 150,000 data points or more have also been processed at a time.

For such large problems, the following alternative implementation, called pseudo-parallel processing, leads to substantial reduction in the storage needed. As depicted on the right-hand side of Fig. 4.5, instead of performing the simulations at a stretch over N data points for each perturbation, we handle each data point separately. At each data point, we perturb each of the n_q parameters, compute the responses, compute the gradients according to Eq. (4.30), and incrementally build up the matrices \mathcal{F} and \mathcal{G} . Having done this for all n_q parameters, then we jump to the next data point, restore the perturbed states for continuation and repeat the process to process the next point, until all the data points are processed. Restoration of appropriate states at each point needs saving perturbed states for each parameter, requiring array of the size $n_x \times n_q$. This pseudo parallel approach characterized by horizontal processing at each point needs just two arrays, one of size $(n_y \times n_q)$ and the other of $(n_x \times n_q)$. Compared with the direct implementation discussed earlier, requiring arrays of total size $(N \times n_y \times n_q)$, the storage requirements of the second approach are negligible. We have called this approach pseudo-parallel, because strictly speaking the terminology of parallel processing has, in general, a different connotation, implying several simulations running at the same time, whereas we still perform perturbations and simulations sequentially, but the order of performing the same over N data points and for n_q has been interchanged, leading to remarkable saving in the storage requirements.

XIX. OEM Software

There are so very many ways to implement the output error method that attempts to standardize the software program will be in vain, because the model postulates vary from case to case (remember we mostly deal with a general nonlinear system of arbitrary complexity). Furthermore, the format of flight data to be analyzed will vary from organization to organization, more likely from project to project within the organization itself. Nevertheless, the basic parameter estimation algorithm remains the same. We provide in this book a simple version of software for general nonlinear systems. It caters for estimating the system parameters and zero shifts from multiple experiments, but treats the initial conditions as known. However, this is not a serious limitation, because based on the discussion in Chapter 3, Sec. VII, reasonable values can be specified. In the case of linear systems, the bias parameters of the state and observation equations can be included and estimated along with the system parameters. The sophisticated features discussed in Sec. XVIII are also not included in this simpler version. The software is adequate to analyze the test cases that will be considered in this and few other chapters. Furthermore, it allows us to understand more closely the various algorithmic steps that we have so far studied in this

chapter. The software that is provided runs under Matlab^{®,58} mainly because the current trend shows that in the future more and more flight vehicle system identification work will be carried out under the Matlab operating environment.

Extension of the basic software to cater for individual needs or further advanced options is left to the reader. Such extensions will be required to 1) estimate initial conditions, 2) allow estimation of time delays, and 3) treat bias parameters separately for multiple time segments being analyzed simultaneously. The logic behind these extensions has been discussed in Chapter 3. Programming of the first and the third extension would require treating the complete set of unknowns in different sets, namely 1) system parameters (aerodynamic derivatives) common to all time segments and 2) separate bias parameters as well as initial conditions for each of the n_E experiments.

Other estimation software packages are available as Matlab Toolboxes based on restricted circulation or at minor a license fee.^{59,60} It is also possible to include models developed in Matlab/Simulink[®] using the block-diagram functionality.⁶⁰ Matlab and Simulink provide an easy-to-use computing and modeling description environment. We do not go into the details of such software tools, except to say that they are based on the basic theory that we have covered in this book. More sophisticated software tools in FORTRAN language have been developed in the past, and extensively used for flight vehicle system identification, for example Refs. 57, 61, and 62 among others. Here too, we do not go into any further details of these software tools.

Figure 4.6 shows a flow chart of the computational procedure for the output error method. The function names for various parts are indicated at the bottom of each block. The source codes (Matlab m-files) for output error method are provided in the directory /FVSysID/chapter04/, and a README providing the details of various functions and test cases covered here. The starting point is the main program is called “ml_oem,” and provides the user interface to define the model, choose the optimization and integration methods, and other information:

test_case	index for the test case
iOptim	index for the optimization method
integMethod	1: Gauss–Newton; 2: Levenberg–Marquardt
niter_max	flag for integration method (ruku4, ruku3, ruku2)
tolR	maximum number of iterations
par_step	relative change in the cost function for termination
	parameter perturbation size for gradient approximations

Default values are defined for the maximum number of iterations niter_max, for the convergence criterion tolR, and for the step size for parameter perturbations par_step. They can be suitably changed, if necessary.

A particular system model to be analyzed is denoted through the integer flag “test_case,” which is uniquely associated with the state space model and with the user-provided interface in the form of a function defining the model parameters, flight data and other relevant details. Different state space models, to be coded by

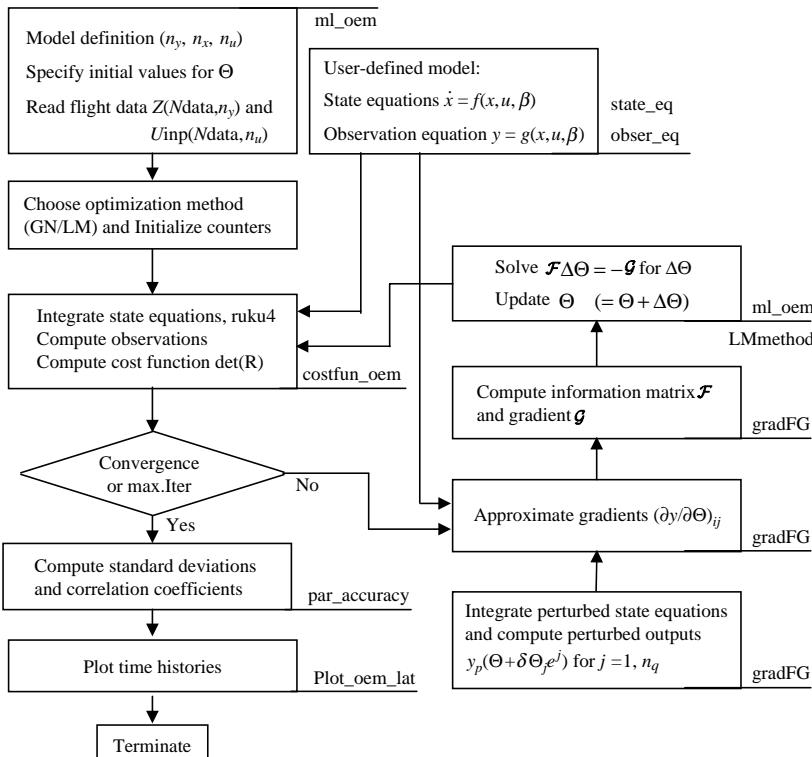


Fig. 4.6 Details of implemented output error method.

the user, are invoked by defining the following flags and strings in a function `mDefCase**` called in “`ml_oem`,” where `**` denotes the test case index:

<code>state_eq</code>	function name: to code the right-hand side of state equations
<code>obser_eq</code>	function name: to code the right-hand side of observation equations
<code>Nx</code>	number of state variables (n_x)
<code code="" ny<=""></code>	number of output variables (n_y)
<code>Nu</code>	number of input variables (n_u)
<code>Nparsys</code>	number of system parameters
<code>Nparam</code>	total number of system and bias parameters
<code>iArtifStab</code>	flag for artificial stabilization (= 0 for OEM)
<code>param</code>	starting values for unknown parameters (Θ)
<code>parFlag</code>	flags for free and fixed parameters = 1: free parameters (to be estimated) = 0: fixed parameters (not to be estimated)
<code>NparID</code>	total number of unknown parameters, that is, nonzero elements of <code>parFlag</code> (n_q)

x0	initial conditions on state variables (kept fixed)
StabMat	artificial stabilization matrix (n_x, n_y)

The option of artificial stabilization is not relevant for the present chapter, but it is required in Chapter 9, which extends the basic OEM method to unstable aircraft. In all examples covered in this book, except for those in Chapter 9, the flag iArtifStab and the matrix StabMat are dummy variables, and we initialize them to zero and a null matrix.

The variable names used in the program ml_oem and calling functions are shown in the above list on the left-hand side, followed by the description and the notation used to denote these variables in the text. It is obvious that model specification in terms of (N_x, N_y, N_u, N_{param}) must match those coded in the above two user functions for the postulated model. It is also required to specify suitable starting values for the parameters Θ and the integer flags parFlag to indicate whether the particular parameter is free (to be estimated) or fixed (not to be estimated). This option to handle parameters as either unknowns or as known, or not identifiable from data analyzed, and hence to be kept fixed, makes the software flexible. For example, the function codes for the state and observation equations with a larger number of parameters can be used without modification to estimate a subset of them, depending upon the data available.

The flight data to be analyzed is also loaded in the function for model definition, and requires specification or assignment of the following information:

Ndata	number of data points
dt	sampling time
t	time vector [= 0 to ($N_{data}-1$)*dt]
Z(Ndata,Ny)	flight data for measured outputs (N, n_y),
Uinp(Ndata,Nu)	flight data for measured control inputs (N, n_u)

While loading the flight data in the arrays Z(Ndata,Ny) and Uinp(Ndata,Nu), it is obvious that the units of the variables must match those used in the postulated model and programmed in the state and observation functions “state_eq” and “obser_eq,” respectively.

Since we provide an option to analyze multiple experiments, it is necessary to define additionally the following time segment related information:

Nzi	number of time segments to be analyzed simultaneously (n_E)
izhf	cumulative index at which the maneuvers end when concatenated.

For example, if we analyze three time segments, each having N_{ts1}, N_{ts2} and N_{ts3} data points respectively, then izhf is defined as [$N_{ts1}; N_{ts1} + N_{ts2}; N_{ts1} + N_{ts2} + N_{ts3}$]. The total number of data points, Ndata, is just the sum of data points for each of the Nzi segments. In this chapter we will be dealing with test cases analyzing a single maneuver at a time, but the option of multiple experiments will be invoked in examples covered in other chapters.

From Fig. 4.6 it can be seen that the function “costfun_oem” integrates the states and computes the observations and the cost function value; the function “gradFG” propagates the perturbed state equations, computes the perturbed system responses and computes the gradient vector \mathcal{G} and information matrix \mathcal{F} ; the parameter updates $\Delta\Theta$ are computed in the main program “ml_oem” itself if the Gauss–Newton method is applied, otherwise the function LMmethod for the Levenberg–Marquardt method is called. We prefer to use Cholesky factorization to solve for $\Delta\Theta$; other methods can be easily invoked in the main program. The convergence checking is performed in “ml_oem,” so also using the step-size control including halving of the parameter steps to overcome intermediate divergence. At the end, we compute the standard deviations and correlation coefficients in “par_accuracy” and finally make plots of the time histories and of estimates.

For each test case, a program is provided to plot time histories of the control variables, and generate superimposed plots of measured and computed output variables that allow direct comparison. In a separate figure, the estimates plotted against iteration count show the convergence of the optimization method. The vertical bars are provided, showing the standard deviations. The plot programs are called “plots_TCnn_oem_ttt.m,” where nn refers to the test case index and ttt to additional explanation usually giving details of model or aircraft analyzed.

XX. Examples

Although we will cover some typical examples related to flight vehicles in a separate chapter, we consider here two simple examples to demonstrate the output error method and the use of the software provided. The first example pertains to analysis of flight data recorded during an aileron and rudder input maneuver, performed under seemingly steady atmospheric conditions. To this example, we will apply both the Gauss–Newton and Levenberg–Marquardt methods. The second example also pertains to the lateral-directional motion, analyzing simulated data that has been generated with a moderate to high level of atmospheric turbulence. The test cases are selected through the flag “test_case” and run using the program “ml_oem.” The flight data analyzed in these cases is supplied to the readers as sample flight data for verification and to try out other options and possibly to verify software that they may modify or develop themselves. These data files are found under the directory /FVSysID/flt_data/.

A. Lateral-directional Motion

The model pertaining to the lateral-directional motion is postulated as follows:
State equations

$$\begin{aligned}\dot{p} &= L_p p + L_r r + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r + L_\beta \beta + b_{xp} \\ \dot{r} &= N_p p + N_r r + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r + N_\beta \beta + b_{xr}\end{aligned}\quad (4.90)$$

Observation equations

$$\begin{aligned}
 \dot{p}_m &= L_p p + L_r r + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r + L_\beta \beta + b_{y\dot{p}} \\
 \dot{r}_m &= N_p p + N_r r + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r + N_\beta \beta + b_{y\dot{r}} \\
 a_{ym} &= Y_p p + Y_r r + Y_{\delta_a} \delta_a + Y_{\delta_r} \delta_r + Y_\beta \beta + b_{yay} \\
 p_m &= p + b_{yp} \\
 r_m &= r + b_{yr}
 \end{aligned} \tag{4.91}$$

where the subscript m denotes the measured variables, p the roll rate, r the yaw rate, a_y the lateral acceleration, \dot{p} the roll acceleration, \dot{r} the yaw acceleration, δ_a the aileron deflection, δ_r the rudder deflection, β the angle of sideslip, and $L_{(.)}$, $N_{(.)}$, and $Y_{(.)}$ the dimensional aerodynamic derivatives. As discussed in Chapter 3, Sec. V.B, the postulated model contains bias terms for the state and observation equations $b_{x(.)}$ and $b_{y(.)}$, respectively. The readers might recall that the lumped bias parameters provide a convenient means to appropriately account for the initial conditions and systematic errors in the measurements. Besides the aileron and rudder inputs, as discussed in Chapter 3, Sec. VI, the angle of sideslip β is treated as a pseudo-control input. The unknown parameter vector Θ consisting of the dimensional derivatives and the bias parameters is given by:

$$\Theta^T = [L_p \ L_r \ L_{\delta_a} \ L_{\delta_r} \ L_\beta \ N_p \ N_r \ N_{\delta_a} \ N_{\delta_r} \ N_\beta \ Y_p \ Y_r \ Y_{\delta_a} \ Y_{\delta_r} \ Y_\beta \ b_{x\dot{p}} \ b_{x\dot{r}} \ b_{y\dot{p}} \ b_{y\dot{r}} \ b_{yay} \ b_{yp} \ b_{yr}] \tag{4.92}$$

Thus, the postulated model is defined as follows:

	No. of variables	Function name
States	2 p, r	“xdot_TC01_attas_lat”
Outputs	5 $\dot{p}, \dot{r}, a_y, p, r$	“obs_TC01_attas_lat”
Inputs	3 $\delta_a, \delta_r, \beta$	

The right-hand sides of the state equation, Eq. (4.90), are programmed in the function “xdot_TC01_attas_lat,” and those of the observation equation, Eq. (4.91), in “obs_TC01_attas_lat.” Thus, the model definition for this example provided by the function “/FVSysID/chapter04/mDefCase01.m,” which is called from the main program “/FVSysID/chapter04/ml_oem,” is as follows:

```

test_case = 1; % index for the test case
state_eq = 'xdot_TC01_attas_lat'; % function for state equations
obser_eq = 'obs_TC01_attas_lat'; % function for observation equations
Nx = 2; % number of states
Ny = 5; % number of observation variables
Nu = 3; % number of input (control) variables
NparSys = 15; % number of system parameters

```

$$\begin{aligned} \text{Nparam} &= \text{NparSys} + \text{Nx} + \text{Ny}; & \% \text{ total number of parameters} \\ \text{dt} &= 0.04; & \% \text{ sampling time} \end{aligned}$$

Thus, the index for test case is set to one in the main program `ml_oem`. The flight data analyzed were gathered from flight tests carried out with the research aircraft ATTAS.⁶³ The rolling and yawing motions were excited in a sequence through pilot-applied aileron and rudder inputs. The flight maneuvers were carried out under seemingly steady atmospheric conditions, when the turbulence was considered negligible. The data was recorded with a sampling time of 40 ms and the duration of the maneuver was 60 s. The resulting lateral-directional motion was recorded and analyzed as a single maneuver. The data to be analyzed is loaded from the data file “/FVSSysID/flt_data/fAttasAilRud1.mat.” The number of time segments is given by $\text{Nzi} = 1$ and the izhf is set to the total number of data points. The output and input variables in data arrays $Z(\text{Ndata}, \text{Ny})$ and $Uinp(\text{Ndata}, \text{Nu})$ are appropriately assigned; a list of channels recorded is provided in the README. Suitable starting values are specified for the unknown parameters; they are mostly 50% off the expected values. The integer flags `parFlag` are set to 1, indicating that all parameters are to be estimated. The flag `integMethod` is set to `ruku4` for the fourth order Runge–Kutta integration method. All these details can be easily traced from the model definition function “`mDefCase01.m`.”

The results of parameter estimation applying the output error method are summarized in Table 4.1, showing convergence obtained from the Gauss–Newton and Levenberg–Marquardt methods. It is observed that the Gauss–Newton method shows intermediate divergence during the first iteration. As pointed out in Sec. XII.A, the heuristic approach of halving the parameter update proved quite sufficient to reduce the cost function through a single step. The subsequent convergence was smooth and the process was terminated when the relative change in the cost function, i.e., determinant of the covariance matrix R , was less than 0.0001. In the case of the Levenberg–Marquardt method (selected by

Table 4.1 Convergence of Gauss–Newton and Levenberg–Marquardt methods for test case 1

Iteration	Gauss–Newton: cost function	Levenberg–Marquardt: cost function (LM parameter)
0	1.3568×10^{-11}	1.3568×10^{-11}
1	3.6039×10^{-10}	3.5663×10^{-10} ($\lambda: 1.0 \times 10^{-2}$)
	6.7975×10^{-15}	3.2505×10^{-10} ($\lambda: 1.0 \times 10^{-1}$)
		4.0265×10^{-13} ($\lambda: 1.0 \times 10^{-1}$)
2	2.437×10^{-19}	8.5981×10^{-18} ($\lambda: 1.0 \times 10^{-2}$)
3	9.3597×10^{-20}	1.6038×10^{-19} ($\lambda: 1.0 \times 10^{-3}$)
4	9.1717×10^{-20}	9.2428×10^{-20} ($\lambda: 1.0 \times 10^{-4}$)
5	9.1702×10^{-20}	9.1707×10^{-20} ($\lambda: 1.0 \times 10^{-5}$)
6	9.1702×10^{-20}	9.1702×10^{-20} ($\lambda: 1.0 \times 10^{-6}$)

setting the integer flag $iOptim = 2$ in `ml_oem`), we start with the same initial parameter values and cost function, and the default value for the LM parameter $\lambda = 0.001$, as suggested in Sec. XIII. During the first iteration, intermediate divergence is encountered; at this stage the algorithm adapts the LM parameter λ until a reduction in the cost function is achieved. In the present case, this required three steps, changing λ from 0.001 to 0.01; note that during the third step of LM parameter adjustment, λ is not reduced as a different condition is satisfied here; see Sec. XIII. The λ values are shown in brackets in the third column of Table 4.1. During the subsequent iterations, it turns out that the Gauss–Newton directions are the best choice and the optimization progresses smoothly; hence λ is reduced continuously from iteration to iteration. Convergence is achieved in the same number of iterations, which is more or less a coincidence. Depending upon the starting values and the nature of the cost function, the two methods may progress towards the minimum differently and may require a different number of iterations.

The time history match is shown in Fig. 4.7 only once, because there was no qualitative difference between the two methods. The three plots from the bottom show the control inputs, namely the angle of sideslip, the rudder and aileron deflections, respectively. The five plots from the top show the match between the flight-measured responses (shown as continuous lines) and the model-estimated responses (shown as dashed lines). The match between the two is excellent. The computations in the postulated model are carried out in radians and radians per second, whereas the time histories are plotted in degrees and degrees per second.

The convergence plots for the derivatives $L_{(.)}$ and $N_{(.)}$ obtained from the Gauss–Newton method are provided in Fig. 4.8, showing the error bars, i.e., standard deviations, given by Eq. (4.88). All the parameters were estimated with very low standard deviations and there was no correlation between the aerodynamic derivatives. Numerical values and the standard deviations are not provided here, but it can be easily verified that both the methods yield the same results within the numerical accuracy and round-off errors.

B. Application of Output Error Method to Data with Process Noise

To evaluate the performance of more advanced estimation algorithms accounting for process noise that we will be addressing in the next chapter, aircraft responses pertaining to lateral-directional motion with an appreciable level of turbulence were generated through simulation.⁶⁴ The nominal values of the aerodynamic derivatives used correspond to those obtained from flight tests carried out in steady atmospheric conditions with the research aircraft de Havilland DHC-2. Equations (4.90) and (4.91), incorporating additional state and measurement noise, are used to generate the data. The angle of sideslip has been replaced through the lateral-velocity component v , and the corresponding derivatives too. To provide realistic control inputs, the rudder and aileron excitations actually applied in a particular flight test are used to excite the postulated model. Independent process and measurement noise vectors are generated using pseudo-random noise generators. The state noise matrix was assumed to be diagonal. Note that state variables p and r are only affected by the state noise, and not

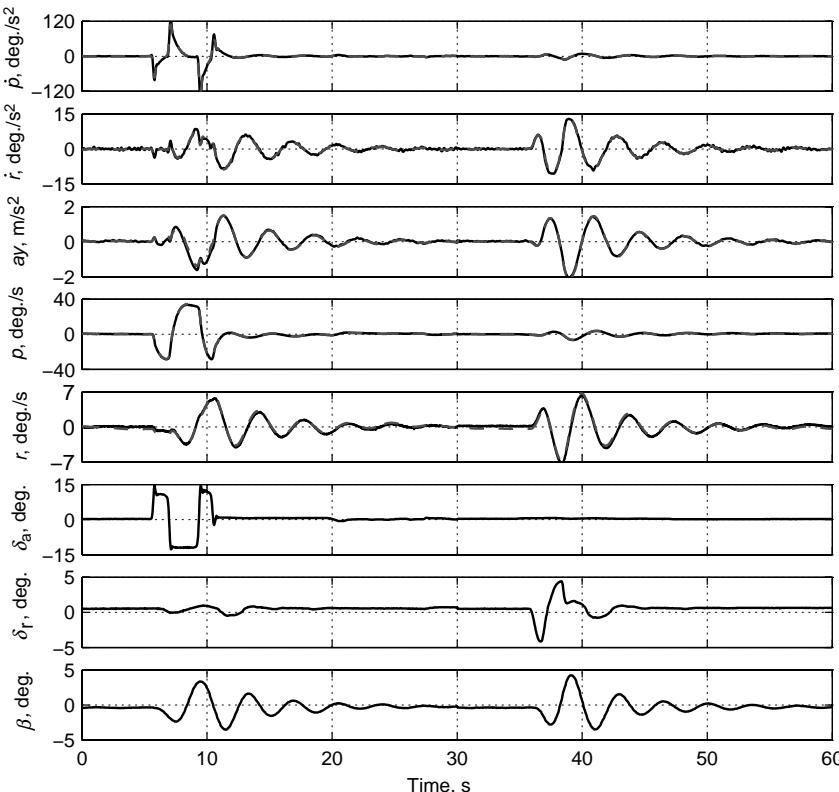


Fig. 4.7 Time histories of control inputs and a comparison of flight measured and model predicted outputs for test case 1 (—, flight measured; ---, model predicted).

the other control inputs δ_a , δ_r , and ν . A total of 16 s of data with a sampling time of 0.05 s were generated. These simulated responses were analyzed applying parameter estimation methods; see Fig. 4.9. The time histories of these data are provided as /FVsysID/flt_data/y13aus_da1.asc.

To these data with process noise we apply the output error method. We use the model postulated in Eqs. (4.90) and (4.91) to estimate the unknown parameter vector of Eq. (4.91), except that, as already pointed out, the pseudo-control input β has been replaced with ν . This test case is designated as test_case = 2. The model definition is very similar to that of test_case = 1. The data to be analyzed, that is, the arrays Z(Ndata,Ny) and Uinp(Ndata,Nu), are loaded from the file /FVsysID/flt_data/y13aus_da1.asc. The model definition for this example is provided by the function “/FVsysID/chapter04/mDefCase02.m,” which is called from the main program “/FVsysID/chapter04/ml_oem” for test_case = 2. To apply the Gauss–Newton, we set the integer flag iOptim = 1.

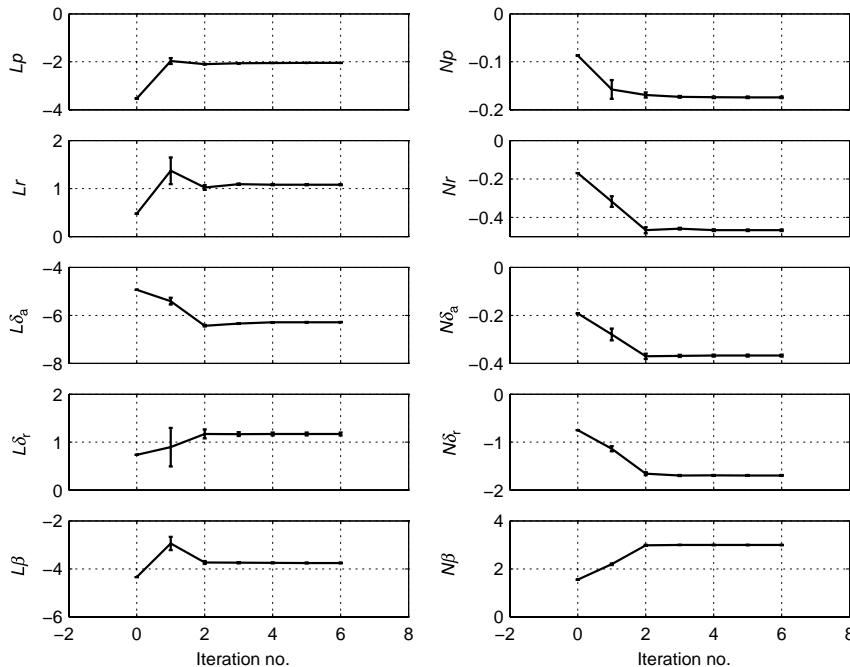


Fig. 4.8 Convergence plot of dimensional derivatives estimated from flight data for test case 1.

In this case, it is observed that the output error method has severe convergence problems. Almost at every iteration, intermediate divergence is encountered. The process terminates after 42 iterations, yielding parameter values that are not anywhere near the expected values. The estimated parameters are not specified here, because we will get back to this case in Chapter 5, when we apply the filter error method and make a comparison with the output error method. The time history plots are provided in Fig. 4.9, showing the control inputs and the match for the output variables. The discrepancies in the match are evident. Although this example is a simple one, it helps to demonstrate the limitations of the output error method when we encounter process noise, that is, for flight data gathered under turbulent atmospheric conditions.

XXI. Concluding Remarks

In this chapter we presented the general principle of maximum likelihood estimation, highlighting the exact physical interpretation, and showed that it can be applied to nonlinear dynamical systems of arbitrary complexity. It was shown that the maximum likelihood estimator has several desirable statistical properties, namely unbiased estimates, asymptotic consistency, normality, and efficiency. The advantages of the method are that it makes optimal use of the measured data and as a byproduct provides, without extra computations, statistical

OUTPUT ERROR METHOD

125

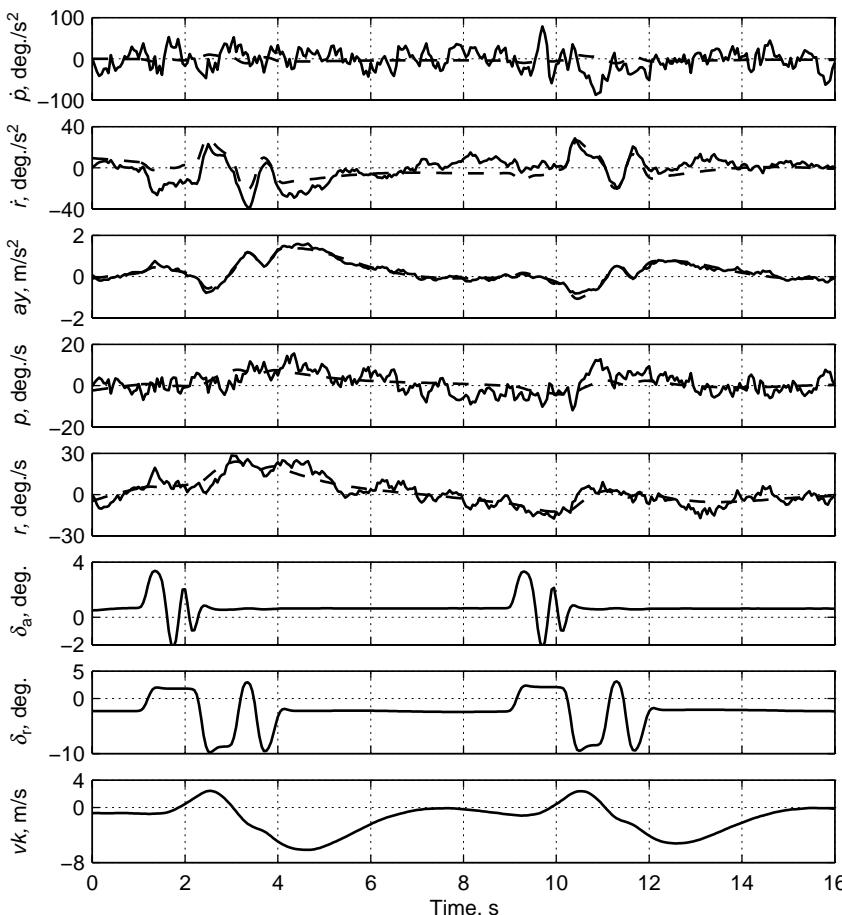


Fig. 4.9 Output error method applied to data with moderate to high level of turbulence, test case 2; (—, flight measured; ---, model predicted).

information on the accuracy of the parameter estimates in terms of the standard deviations, and correlation coefficients. The subtle issues associated with evaluation of these statistical accuracies are deferred to Chapter 11, dealing with model validation.

The output error method is one of the most widely applied algorithms to estimate aircraft parameters. It considers measurement noise only, but this assumption simplifies the algorithm significantly. It is also considered acceptable from a practical viewpoint, because we generally carry out flight tests specifically for parameter estimation purposes, and as such these flights can also be carried out when the atmospheric conditions are stable. The relaxation strategy provides the best approach to optimize the likelihood function, and overcomes convergence problems through the closed-form solution for the estimation of the

measurement noise covariance matrix. We studied in some depth the three most commonly applied optimization methods, namely unconstrained Gauss–Newton, bounded-variable Gauss–Newton and Levenberg–Marquardt. Various options to overcome the problems of intermediate divergence in the unconstrained Gauss–Newton have been presented; these finer practical details are very important to enlarge the scope of application of these methods. Several other numerical aspects, like approximation of sensitivity coefficients, nonlinear model structures, and structure of the sensitivity matrix for large-scale systems, were discussed.

We also covered briefly the aspects of Bayesian estimation, the most general form of an estimator, and showed that a priori information about the aerodynamic derivatives can be accounted for in the output error method. We further argued that, owing to uncertainties in the error covariance matrix for such a priori values, the approach is rarely used in practice.

Finally, samples of flight data and basic software implemented under Matlab have been provided to follow the various computational steps of the algorithm. It has been applied to the two examples that were presented to demonstrate the use of the software and to evaluate the performance of the output error method.

References

- ¹Fisher, R. A., “On an Absolute Criterion for Fitting Frequency Curves,” *Messenger of Mathematics*, Macmillan, London, Vol. 41, 1912, pp. 155–160.
- ²Fisher, R. A., “On the Mathematical Foundations of Theoretical Statistics,” *Philosophical Transactions of the Royal Society of London*, Vol. 222, 1922, pp. 309–368.
- ³Fisher, R. A., “Theory of Statistical Estimations,” *Proceedings of the Cambridge Philosophical Society*, Vol. 22, 1925, pp. 700–725.
- ⁴Iliff, K. W., “Parameter Estimation for Flight Vehicles,” *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 5, 1989, pp. 609–622.
- ⁵Klein, V., “Estimation of Aircraft Aerodynamic Parameters from Flight Data,” *Progress in Aerospace Sciences*, Pergamon, Oxford, UK, Vol. 26, No. 1, 1989, pp. 1–77.
- ⁶Hamel, P. G. and Jategaonkar, R. V., “Evolution of Flight Vehicle System Identification,” *Journal of Aircraft*, Vol. 33, No. 1, 1996, pp. 9–28.
- ⁷Jategaonkar, R. V., Fischenberg, D., and von Gruenhagen, W., “Aerodynamic Modeling and System Identification from Flight Data—Recent Applications at DLR,” *Journal of Aircraft*, Vol. 41, No. 4, 2004, pp. 681–691.
- ⁸Wang, K. C. and Iliff, K. W., “Retrospective and Recent Examples of Aircraft Parameter Identification at NASA Dryden Flight Research Center,” *Journal of Aircraft*, Vol. 41, No. 4, 2004, pp. 752–764.
- ⁹Morelli, E. A. and Klein, V., “Application of System Identification to Aircraft at NASA Langley Research Center,” *Journal of Aircraft*, Vol. 42, No. 1, 2005, pp. 12–25.
- ¹⁰Kendall, M. G. and Stuart, A., *The Advanced Theory of Statistics*, 3rd ed., Griffin, London, 1969.
- ¹¹Waerden, B. L., *Mathematical Statistics*, Springer, New York, 1969.
- ¹²Sorenson, H. W., *Parameter Estimation—Principles and Problems*, Marcel Dekker, New York, 1980.
- ¹³Davenport, W. B. and Root, W. L., *Random Signals and Noise*, McGraw-Hill, New York, 1958.

- ¹⁴Goodwin, G. C. and Payne, R. L., *Dynamic System Identification*, Academic Press, New York, 1977.
- ¹⁵Wald, A., "Note on the Consistency of the Maximum Likelihood Estimates," *Annals of Mathematical Statistics*, Vol. 20, 1949, pp. 595–601.
- ¹⁶Cramér, H., *Mathematical Methods of Statistics*, Princeton University Press, Princeton, NJ, 1946.
- ¹⁷Eykhoff, P., *System Identification—Parameter and State Estimation*, John Wiley & Sons, New York, 1977.
- ¹⁸Maine, R. E. and Iliff, K. W., "Identification of Dynamic Systems," AGARD AG-300, Vol. 2, Jan. 1985.
- ¹⁹Iliff, K. W., "Maximum Likelihood Estimation of Lift and Drag from Dynamic Aircraft Maneuvers," *Journal of Aircraft*, Vol. 14, No. 12, 1977, pp. 1175–1181.
- ²⁰Trankle, T. L., Vincent, J. H., and Franklin, S. N., "System Identification of Nonlinear Aerodynamic Models," AGARD AG-256, Paper 7, March 1982.
- ²¹Jategaonkar, R. V. and Plaetschke, E., "Maximum Likelihood Estimation from Flight Test Data for General Non-Linear Systems," DFVLR-FB 83-14, March 1983.
- ²²Taylor, L. W. and Iliff, K. W., "System Identification Using a Modified Newton-Raphson Method—A Fortran Program," NASA TN D-6734, May 1972.
- ²³Plaetschke, E., "Kennwertermittlung mit Maximum-Likelihood-Verfahren," DFVLR-IB 154-74/20, 1974 (in German).
- ²⁴Iliff, K. W. and Taylor, L. W., "Determination of Stability Derivatives from Flight Data Using a Newton-Raphson Minimization Technique," NASA TN D-6579, March 1972.
- ²⁵Ortega, J. M. and Rheinboldt, W. C., *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
- ²⁶Balakrishnan, A. V. (ed.), *Communication Theory*, McGraw Hill, New York, 1968.
- ²⁷Kumar, K. S. P. and Sridhar, R., "On the Identification of Control Systems by the Quasi-Linearization Method," *IEEE Transactions on Automatic Control*, Vol. AC-9, No. 2, 1964, pp. 151–154.
- ²⁸Larson, D. B. and Fleck, J. T., "Identification of Parameters by the Method of Quasi-linearization," Cornell Aeronautical Laboratory, Buffalo, NY, CAL Report No. 164, May 1968.
- ²⁹Jategaonkar, R. V. and Plaetschke, E., "Maximum Likelihood Estimation of Parameters in Nonlinear Flight Mechanics Systems," *Proceedings of the IFAC Symposium on "Identification and System Parameter Estimation"*, York, 1985, pp. 665–668.
- ³⁰Jategaonkar, R. V. and Thielecke, F., "Aircraft Parameter Estimation—A Tool for Development of Aerodynamic Databases," *Sadhana*, Vol. 25, Part 2, Indian Academy of Sciences, Bangalore, April 2000, pp. 119–135.
- ³¹Spall, J. C., "An Overview of the Simultaneous Perturbation Method for Efficient Optimization," *John Hopkins APL Technical Digest*, Vol. 19, No. 4, 1998, pp. 482–492.
- ³²Bischoff, C., Carle, A., Khademi, P., and Mauer, A., "ADIFOR 2.0: Automatic Differentiation of Fortran-77 Programs," *IEEE Computational Science and Engineering*, Vol. 3, No. 3, 1996, pp. 18–32.
- ³³Verma, A., "An Introduction to Automatic Differentiation," *Current Science*, Vol. 78, No. 7, 2000, pp. 804–807.
- ³⁴Foster, G. W., "The Identification of Aircraft Stability and Control Parameters in Turbulence," RAE TR-83025, March 1983.

- ³⁵Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, San Diego, CA, 1981.
- ³⁶Jategaonkar, R. V., "Bounded-Variable Gauss–Newton Algorithm for Aircraft Parameter Estimation," *Journal of Aircraft*, Vol. 37, No. 4, 2000, pp. 742–744.
- ³⁷Oswald, W. B., "General Formulas and Charts for the Calculation of Airplane Performance," NACA Report No. 408, 1932.
- ³⁸Weiss, S., Gockel, W., Mönnich W., and Rohlf, D., "Identification of Dornier-328 Reversible Flight Control Systems," AIAA Paper 98-4163, Aug. 1998.
- ³⁹Rao, S. S., *Engineering Optimization: Theory and Practice*, John Wiley & Sons, New York, 1996.
- ⁴⁰Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C., "A Limited Memory Algorithm for Bound Constrained Optimization," *SIAM Journal of Scientific Computing*, Vol. 16, No. 5, 1995, pp. 1190–1208.
- ⁴¹Jacob, H. G., "An Engineering Optimization Method with Application to STOL-Aircraft Approach and Landing Trajectories," NASA TN D-6978, Sept. 1972.
- ⁴²Gill, P. E. and Murray, W., "Minimization Subject to Bounds on the Variables," National Physical Laboratory, Middlesex, Report NAC-72, Dec. 1976.
- ⁴³Stark, P. B. and Parker, R. L., "Bounded Variable Least Squares: An Algorithm and Applications," *Journal of Computational Statistics*, Vol. 10, No. 2, 1995, pp. 129–141.
- ⁴⁴Vanderbei, R. J. and Shanno D. F., "An Interior-Point Algorithm for Nonconvex Nonlinear Programming," *Computational Optimization and Applications*, Vol. 13, No. 1–3, 1999, pp. 231–252.
- ⁴⁵Handschin, E., Langer, M., and Kliokys, E., "An Interior Point Method for State Estimation with Current Magnitude Measurements and Inequality Constraints," *Proceedings of the IEEE Power Industry Computer Application Conference*, Salt Lake City, UT, 1995, pp. 385–391.
- ⁴⁶Lustig, I. J., Marsten, R. J., and Shanno, D. F., "Computational Experience with a Primal-Dual Interior-Point Method for Linear Programming," *Linear Algebra and its Applications*, Vol. 152, 1991, pp. 191–222.
- ⁴⁷Marquardt, D. W., "An Algorithm for Least Squares Estimation of Nonlinear Parameters," *SIAM Journal of Applied Mathematics*, Vol. 11, No. 2, 1963, pp. 431–441.
- ⁴⁸Moré, J. J., "The Levenberg–Marquardt Algorithm: Implementation and Theory," in *Springer Lecture Notes in Mathematics*, No. 630, edited by G. A. Watson, Springer, Berlin, 1978, pp. 105–116.
- ⁴⁹Pashikar, A. A. and Jategaonkar, R. V., "Some Investigations Related to Gradients for Aircraft Parameter Estimation," DLR IB 111-98/44, Nov. 1998.
- ⁵⁰Jategaonkar, R. V. and Plaetschke, E., "Non-Linear Parameter Estimation from Flight Test Data Using Minimum Search Methods," DFVLR-FB 83-15, March 1983.
- ⁵¹Murphy, P. C., "A Methodology for Airplane Parameter Estimation and Confidence Interval Determination in Nonlinear Estimation Problems," NASA RP 1153, April 1986.
- ⁵²Nelder, J. A. and Mead, R., "A Simplex Method for Function Minimization," *Computer Journal*, Vol. 7, Issue 4, Jan. 1965, pp. 308–313.
- ⁵³Rowan, T., "The Subplex Method for Unconstrained Optimization," Ph.D. Thesis, Dept. of Computer Sciences, Univ. of Texas, Austin, TX, 1990, pp. 50–74.
- ⁵⁴Powell, M. J. D., "An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Gradients," *Computer Journal*, Vol. 7, 1964, pp. 155–162.

⁵⁵Wilks, S. S., *Mathematical Statistics*, John Wiley & Sons, New York, 1962.

⁵⁶Jategaonkar, R. V. and Mönnich, W., "Identification of DO-328 Aerodynamic Database for a Level D Flight Simulator," AIAA Paper 97-3729, Aug. 1997.

⁵⁷Jategaonkar, R. V. and Thielecke, F., "ESTIMA—An Integrated Software Tool for Nonlinear Parameter Estimation," *Journal of Aerospace Science and Technology*, Vol. 6, No. 8, 2002, pp. 565–578.

⁵⁸"MATLAB, The Language of Technical Computing, Version 6.5.1, Release 13, (Service Pack 1)," The MathWorks Inc., Natick, MA, 2003.

⁵⁹Morelli, E. A., "System Identification Programs for Aircraft (SIDPAC)," AIAA Paper 2002-4704, Aug. 2002.

⁶⁰Weiss, S., "PENSUM: Parameter Estimation of Nonlinear Systems Using Matlab, Version 1.0," DLR-IB 111-1999/32, Dec. 1999.

⁶¹Maine, R. E. and Iliff, K. W., "User's Manual for MMLE3, a General FORTRAN Program for Maximum Likelihood Parameter Estimation," NASA TP-1563, Nov. 1980.

⁶²Murray, J. E. and Maine, R. E., "pEst Version 2.1 User's Manual," NASA TM-88280, Sept. 1987.

⁶³Jategaonkar, R. V., "Identification of the Aerodynamic Model of the DLR Research Aircraft ATTAS from Flight Test Data," DLR-FB 90-40, July 1990.

⁶⁴Jategaonkar, R. V. and Plaetschke, E., "Algorithms for Aircraft Parameter Estimation Accounting for Process and Measurement Noise," *Journal of Aircraft*, Vol. 26, No. 4, 1989, pp. 360–372.

This page intentionally left blank

Chapter 5

Filter Error Method

I. Introduction

THE MAXIMUM likelihood estimates of model parameters accounting only for measurement noise can be efficiently obtained for linear and general non-linear systems. In Chapter 4, we discussed the output error method for this purpose in some detail, presenting the most commonly applied optimization algorithms and several practical issues. Although the output error method has been widely used in the past, and will continue to be used in the future as well, it is necessary to gather the data for estimation purposes from flight tests in a steady atmosphere. In the presence of atmospheric turbulence, the output error method yields poor estimation results, in terms of both convergence and estimates; we saw one such example in Chapter 4, Sec. XX.B.

There are two ways to account for turbulence in parameter estimation, 1) to measure the wind components, or more appropriately to derive them from other measured variables such as true airspeed, inertial speed, attitude angles, and flow angles, and 2) to model generically or explicitly the turbulence mathematically and estimate the corresponding parameters. The first approach is a data pre-processing step that yields wind components along the three body-fixed coordinates, those can be treated as known inputs and accounted for in the estimation through minor modifications of the postulated models. The advantage is that the fairly simple output error method can be applied directly. The approach, however, requires precise measurements of the said variables. Any inaccuracies in the measurements, for example those resulting from calibration errors or time delays in the recorded flow angles, will affect the derived wind, and consequently the estimates of the aerodynamic derivatives. Furthermore, accounting for the effects of turbulence on other motion variables, for example angular rates, is more involved and generally not considered. Nevertheless, it is a viable approach. The second approach of mathematically accounting for turbulence is algorithmically more involved, but at the same time comparatively more flexible, because it is not limited by the difficulties just mentioned. Here, we consider only the second approach, because it has been used more commonly in practice than the other.

The presence of process noise (turbulence) complicates parameter estimation considerably. Since the nonmeasurable process noise makes the system stochastic, as a consequence, it requires a suitable state estimator to propagate the states;

for this reason we avoid here the term “integrate the state equations.” The state estimation is performed using either a Kalman filter or an extended Kalman filter, depending upon whether the postulated model is linear or nonlinear. There are two possible approaches to account for process noise in parameter estimation: 1) techniques belonging to the general class of output-error, and 2) the filtering approach. The second approach is an indirect approach which transforms the parameter estimation problem into a nonlinear state estimation problem by artificially defining the unknown parameters as additional state variables. We will discuss this approach and its limitations in Chapter 7. In this chapter we focus on the first approach, namely the techniques called filter error methods (FEM). These techniques are a logical extension of the output error method. Such methods accounting for both process and measurement noise, although essentially more complex, are not only necessary to analyze flight-test data from flights in turbulent atmosphere, but can even yield improved estimation results for flight data in a seemingly smooth air.

Filter error methods represent the most general stochastic approach to aircraft parameter estimation that was proposed by Balakrishnan.¹ With the pioneering work of Mehra^{2–4} and Iliff⁵ during the early 1970s, these techniques provided capabilities to estimate aircraft parameters from flight in turbulent atmosphere. Highly successful applications of estimation methods to linear systems, combined with the fact that the linear filtering theory provides an optimal solution to state estimation in linear systems, had paved the way for the use of filter error methods for linear systems.⁶ Several applications of aircraft parameter estimation have since been reported.^{7–9} Although extension of the filter error methods to nonlinear systems is a natural further step, such extensions and applications are particularly complicated. During the 1980s Jategaonkar and Plaetschke demonstrated a workable approach to handling general model structures using a numerical approach and incorporating a mixed state estimator based on a prediction step with the nonlinear model and a correction step using a first-order linear approximation.^{10–12}

Besides the capability to account for process and measurement noise, the filter error methods offer more advantages; they lead, in general, to a more nearly linear optimization problem that has fewer local minima and a faster convergence rate, because these methods are less sensitive to stochastic disturbances. Owing to the basic formulation which we will address in this chapter, they contain a feedback proportional to the fit error. This feedback numerically stabilizes the filter error algorithm and also helps to improve the convergence properties. The stabilizing property of the filter error algorithm makes it suitable for open-loop identification of unstable aircraft. We will address this issue of unstable aircraft in more details separately, but the filter error methods work for unstable aircraft; output error methods may not.

Figure 5.1 shows a block schematic of the filter error method. In this chapter, we begin with the theoretical development and computational aspects for linear systems, followed by a discussion of various possible formulations. We then move on to the ultimate step of implementation for general nonlinear systems. As in Chapter 4, we highlight numerical intricacies, practical aspects and limitations, providing our recommendations in each case. We also take a critical look at the equivalence between the output-error and filter-error methods. We do not go into

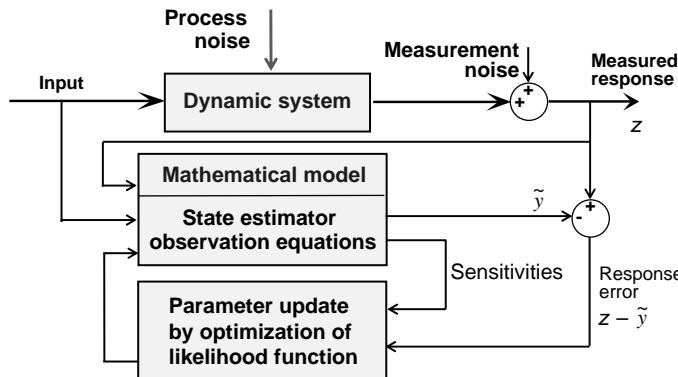


Fig. 5.1 Block schematic of filter error method.

the details of the statistical properties of the estimates,¹³ because the technique is based on the principle of maximum likelihood, whose desirable statistical properties have already been discussed in Chapter 4. We study the performance of the filter error method in two sample cases using estimation software capable of handling nonlinear systems. We also supply the software with the book to enable the reader to follow the discussion and to gain more insight into the algorithmic steps and numerical aspects. The data analyzed is also supplied as sample data to verify the results. Yet another aircraft example is presented to bring out the practical utility of these techniques.

II. Filter Error Method for Linear Systems

Although it has been said that our main focus in this book is on nonlinear systems, we consider in this chapter first the linear system representation, mainly to highlight the mathematical details of a fairly complex algorithm, and thereby appreciate the difficulties encountered in handling nonlinear systems with process noise. In such a case the dynamic system is assumed to be described by the following stochastic linear mathematical model:

$$\dot{x}(t) = Ax(t) + Bu(t) + Fw(t) + b_x, \quad x(t_0) = 0 \quad (5.1)$$

$$y(t) = Cx(t) + Du(t) + b_y \quad (5.2)$$

$$z(t_k) = y(t_k) + Gv(t_k), \quad k = 1, 2, \dots, N \quad (5.3)$$

where x is the $(n_x \times 1)$ state vector, u is the $(n_u \times 1)$ control input vector, and y is the $(n_y \times 1)$ observation (or also called model output) vector. The measurement vector z is sampled at N discrete time points. Matrices A , B , C , and D contain the unknown system parameters. Matrices F and G represent the process and measurement noise distribution matrices respectively. Based on the discussion of Chapter 3, Sec. V.B, we once again consider the bias terms b_x and b_y for the state and observation equations. Recall that these bias parameters accounts for nonzero initial conditions and biases in the measurements of the control

inputs and output variables automatically. Such a treatment of unknown initial conditions and measurement biases was found to be more efficient than treating them separately. Thus, in a most general case, it is necessary to estimate the parameter vector Θ consisting of the elements of the matrices A , B , C , and D , the bias parameters b_x and b_y , and elements of the noise distribution matrices F and G .

There are two important assumptions that we make for the process and the measurement noise. First, it is assumed that they affect the dynamic system linearly, which allows additive noise representation in Eqs. (5.1) and (5.3). Second, the process noise $w(t)$ is assumed to be characterized by a zero-mean Gaussian noise with an identity spectral density, and the measurement noise $v(t_k)$ is assumed to be characterized by a sequence of Gaussian random variables with zero mean and identity covariance. It is further assumed that the process noise and the measurement noise, $w(t)$ and $v(t_k)$, are uncorrelated and mutually independent. The noise distribution matrices F and G are in general unknown.

Following the maximum likelihood principle presented in Chapter 4, Sec. II, the cost function of Eq. (4.13) in terms of the residuals reduces in the present case to:

$$\begin{aligned} J(\Theta, R) = & \frac{1}{2} \sum_{k=1}^N [z(t_k) - \tilde{y}(t_k)]^T R^{-1} [z(t_k) - \tilde{y}(t_k)] \\ & + \frac{N}{2} \ln[\det(R)] + \frac{Nm}{2} \ln(2\pi) \end{aligned} \quad (5.4)$$

where \tilde{y} is the system output based on the predicted states incorporating the Kalman filter, and R is the steady state covariance matrix of the residuals (also called innovations). Parameter estimates for Θ and R are obtained by minimization of the likelihood function, Eq. (5.4). In order to evaluate the cost function, the first task is to compute \tilde{y} , which depends upon the estimated states.

Although we have made the simplifying assumption of process noise being additive, it still complicates the estimation procedure considerably, because the system is no longer deterministic. As a consequence it is not possible to simply integrate the state variables (as was done in the case of the output error method presented in Chapter 4). Rather, it is now necessary to incorporate some suitable state estimator. For linear systems, the Kalman filter provides an optimal state estimator;^{14,15} see Appendix F for the derivation of the Kalman filter algorithm.

The Kalman filter (state estimator) consists of two steps and for the linear model postulated in Eqs. (5.1) and (5.2) it is given by:

Prediction step

$$\tilde{x}(t_{k+1}) = \Phi \hat{x}(t_k) + \Psi B \bar{u}(t_k) + \Psi b_x \quad (5.5)$$

$$\tilde{y}(t_k) = C \tilde{x}(t_k) + D u(t_k) + b_y \quad (5.6)$$

Correction step

$$\hat{x}(t_k) = \tilde{x}(t_k) + K [z(t_k) - \tilde{y}(t_k)] \quad (5.7)$$

where \tilde{x} and \hat{x} denote the predicted and corrected state vectors, respectively, \bar{u} the average of the control inputs at the two discrete time points, \tilde{y} the predicted output variables, $[z(t_k) - \tilde{y}(t_k)]$ the residuals (innovations), $\Phi = e^{\Delta t}$ the state transition matrix, Ψ the integral of the state transition matrix (see Chapter 3, Sec. VIII.B), and K the Kalman filter gain matrix; Δt is the sampling interval. This filter is used only for the purpose of obtaining the true state variables from the noisy measurements, that is, for the state estimation only. In Eq. (5.5) we have omitted the initial conditions, because, as already pointed out earlier, they are zero. We will discuss later in this chapter about the exact sequence of computing Eqs. (5.5)–(5.7).

In many applications, particularly when the system under investigation is time-invariant and the deviations from the nominal trajectory are small, it is often adequate to use a steady-state filter for state estimation. This simplification results in a significant reduction of the computational burden. The Kalman gain matrix K is a function of the covariance matrix of the residuals R , of the covariance matrix of the state-prediction error P , and of the observation matrix C . It is given by:

$$K = PC^T R^{-1} \quad (5.8)$$

The covariance matrix of the innovations (residuals), R , is related to the measurement noise covariance matrix GG^T through the covariance matrix of the state-propagation error P and the observation matrix C .

$$R = GG^T + CPC^T \quad (5.9)$$

The covariance matrix of the state-prediction error P is obtained by solving the Riccati equation, which we will discuss in Sec. IV.A. The state-prediction error covariance matrix P accounts for the process noise and uncertainties in the initial conditions on the states, and thereby improves the state estimates and in turn the parameter estimates.

III. Process Noise Formulations

We now have all the details to discuss the implications of the filter error method and possible various techniques of parameter estimation in the presence of process and measurement noise. There are three different approaches that have evolved, based on the way the noise distribution matrices F and G or the related matrices R and K are estimated.^{6,16}

A. Natural Formulation

In any optimization problem, the very first approach that comes *naturally* to mind is to treat the parameters as elements of the unknown parameter vector, and then apply the best possible optimization method to minimize the cost function with respect to the unknown parameter vector.^{6,7} The natural formulation results from this basic idea in which the unknown covariances of both process and measurement noise are explicitly treated as unknowns to be estimated along with the other system parameters appearing in the system matrices A , B ,

C , and D . The complete vector of unknown parameters in this case is given by

$$\Theta = [(A_{ij}, i = 1, n_x; j = 1, n_x)^T \quad (B_{ij}, i = 1, n_x; j = 1, n_u)^T \\ (C_{ij}, i = 1, n_y; j = 1, n_x)^T \quad (D_{ij}, i = 1, n_y; j = 1, n_u)^T \\ (F_{ij}, i = 1, n_x; j = 1, n_x)^T \quad (G_{ij}, i = 1, n_y; j = 1, n_y)^T]^T \quad (5.10)$$

where i and j are the general indices, and each term on the right-hand side within brackets is a column vector of appropriate size. Although Eq. (5.5) is in terms of the state transition matrix Φ , we prefer to deal with the system matrix A using the relationship $\Phi = e^{A\Delta t}$. In general, the bias terms b_x and b_y appearing in Eqs. (5.5) and (5.6) will have to be included in Eq. (5.10), but they have been dropped from the general discussion in this section without loss of generality, because they are treated similarly to other parameters, and hence do not affect these investigations. Estimation of parameters appearing in Eq. (5.10) is a brute-force approach to which we can apply any one of the optimization methods. We have already discussed in some depth different methods in Chapter 4, Secs. VI–XIV, and found that the best choice is either one of the Gauss–Newton methods or the Levenberg–Marquardt method.

The two properties of the natural formulation that need some consideration are 1) convergence and 2) computational burden. The maximum likelihood cost function defined in Eq. (5.4) is derived in terms of the covariance matrix of the residuals. The noise distribution matrices appear only indirectly through the relation given in Eq. (5.9). Recalling our discussion in Chapter 4, Sec. V on the reasons for choosing the relaxation strategy, we will face the same difficulties here too, if not more so, because of the complex functional relationship between the various system parameters and F and G . This leads to serious convergence problems. The choice of the initial values for the elements of G is also critical; very low values affect the convergence whereas too large values may lead to physically unrealistic values, and in the worst case to divergence.⁶ The second factor which also does not favor this approach is the increased size of the unknown parameters and the need for the corresponding gradients, resulting in more computational burden. Although this was also a serious consideration in the 1960s and 1970s, with the availability of powerful computers and workstations, it is just a minor nuisance today. Based on these considerations, although the natural formulation may be theoretically a possible approach, it is not a practical one.

B. Innovation Formulation

This approach is very similar to that followed in the case of the output error method.^{2,6,9} It uses the relaxation strategy, in which the covariance matrix of the residuals is obtained by a general closed form solution derived in Chapter 4, Sec. V.B. This derivation is valid in the present case as well, because it is based on the residuals, and not on the way the residuals are computed, that is, whether they stem from linear or nonlinear systems, or from systems with process noise, or with both measurement and process noise. In the present

case, the maximum likelihood estimate of R , obtained by setting $\partial J/\partial R = 0$, is given by

$$R = \frac{1}{N} \sum_{k=1}^N [z(t_k) - \tilde{y}(t_k)][z(t_k) - \tilde{y}(t_k)]^T \quad (5.11)$$

This explicit solution for R overcomes the convergence problems encountered in the natural formulation and also reduces the number of unknown parameters to be estimated in the second step applying the Gauss–Newton or another method.

In the second step, the innovation formulation tries to further simplify the parameter optimization problem by considering just the three equations (5.5)–(5.7), and estimating along with the system parameters the elements of gain matrix K directly, instead of obtaining the same through Eq. (5.8). The parameter vector in this approach is given by

$$\Theta = [(A_{ij}, i = 1, n_x; j = 1, n_x)^T \quad (B_{ij}, i = 1, n_x; j = 1, n_u)^T \\ (C_{ij}, i = 1, n_y; j = 1, n_x)^T \quad (D_{ij}, i = 1, n_y; j = 1, n_u)^T \\ (K_{ij}, i = 1, n_x; j = 1, n_y)]^T \quad (5.12)$$

The basic idea behind this approach has been to eliminate the most complex part of the algorithm required otherwise to estimate the process noise distribution matrix F , which appears indirectly in K given by Eq. (5.8) through P . The covariance matrix of the state prediction error P is obtained by solving the Riccati equation. Furthermore, the optimization algorithm needs gradients of P as well (we will talk about these details in the next section). These two steps are the most complex part of the algorithm.

The above choice of estimation of K eliminates the most complex part of the process noise method, namely computing P and its gradients. Although from a computational point of view, the approach may appear appealing, it has two drawbacks. First, for better identifiability and accuracy of the estimates we usually have more observation variables than state variables, which means that the size of K (viz. $n_x \times n_y$) is larger than that of F (viz. $n_x \times n_x$). Thus, we may have to estimate a larger number of gain matrix parameters to account for the process noise. It may lead to identifiability problems. Moreover, the elements of K have no direct physical meaning, making interpretation and judgment about the estimates difficult. Secondly, the gain matrix K as well as the system parameters and the covariance matrix of residuals R are estimated independent of each other in two separate steps given by Eqs. (5.12) and (5.11), respectively. This may lead to incompatible R and K estimates. They may not satisfy the necessary condition of $KC \leq 1$ (discussion of this requirement is deferred to the next section). Constrained nonlinear optimization is necessary to account for this inequality. For larger dimensioned K this becomes very tedious.

C. Combined Natural cum Innovation Formulation

The combined formulation proposed by Maine and Iliff^{6,17} takes advantage of the best features of both the natural formulation (viz. estimation of the process noise matrix using the Gauss–Newton method) and the innovation formulation (viz. explicit estimation of the covariance matrix of residuals). Instead of the gain matrix K as in the innovation formulation, here the process noise distribution matrix F is treated as an unknown along with the other system parameters. The complete parameter vector is then given by

$$\Theta = [(A_{ij}, i = 1, n_x; j = 1, n_x)^T \quad (B_{ij}, i = 1, n_x; j = 1, n_u)^T \\ (C_{ij}, i = 1, n_y; j = 1, n_x)^T \quad (D_{ij}, i = 1, n_y; j = 1, n_u)^T \\ (F_{ij}, i = 1, n_x; j = 1, n_x)^T]^T \quad (5.13)$$

We apply the two-step relaxation strategy, estimating explicitly the covariance matrix of the residuals given by Eq. (5.11), and then estimate the above parameter vector of Eq. (5.13) applying the Gauss–Newton method. Estimation of R does not pose any difficulty. In the second step, we have a much smaller number of parameters to be estimated than in the innovation formulation (F is smaller dimensioned than K). The main advantage is that the elements of F have physical meaning. In this case too, the inequality constraint of $KC \leq 1$ will have to be met, requiring constrained optimization, but of a reduced dimension. The small penalty that we pay here is that we have to solve for the Riccati equation to obtain the covariance matrix of the state prediction error P . However, this formulation has been found to be more practical with regard to convergence, parameter estimates, and computational burden.

IV. Filter Error Algorithm

Based on the comparative discussion of the three possible formulations, we prefer and restrict our attention hereafter to the combined formulation. In such an algorithm, it is required to minimize the cost function defined in Eq. (5.4) subject to the system equations formulated in Eqs. (5.1)–(5.3). The important steps of such an algorithm are¹⁶

- 1) Specify some suitable starting values for the system parameters Θ , that is, Eq. (5.13) consisting of elements of the matrices A , B , C , and D , and also of the process noise distribution matrix F .
- 2) Specify suitable starting values for R or generate the estimate of R (we will talk about this in Sec. VI).
- 3) Do the state estimation applying the Kalman filter [requires computation of Gain matrix K , state transition matrix Φ , its integral Ψ , and of course the predicted observation variables $\tilde{y}(t_k)$, that is, Eqs. (5.5)–(5.8)].
- 4) Compute the covariance matrix of the residuals, given by Eq. (5.11).
- 5) Apply the Gauss–Newton method to update Θ , which requires computation of the necessary gradients and checking for the inequality constraint on KC .
- 6) Iterate on steps 3–5 until convergence.

As already stated, the Kalman filter provides the optimal state estimator for linear systems, the computational details of which are given by Eqs. (5.5)–(5.9). The procedure to compute the state transition matrix Φ and its integral Ψ is given by Eqs. (3.38) and (3.39), in Chapter 3, Sec. VIII.B. It now remains to study procedures to compute the covariance matrix of the state prediction error P and to compute the response gradients and those of P required in the Gauss–Newton algorithm.

A. Solution to Riccati Equation

The covariance matrix of the state prediction error, P , is obtained by solving a Riccati equation. We use here a steady-state form of the Riccati equation. Furthermore, the continuous-time Riccati equation, which can be solved efficiently, is preferred over the discrete-time equation.¹⁸ The first-order approximation of this equation is obtained as

$$AP + PA^T - \frac{1}{\Delta t} PC^T R^{-1} CP + FF^T = 0 \quad (5.14)$$

Equation (5.14) is solved using the well-known Potter's method,¹⁹ based on the Eigenvector decomposition. We present here just the essential details of the method to understand the necessary computational steps to solve Eq. (5.14) for P . We start by constructing the Hamiltonian matrix H , defined as

$$H = \begin{bmatrix} A & -FF^T \\ -\frac{1}{\Delta t} C^T R^{-1} C & -A^T \end{bmatrix} \quad (5.15)$$

H is a real, general nonsymmetric matrix of the dimensions $2n_x \times 2n_x$, where n_x is the number of state variables. Now compute the eigenvalues and eigenvectors of H , which may be complex quantities.

Let the eigenvalues and the matrix of eigenvectors of H be denoted by Λ and E , respectively. The eigenvector matrix, E , is now partitioned into two equal parts such that the left half contains the eigenvectors corresponding to unstable eigenvalues, that is, corresponding to eigenvalues with positive real part. The controllability by the process noise and observability ensure that exactly half the number of eigenvalues will have positive real parts and that the eigenvectors can be determined uniquely. The rearranged matrix of eigenvectors is further decomposed into $(n_x \times n_x)$ size matrices as

$$E = \begin{bmatrix} E_{11} & | & E_{12} \\ E_{21} & | & E_{22} \end{bmatrix} \quad (5.16)$$

The solution to the steady-state Riccati equation is then given by

$$P = -E_{11}E_{21}^{-1} \quad (5.17)$$

It can be proved that the solution for P given by Eq. (5.17), which is an $(n_x \times n_x)$ matrix, is positive definite, because, as said above, the eigenvectors are uniquely

determined.¹⁹ The covariance matrix of the state prediction error, P , is a steady-state matrix and as such does not depend on time.

Potter's method assumes the partitioned eigenvector matrices to be nonsingular. If the eigenvalues are nearly equal, the method may show poor numerical stability, because although the partitioned eigenvector matrices are still nonsingular, they remain ill-conditioned. In our specific case, since the gradients of P are obtained through numerical difference approximations (to be addressed in Sec. V) using small perturbations in the system matrices, it may lead to ill-conditioning. Small perturbations in the state matrix elements can lead to drastic changes in the partitioned eigenvector matrix, which in turn causes poor numerical stability. In such cases gradients may not be reliable or accurate. An improved method which is more stable and not overly sensitive to small changes has been demonstrated by Holley and Wei.²⁰ It alleviates the problems due to an ill-conditioned partitioned eigenvector matrices. However, in several cases that we have analyzed, the classical Potter's method as elaborated in Eqs. (5.14)–(5.17) proved adequate.

B. Parameter Update

Having computed the predicted observation variables applying the Kalman filter, the computation of the covariance matrix of the residuals is straightforward; it is just a simple summation over the N data points given by Eq. (5.11). In the next step we apply the Gauss–Newton method to update the parameter vector Θ of Eq. (5.13), which yields

$$\Theta_{i+1} = \Theta_i + \Delta\Theta, \quad \text{and} \quad \mathcal{F}\Delta\Theta = -\mathcal{G} \quad (5.18)$$

where i is the iteration index and \mathcal{F} and \mathcal{G} are given by

$$\mathcal{F} = \sum_{k=1}^N \left[\frac{\partial \tilde{y}(t_k)}{\partial \Theta} \right]^T R^{-1} \left[\frac{\partial \tilde{y}(t_k)}{\partial \Theta} \right] \quad (5.19)$$

$$\mathcal{G} = - \sum_{k=1}^N \left[\frac{\partial \tilde{y}(t_k)}{\partial \Theta} \right]^T R^{-1} [z(t_k) - \tilde{y}(t_k)] \quad (5.20)$$

It is exactly the same equation that we derived for the output error method in Eqs. (4.29) and (4.30) in Chapter 4, except that y is replaced through \tilde{y} . Equation (5.19) is an approximation to the second gradient of the cost function $\partial^2 J / \partial \Theta^2$; we have already discussed in Chapter 4, Sec. VI the reasons for and suitability of this simplification. As in the case of the output error method, the Gauss–Newton method may show local intermediate divergence. In such cases we apply the heuristic procedure of parameter-halving that we discussed in Chapter 4, Sec. X.A.

From Eq. (5.11) we know how to compute R . We also know the procedure, given by Eqs. (5.5)–(5.8) and (5.14)–(5.17), to compute the predicted responses. To compute the parameter update $\Delta\Theta$ applying Eqs. (5.18)–(5.20), we now need a procedure to compute the sensitivity matrix, that is, the response gradients,

defined as

$$\left(\frac{\partial \tilde{y}}{\partial \Theta} \right)_{ij} = \frac{\partial \tilde{y}_i}{\partial \Theta_j} \quad (5.21)$$

In a classical way the sensitivity matrix $\partial \tilde{y} / \partial \Theta$ is obtained by solving the sensitivity equations, which are derived by partial differentiation of the system equations with respect to each element of the unknown system parameters. Partial differentiation of Eq. (5.5) leads to

$$\begin{aligned} \frac{\partial \tilde{x}(t_{k+1})}{\partial \Theta} &= \Phi \frac{\partial \hat{x}(t_k)}{\partial \Theta} + \frac{\partial \Phi}{\partial \Theta} \hat{x}(t_k) + \Psi \frac{\partial B}{\partial \Theta} \bar{u}(t_k) + \frac{\partial \Psi}{\partial \Theta} B \bar{u}(t_k) \\ &\quad + \Psi \frac{\partial b_x}{\partial \Theta} + \frac{\partial \Psi}{\partial \Theta} b_x \end{aligned} \quad (5.22)$$

Equation (5.22) gives the exact gradient of Eq. (5.5), and requires $\partial \Phi / \partial \Theta$ and $\partial \Psi / \partial \Theta$, the gradients of the state transition matrix Φ and of its integral Ψ . The computational burden to obtain $\partial \Phi / \partial \Theta$ and $\partial \Psi / \partial \Theta$ is significant. Hence, we approximate the gradient computation in Eq. (5.22) through⁶

$$\begin{aligned} \frac{\partial \tilde{x}(t_{k+1})}{\partial \Theta} &\approx \Phi \frac{\partial \hat{x}(t_k)}{\partial \Theta} + \Psi \frac{\partial A}{\partial \Theta} \bar{x}(t_k) \\ &\quad + \Psi \frac{\partial B}{\partial \Theta} \bar{u}(t_k) + \Psi \frac{\partial b_x}{\partial \Theta}, \quad \frac{\partial \tilde{x}(1)}{\partial \Theta} = 0 \end{aligned} \quad (5.23)$$

where, as already pointed out in Chapter 3, Sec. VIII.A, \bar{u} is the average of the inputs at t_{k+1} and t_k , and likewise $\bar{x} = 1/2[\tilde{x}(t_{k+1}) + \hat{x}(t_k)]$. Equation (5.23) is accurate to the order of $(\Delta t)^2$ and adequate since flight data is usually sampled at a high rate, that is, with small sampling time. This simplification leads to sensitivity equations which have the same state transition matrix Φ and the same integral of the state transition matrix, Ψ , both of which are evaluated anyway during the computation of the system responses from Eqs. (5.5)–(5.7).

Partial differentiation of Eqs. (5.6) and (5.7) yields

$$\frac{\partial \tilde{y}(t_k)}{\partial \Theta} = C \frac{\partial \tilde{x}(t_k)}{\partial \Theta} + \frac{\partial C}{\partial \Theta} \tilde{x}(t_k) + \frac{\partial D}{\partial \Theta} u(t_k) + \frac{\partial b_y}{\partial \Theta} \quad (5.24)$$

$$\frac{\partial \hat{x}(t_k)}{\partial \Theta} = \frac{\partial \tilde{x}(t_k)}{\partial \Theta} - K \frac{\partial \tilde{y}(t_k)}{\partial \Theta} + \frac{\partial K}{\partial \Theta} [z(t_k) - \tilde{y}(t_k)] \quad (5.25)$$

Equations (5.23)–(5.25) form the set of sensitivity equations. All the terms appearing in Eqs. (5.23)–(5.25) have been defined, except for $\partial K / \partial \Theta$, the gradient of the Kalman gain matrix.

Partial differentiation of Eq. (5.8) with respect to Θ yields

$$\frac{\partial K}{\partial \Theta} = \frac{\partial P}{\partial \Theta} C^T R^{-1} + P \frac{\partial C^T}{\partial \Theta} R^{-1} \quad (5.26)$$

Note that $\partial K / \partial \Theta$ and $\partial P / \partial \Theta$ and some other terms in the following are three-dimensional arrays. During a particular step of the Gauss–Newton method, the covariance matrix R is fixed and hence only partial differentiations of P and C appear on the right-hand side of Eq. (5.26). Computation of $\partial K / \partial \Theta$ using Eq. (5.26) requires the gradient of P , the covariance matrix of the state prediction error. It is obtained by partial differentiation of the Riccati equation (5.14):

$$\begin{aligned} A \frac{\partial P}{\partial \Theta} + \frac{\partial A}{\partial \Theta} P + P \frac{\partial A^T}{\partial \Theta} + \frac{\partial P}{\partial \Theta} A^T - \frac{1}{\Delta t} \frac{\partial P}{\partial \Theta} C^T R^{-1} C P \\ - \frac{1}{\Delta t} P \frac{\partial C^T}{\partial \Theta} R^{-1} C P - \frac{1}{\Delta t} P C^T R^{-1} \frac{\partial C}{\partial \Theta} P \\ - \frac{1}{\Delta t} P C^T R^{-1} C \frac{\partial P}{\partial \Theta} + F \frac{\partial F^T}{\partial \Theta} + \frac{\partial F}{\partial \Theta} F^T = 0 \end{aligned} \quad (5.27)$$

Combining together the (1st and 8th), (4th and 5th), (2nd, 7th and 10th) and (3rd, 6th and 9th) terms on the left-hand side of Eq. (5.27), and after simple manipulations using the matrix algebra results $(A + B)^T = A^T + B^T$ and $(AB)^T = B^T A^T$, as well as P and R being symmetrical, $P^T = P$, $R^T = R$, and $(R^{-1})^T = R^{-1}$, it can be shown that Eq. (5.27) can be rewritten in the form

$$\bar{A} \frac{\partial P}{\partial \Theta} + \frac{\partial P}{\partial \Theta} \bar{A}^T = \bar{C} + \bar{C}^T \quad (5.28)$$

where

$$\bar{A} = A - \frac{1}{\Delta t} P C^T R^{-1} C = A - \frac{1}{\Delta t} K C \quad (5.29)$$

and

$$\bar{C} = -\frac{\partial A}{\partial \Theta} P + \frac{1}{\Delta t} P C^T R^{-1} \frac{\partial C}{\partial \Theta} P - \frac{\partial F}{\partial \Theta} F^T \quad (5.30)$$

For each element of Θ , we obtain one equation in the above form of Eq. (5.28). For the n_q unknown parameters, this leads to a set of Lyapunov equations of the form $A X + X A^T = B$. There are several general procedures to solve this set of equations. However, since the matrix \bar{A} remains the same for the complete set, they are solved more efficiently using the transformation:

$$\bar{A}' = T^{-1} \bar{A} T \quad (5.31)$$

and

$$(\bar{C} + \bar{C}^T)' = T^{-1}(\bar{C} + \bar{C}^T)T^{*-1} \quad (5.32)$$

where T is the matrix of eigenvectors of \bar{A} . This similarity transformation makes \bar{A}' diagonal and reduces the Lyapunov equation (5.28) to

$$\bar{A}' \left(\frac{\partial P}{\partial \Theta} \right)' + \left(\frac{\partial P}{\partial \Theta} \right)' \bar{A}'^* = (\bar{C} + \bar{C}^T)' \quad (5.33)$$

Since \bar{A}' is diagonal, Eq. (5.33) can be solved for $(\partial P / \partial \Theta)'$ easily. In a general case, the solution to $AX + XA^T = B$ for a diagonal A is given by $X_{ij} = B_{ij}/(A_{ii} + A_{jj})$.

The gradient of the covariance matrix P is then obtained through back transformation

$$\frac{\partial P}{\partial \Theta} = T \left(\frac{\partial P}{\partial \Theta} \right)' T^* \quad (5.34)$$

As in the case of the Hamiltonian matrix, Eq. (5.15), the matrix \bar{A} is a general real matrix, and as such the eigenvectors may be complex quantities.

Equations (5.18)–(5.20) for the parameter update using the Gauss–Newton method yields the unconstrained minimum of the cost function, in which all the parameters are treated independent of each other. In the mixed formulation discussed here, direct estimation of R instead of explicit F and G matrices eliminates the convergence difficulties. However, since R is a function of F and G , it is necessary to ensure that the measurement noise covariance matrix GG^T , as will be obtained indirectly from Eq. (5.9), is positive semi-definite (to be physical meaningful). To meet this condition, all the eigenvalues of KC must be less than or equal to 1.⁶ However, KC tends to be diagonal dominant and hence the eigenvalue constraint is approximated by constraining the diagonal elements of KC to be less than 1. Thus, it is now necessary to verify whether the above unconstrained solution satisfies the constraints of the diagonal elements of KC to be less than unity. These nonlinear inequality constraints lead to a nonlinear programming problem. The quadratic programming method, which requires a linear approximation of the constraints, is used to solve this complex optimization problem.²¹ The linear approximation to the constraints is given by

$$(KC)_{ii} + \frac{\partial(KC)_{ii}}{\partial \Theta} \Delta \Theta \leq 1 \quad (5.35)$$

where the gradient of KC is given by:

$$\frac{\partial(KC)}{\partial \Theta} = \frac{\partial K}{\partial \Theta} C + K \frac{\partial C}{\partial \Theta} \quad (5.36)$$

Let ℓ be the number of constraints violated by the unconstrained solution given by Eq. (5.18). In this case an auxiliary vector S and a matrix M are formed such

that

$$S_i = 1 - (KC)_{ii} \quad (5.37)$$

and

$$M(i,j) = \frac{\partial(KC)_{ii}}{\partial\Theta}, \quad i = 1, \ell; j = 1, n_q \quad (5.38)$$

where n_q is the total number of unknown parameters.

The constrained solution is then given by

$$\Delta\Theta = \Delta\Theta - \left(\frac{\partial^2 J}{\partial\Theta^2} \right)^{-1} M^T \left\{ M \left(\frac{\partial^2 J}{\partial\Theta^2} \right)^{-1} M^T \right\}^{-1} S \quad (5.39)$$

To solve a general nonlinear programming problem using quadratic programming, it is usually required to iterate on Eqs. (5.35)–(5.39) until all the inequality constraints are simultaneously satisfied. In the present case, however, such iterations are not necessary, because the final solution converges in the iterative unconstrained optimization loop.

The maximum likelihood estimate of R is obtained by equating the gradient of the cost function, Eq. (5.4), with respect to R to zero. Since innovations are functions of the Kalman gain matrix K , the exact equation for R is complex and computationally tedious. An asymptotic approximation to R is given by

$$R = \frac{1}{N} \sum_{k=1}^N [z(t_k) - \tilde{y}(t_k)][z(t_k) - \tilde{y}(t_k)]^T \quad (5.40)$$

which is the same as Eq. (5.11). Computation of R using Eq. (5.40) corresponds to the innovation formulation. In the case of maximum likelihood estimation accounting for measurement noise only, the gain matrix K is zero. The residuals are independent of R , and Eq. (5.40) itself provides the maximum likelihood estimate of the noise covariance.

The two steps to compute R explicitly and $\Delta\Theta$ applying the Gauss–Newton method, Eqs. (5.40) and (5.18), are carried out independently. They do not account for the influence of each on the estimates of the other. This may yield strongly correlated matrices F and R , which affects the convergence. To account for this correlation, Ref. 6 suggests the following heuristic procedure to compensate the estimates of the F -matrix, whenever R is revised:

$$F_{il}^{\text{new}} = F_{il}^{\text{old}} \left(\frac{\sum_{j=1}^{n_y} C_{ij}^2 r_j^{\text{old}} \sqrt{r_j^{\text{old}}/r_j^{\text{new}}}}{\sum_{j=1}^{n_y} C_{ij}^2 r_j^{\text{old}}} \right) \quad (5.41)$$

where r_j is the j th diagonal element of R^{-1} , C_{ij} the (i, j) th element of C , and the superscripts “old” and “new” denote the previous and revised estimates

respectively. Equation (5.41) implies that the elements ($l = 1, n_x$) of the i th row of F are multiplied by the factor within the bracket on the right-hand side. The correction given by Eq. (5.41) is applied only to elements of F which are being estimated.

In practice, it could happen that the correction of F using the new values of R^{-1} according to Eq. (5.41) may lead to slight local divergence. The reasons are twofold: 1) the estimates of F and R are correlated, and 2) the heuristic procedure of Eq. (5.41) is a first-order approximation. In such a case of local divergence resulting from correction of F , we once again apply to the corrected F values the parameter-halving procedure that we discussed in Chapter 4, Sec. X.A. Since the local divergence here, if any, is due to correlation between R and F , very few halving steps are generally enough; we allow just two halving steps. In the eventuality that two halving steps of F -corrections are not sufficient to yield a smaller cost function value, then it is recommended to proceed to the next full step of Gauss–Newton without performing the F -correction during the particular iteration.

From the above development it is now obvious that the formulation adapted makes use of the innovation formulation as a first step to estimate the noise covariance of the residuals. In the second step, instead of the gain matrix K , the state noise distribution matrix F is estimated along with the other system parameters applying the Gauss–Newton method. This is similar to the procedure of the natural formulation. The main difficulties in the estimation of the gain matrix encountered in the innovation formulation are thus circumvented. Although F is an $(n_q \times n_q)$ matrix, it is a common practice to treat it as a diagonal matrix. This simplification helps not only to reduce the additional computational burden, but also to avoid any identification problem.

Although the whole procedure is fairly complex, the detailed step-by-step treatment in this section provides us with an insight into the computational complexity of the algorithm based on an analytical approach. Efficient matrix operations provide capabilities to handle conveniently different models postulated in the form of Eqs. (5.1) and (5.2). It is probably a little more accurate compared with the numerical approach that we will follow in the next section for the nonlinear system, but it is limited to linear systems only.

If it is necessary to account for measurement noise only, then in such a case of $F = 0$, the Kalman filter simply integrates the state equations. The predicted and corrected states, \tilde{x} and \hat{x} , are identical. The gain matrix K is zero (there is a subtle difference in such cases, but we will talk more about it in Sec. IX). The corresponding equations, namely Eqs. (5.14)–(5.17) and (5.26)–(5.39), are redundant. Thus, neglecting the process noise significantly simplifies the estimation algorithm. Such an algorithm accounting for measurement noise only, referred to as the output error method discussed in the previous chapter, is a special case of the filter error method.

V. Filter Error Method for Nonlinear Systems

Having discussed the filter error method for linear systems, we now turn our attention in this section to nonlinear systems. In a general case, the mathematical

model in state space is given by the stochastic equations:

$$\dot{x}(t) = f[x(t), u(t), \beta] + Fw(t), \quad x(t_0) = x_0 \quad (5.42)$$

$$y(t) = g[x(t), u(t), \beta] \quad (5.43)$$

$$z(t_k) = y(t_k) + Gv(t_k) \quad (5.44)$$

where f and g are the n_x and n_y dimensional general nonlinear real valued vector functions. They are assumed to have sufficient differentiability to be able to invoke Taylor series expansion. The rest of the variables have been defined in Sec. II. As in the case of linear systems, in the present case too, the noise is assumed to be additive, and the noise distribution matrices F and G are considered to be time-invariant as well as independent of each other.

The basic difference compared with the linear systems is the way we now compute the states and the gradients. The difficulties of extending the filter error algorithm to nonlinear systems are twofold: 1) to efficiently implement the algorithms that provide flexibility to handle conveniently different model structures without software modifications, and 2) to derive a suitable filter for nonlinear state estimation.

From Chapter 4, Sec. VIII we recall that the algebraic differentiation of nonlinear model postulates may be difficult and, more importantly, any changes in the model structure entail re-derivation of the sensitivity equations and subsequent programming changes. The numerical approach of finite difference approximation, which was discussed in the case of the output error method and also found efficient, can be extended to the filter error method to compute the response gradients. Since optimal filters for nonlinear systems are practically unrealizable except for some simple trivial cases, an extended Kalman filter (EKF) based on a first-order approximation of the state and measurement equations can be used for nonlinear filtering. A short treatment on EKF as a state estimator is provided in Appendix F.

In many cases, for example, when system models are time-invariant, contain weak to moderate nonlinearities, and the deviations from the nominal trajectory are not large, it may be adequate to use a steady-state filter for state estimation. This simplification results in a significant reduction of the computational burden. However, if the system response is dominated by nonlinearities or when the deviations from the nominal trajectory are large, it may be necessary to incorporate a time-varying filter. Both these possibilities are discussed here.

A. Steady-state Filter

The mathematical details of the maximum likelihood estimation using a steady-state filter as a state estimator are similar to the one we discussed in the foregoing section. The cost function given by Eq. (5.4) is to be minimized subject to the system equations formulated in Eqs. (5.42)–(5.44). As in the case of linear systems, we follow the combined formulation of Sec. III.C and apply the two-step procedure to estimate R using Eq. (5.11) and update the parameters applying the Gauss–Newton method as in Eqs. (5.18)–(5.20). The important steps of such an algorithm can be summarized as follows:^{10–12}

- 1) Specify some suitable starting values for the parameter vector Θ consisting of parameters appearing in the system functions f and g , initial conditions x_0 and elements of the process noise distribution matrix F .
- 2) Specify suitable starting values for R or generate the estimate of R (we will discuss this in Sec. VI).
- 3) Do the state estimation applying the extended Kalman filter [this requires computation of the gain matrix K , numerical integration of the state equations, and of course computation of the predicted observation variables $\tilde{y}(t_k)$].
- 4) Compute the covariance matrix of the residuals, Eq. (5.11).
- 5) Apply the Gauss–Newton method, Eqs. (5.18)–(5.20), to update Θ , which requires computation of the necessary gradients and checking for the inequality constraint on KC .
- 6) Iterate on steps 3–5 until convergence.

As already pointed out, since optimal filters for nonlinear systems are practically unrealizable, an extended Kalman filter is used for nonlinear state estimation. To retain the system nonlinearities in the state estimation as far as possible, a mixed version incorporates a prediction step with the nonlinear model and a correction step based on a first-order approximation of the state and measurement equations. Such a two-step nonlinear constant-gain filter can be represented as

Prediction step

$$\tilde{x}(t_{k+1}) = \hat{x}(t_k) + \int_{t_k}^{t_{k+1}} f[x(t), \bar{u}(t_k), \beta] dt, \quad \hat{x}(t_0) = x_0 \quad (5.45)$$

$$\tilde{y}(t_k) = g[\tilde{x}(t_k), u(t_k), \beta] \quad (5.46)$$

Correction step

$$\hat{x}(t_k) = \tilde{x}(t_k) + K[z(t_k) - \tilde{y}(t_k)] \quad (5.47)$$

where \bar{u} could be the average of the control inputs at the two discrete time points or, as discussed in Chapter 3, Sec. VIII.A, appropriately interpolated values between $u(t_k)$ and $u(t_{k+1})$ for the intermediate steps of the numerical integration formula.

The steady-state gain matrix K is a function of the steady-state R , the steady-state P , and C of the linearized system. It is given by

$$K = PC^T R^{-1} \quad (5.48)$$

where

$$C = \left[\frac{\partial g[x(t), u(t), \beta]}{\partial x} \right]_{t=t_0} \quad (5.49)$$

A steady-state form of the Riccati equation is used to obtain P . We adopt here exactly the same procedure consisting of Eqs. (5.14)–(5.17) that has

already been described in Sec. IV.A, except that we use linearized system matrices A and C .

For a small perturbation δx_j in each of the n_q state variables, the elements of system matrices A and C are approximated using the central-difference formulas as

$$A_{ij} \approx \left. \frac{f_i[x + \delta x_j e^j, u, \beta] - f_i[x - \delta x_j e^j, u, \beta]}{2\delta x_j} \right|_{x=x_0} \quad (5.50)$$

$$C_{ij} \approx \left. \frac{g_i[x + \delta x_j e^j, u, \beta] - g_i[x - \delta x_j e^j, u, \beta]}{2\delta x_j} \right|_{x=x_0} \quad (5.51)$$

where e^j is a column vector with one in the j th row and zeros elsewhere. In the case of nonlinear systems, x_0 are updated iteratively in the parameter-estimation loop. Linearization of the system equations at each iteration about this convenient point x_0 can be used to compute the constant-gain matrix. We would like to highlight once again that the actual postulated nonlinear system equations, functions f and g , are used to extrapolate the state estimates \tilde{x} by numerical integration and for computation of \tilde{y} , the predicted system responses. However, the state variable correction in Eq. (5.47), which depends on K , is based on a first-order approximation.

In contrast to the filter error method for linear systems, in the present case all of the required response gradients are approximated using the finite-difference approximation and not by solving the analytically derived sensitivity equations. For a small perturbation $\delta \Theta_j$ in each of the n_q unknown variables of the parameter vector Θ , the perturbed response variable y_{pi} for each of the unperturbed variables y_i is computed. The corresponding sensitivity coefficient is then approximated by

$$\left[\frac{\partial \tilde{y}(t_k)}{\partial \Theta} \right]_{ij} = \frac{\tilde{y}_{pi}(t_k) - \tilde{y}_i(t_k)}{\delta \Theta_j} \quad (5.52)$$

The perturbed response variables \tilde{y}_p are obtained from a set of perturbed system equations, similar to Eqs. (5.42) and (5.43), obtained by replacing Θ with $\Theta + \delta \Theta_j e^j$. For each element of Θ , the state and observation variables can be computed using a two-step nonlinear filter:

Prediction step

$$\tilde{x}_{p-j}(t_{k+1}) = \hat{x}_{p-j}(t_k) + \int_{t_k}^{t_{k+1}} f[x_{p-j}(t), \bar{u}(t_k), \beta + \delta \beta_j e^j] dt \quad (5.53)$$

$$\tilde{y}_{p-j}(t_k) = g[\tilde{x}_{p-j}(t_k), u(t_k), \beta + \delta \beta_j e^j] \quad (5.54)$$

Correction step

$$\hat{x}_{p-j}(t_k) = \tilde{x}_{p-j}(t_k) + K_{p-j}[z(t_k) - \tilde{y}_{p-j}(t_k)] \quad (5.55)$$

where the subscript p denotes the perturbed variables and j the index for the parameter being varied.

Computation of the predicted perturbed-state variables \tilde{x}_p from Eq. (5.53) by numerical integration, and of perturbed output variables \tilde{y}_p from Eq. (5.54) is straightforward. However, computation of the corrected perturbed-state variables in Eq. (5.55) requires the perturbed gain matrix K_{p-j} . It is given by

$$K_{p-j} = P_{p-j} C_{p-j}^T R^{-1} \quad (5.56)$$

where the covariance matrix of the perturbed state-prediction error P_{p-j} for the $\Theta + \delta\Theta_j e^j$ perturbed parameter vector is obtained by solving the Riccati equation (5.14) with perturbed system matrices A_{p-j} and C_{p-j} computed for the corresponding perturbation. These linearized perturbed matrices are once again approximated by central-difference formulas given in Eqs. (5.50) and (5.51) evaluated at $\Theta + \delta\Theta_j e^j$.

The extension of finite-difference approximation to the filter algorithm thus involves not only the numerical integration of the perturbed state equations, but also additionally the computation of the perturbed gain matrices for each element of the unknown parameter vector Θ . The perturbed output variables \tilde{y}_p computed from Eq. (5.54) using perturbed states \tilde{x}_p and perturbed gain matrix K_p will automatically account for the respective gradients. Thus, in comparison to the previous formulation for linear systems, the solution of a set of Lyapunov equations for the gradients of P and, from that the computation of gradients of K , has been replaced in the current approach through the solution of the perturbed Riccati equations. The computational burdens in the two approaches are of similar orders of magnitude, except that the finite-difference approximations lead to a flexible algorithm, besides being applicable to general nonlinear models; the linear model, being a simplified case, requires no special treatment.

In Sec. IV it was shown that, to ensure physically meaningful values for indirectly obtained G , the diagonal elements of the matrix KC must be constrained to be less than unity. It was also shown that minimization of the cost function, Eq. (5.4), subject to these nonlinear inequality constraints, leads to a nonlinear programming problem, which is solved by a quadratic programming method. In the present, a similar approach has to be adopted; with the further simplification that the elements of KC are constrained to be less than unity, where the observation matrix C is obtained by a first-order system approximation. Since the algorithmic details, covered in Eqs. (5.35)–(5.39), remain the same, they are not repeated here. It is also required to follow the procedure of Eq. (5.41) to correct the estimates of F whenever R is updated. If we encounter local divergence during the F -correction, we also follow the same philosophy of parameter-halving as elaborated in Sec. IV.

In order to highlight the differences between the two approaches for linear and nonlinear systems, respectively, the computational steps in the two cases are summarized in Table 5.1.

Table 5.1 Summary of filter-error-method computations for linear and nonlinear systems

$\dot{x}(t) = Ax(t) + Bu(t)$	$\dot{x}(t) = f[x(t), u(t), \beta]$
Integrate state equations by state transition matrix and its integral: $\tilde{x}(t_{k+1}) = \Phi\hat{x}(t_k) + \Psi B\bar{u}(t_k)$	Numerical integration of state equations, typically Runge–Kutta second or fourth order
Compute observation variables: $\tilde{y} = C\tilde{x}(t_k) + Du(t_k) + b_y$	Compute observation variables: $\tilde{y} = g[\tilde{x}(t_k), u(t_k), \beta] + b_y$
Compute residuals: $[z(t_k) - \tilde{y}(t_k)]$	
Compute maximum likelihood estimate of R , Eq. (5.11)	
Solve steady-state Riccati equation, Eq. (5.14), for P	
Compute gain matrix: $K = PC^T R^{-1}$	
Compute corrected states: $\hat{x}(t_k) = \tilde{x}(t_k) + K[z(t_k) - \tilde{y}(t_k)]$	
Compute gradients:	Numerical approximation of response gradients from perturbed system equations:
$\frac{\partial \tilde{x}(t_{k+1})}{\partial \Theta} \approx \Phi \frac{\partial \hat{x}(t_k)}{\partial \Theta} + \Psi \frac{\partial A}{\partial \Theta} \bar{x}(t_k) + \Psi \frac{\partial B}{\partial \Theta} \bar{u}(t_k) + \Psi \frac{\partial b_x}{\partial \Theta}$	$\dot{\tilde{x}}_p(t) = f[\hat{x}_p(t), u(t), \beta + \delta\beta]$
$\frac{\partial \tilde{y}(t_k)}{\partial \Theta} = C \frac{\partial \tilde{x}(t_k)}{\partial \Theta} + \frac{\partial C}{\partial \Theta} \tilde{x}(t_k) + \frac{\partial D}{\partial \Theta} u(t_k) + \frac{\partial b_y}{\partial \Theta}$	$\tilde{y}_p = g[\tilde{x}_p(t_k), u(t_k), \beta + \delta\beta]$
$\frac{\partial \hat{x}(t_k)}{\partial \Theta} = \frac{\partial \tilde{x}(t_k)}{\partial \Theta} - K \frac{\partial \tilde{y}(t_k)}{\partial \Theta} + \frac{\partial K}{\partial \Theta} [z(t_k) - \tilde{y}(t_k)]$	$\hat{x}_p(k) = \tilde{x}_p(k) + K_p[z(k) - \tilde{y}_p(k)]$
The gradients of the gain matrix K ,	$\left(\frac{\partial \tilde{y}}{\partial \Theta} \right)_{ij} \approx \frac{\tilde{y}_i(\Theta + \delta\Theta_j e^j) - \tilde{y}_i(\Theta)}{\delta\Theta_j}$
$\frac{\partial K}{\partial \Theta} = \frac{\partial P}{\partial \Theta} C^T R^{-1} + P \left(\frac{\partial C}{\partial \Theta} \right)^T R^{-1}$	for each parameter perturbation, i.e., extended Kalman filter for perturbed state equations: compute perturbed \tilde{x}_p , \tilde{y}_p and \hat{x}_p . This requires K_p , which is obtained from P_p
This requires $\partial P / \partial \Theta$ which are obtained by solving a set of Lyapunov equations [see Eqs. (5.27)–(5.34)]	<i>Note:</i> since gradients of \tilde{x} and \hat{x} are not explicitly used (as in the case of linear systems), the gradients of K are not required. This implies that gradients of P , and hence the Lyapunov equations, are not explicitly required. These effects are indirectly taken care of through perturbed gain matrices K_p , and calls for solving <i>perturbed Riccati equations</i> for each parameter.
Compute parameter update $\Delta\Theta$ by the Gauss–Newton method; Eqs. (5.18)–(5.20)	
Check for the constraints $KC < 1$;	
update $\Delta\Theta$ if necessary; Eqs. (5.35)–(5.39)	
Compensate F -estimates for new R , Eq. (5.41)	
Iterate until convergence	

B. Time-varying Filter

In a few cases, for example when system parameters are time-varying or the response is dominated by nonlinearities or when the deviations from the nominal trajectory are large, it may become necessary to incorporate a time-varying filter for state estimation. The formulation runs parallel to the one we just discussed for the steady-state filter. It can be shown that in the present case the maximum likelihood cost function is given by^{10,12,13}

$$\begin{aligned} J(\Theta, R) = & \frac{1}{2} \sum_{k=1}^N [z(t_k) - \tilde{y}(t_k)]^T R^{-1}(t_k) [z(t_k) - \tilde{y}(t_k)] \\ & + \frac{1}{2} \sum_{k=1}^N \ln[\det(R(t_k))] \end{aligned} \quad (5.57)$$

Starting from suitably specified initial values of Θ , the new updated estimates are obtained applying the Gauss–Newton algorithm:

$$\Theta_{i+1} = \Theta_i + \Delta\Theta, \quad \text{and} \quad \mathcal{F}\Delta\Theta = -\mathcal{G} \quad (5.58)$$

where i is the iteration index and \mathcal{F} and \mathcal{G} are given by

$$\mathcal{F} = \sum_{k=1}^N \left[\frac{\partial \tilde{y}(t_k)}{\partial \Theta} \right]^T R^{-1}(t_k) \left[\frac{\partial \tilde{y}(t_k)}{\partial \Theta} \right] \quad (5.59)$$

$$\mathcal{G} = - \sum_{k=1}^N \left[\frac{\partial \tilde{y}(t_k)}{\partial \Theta} \right]^T R^{-1}(t_k) [z(t_k) - \tilde{y}(t_k)] \quad (5.60)$$

The iterative update equation (5.58) is exactly the same as Eq. (5.18). Equations (5.59) and (5.60) for the information matrix \mathcal{F} and gradient \mathcal{G} are very similar to Eqs. (5.19) and (5.20), except that we now use the time-varying residual covariance matrix $R(t_k)$ at each discrete time point, instead of the steady-state version R .

As in the case of the steady-state filter, we now discuss techniques to propagate the states and compute the output variables. For nonlinear state estimation, we once again use the EKF, based on the nonlinear state equations for prediction and the correction using a first-order approximation of the state and measurement equations. For the system defined in Eqs. (5.42)–(5.44), the EKF consisting of an extrapolation and an update step can be summarized as:

Extrapolation

$$\tilde{x}(t_{k+1}) = \hat{x}(t_k) + \int_{t_k}^{t_{k+1}} f[x(t), \bar{u}(t_k), \beta] dt \quad (5.61)$$

$$\tilde{y}(t_k) = g[\tilde{x}(t_k), u(t_k), \beta] \quad (5.62)$$

$$\tilde{P}(t_{k+1}) \approx \Phi \tilde{P}(t_k) \Phi^T + \Delta t F F^T \quad (5.63)$$

where $\Phi = e^{A\Delta t}$ is the transition matrix with sampling time $\Delta t = t_k - t_{k-1}$ and A is given by

$$A(t_{k+1}) = \left. \frac{\partial f[x(t), u(t), \beta]}{\partial x} \right|_{x=\hat{x}(t_k)} \quad (5.64)$$

Update

$$K(t_k) = \tilde{P}(t_k) C^T(t_k) [C(t_k) \tilde{P}(t_k) C^T(t_k) + G G^T]^{-1} \quad (5.65)$$

$$\hat{x}(t_k) = \tilde{x}(t_k) + K(t_k)[z(t_k) - \tilde{y}(t_k)] \quad (5.66)$$

$$\begin{aligned} \hat{P}(t_k) &= [I - K(t_k) C(t_k)] \tilde{P}(t_k) = [I - K(t_k) C(t_k)] \tilde{P}(t_k) [I - K(t_k) C(t_k)]^T \\ &\quad + K(t_k) G G^T K^T(t_k) \end{aligned} \quad (5.67)$$

where C is given by

$$C(t_k) = \left. \frac{\partial g[x(t), u(t), \beta]}{\partial x} \right|_{x=\tilde{x}(t_k)} \quad (5.68)$$

Equation (5.63) for propagation of the covariance of the state prediction error is an approximation obtained by assuming that Δt is small. In Eq. (5.67), two equivalent forms are provided to compute the updated covariance matrix of the state error \hat{P} . For the reasons elaborated in Appendix F, we use the recommended longer “Joseph” form, which is better conditioned for numerical computations and helps to ensure that P remains positive definite.

The time varying filter further requires the initial state covariance matrix P_0 to be specified. In the case of nonlinear systems, x_0 are estimated along with the other unknown parameters. In such a case P_0 is zero.

The response gradients $\partial y / \partial \Theta$ required in Eqs. (5.59) and (5.69), as well as the gradients of the system functions, Eqs. (5.64) and (5.68), can be evaluated using either the analytical differentiation or finite-difference approximation. For the reasons that we have already discussed in this chapter, they are approximated using finite-differences.^{10,12}

For a small perturbation δx_j in each of the n_q state variables, the gradients in Eqs. (5.64) and (5.68) are approximated using the central difference formulas as

$$A_{ij}(t_{k+1}) = \frac{1}{2\delta x_j(t_k)} [f_i[x(t_k) + \delta x_j e^j, u(t_k), \beta] - f_i[x(t_k) - \delta x_j e^j, u(t_k), \beta]]_{x=\hat{x}(t_k)} \quad (5.69)$$

$$C_{ij}(t_k) = \frac{1}{2\delta x_j(t_k)} [g_i[x(t_k) + \delta x_j e^j, u(t_k), \beta] - g_i[x(t_k) - \delta x_j e^j, u(t_k), \beta]]_{x=\tilde{x}(t_k)} \quad (5.70)$$

Likewise, for a small perturbation $\delta \Theta_j$ in each component of Θ , the perturbed response variables \tilde{y}_{p-j} are computed. The corresponding sensitivity coefficient is then approximated by:

$$\left[\frac{\partial \tilde{y}(t_k)}{\partial \Theta} \right]_{ij} \approx \frac{\tilde{y}_{pi}(t_k) - \tilde{y}_i(t_k)}{\delta \Theta_j} \quad (5.71)$$

which is the same as Eq. (5.62). The perturbed responses $\tilde{y}_p(t_k)$ are obtained from the perturbed system equations. These equations have a form similar to those of Eqs. (5.61)–(5.68), except that each time Θ is replaced through perturbed parameter vector $(\Theta + \delta \Theta_j e^j)$. The reader can easily write these equations. It is easy to infer from Eq. (5.71) that we have to propagate the perturbed state variables $x_p(t_k)$ and the perturbed error-covariance matrix $P_p(t_k)$ for each element of Θ .

All the quantities required in Eq. (5.58)–(5.60) to compute iteratively the parameter improvement $\Delta \Theta$ are now defined. The residuals $[\tilde{z}(t_k) - \tilde{y}(t_k)]$ required in Eq. (5.60) are readily available as a part of Eq. (5.66) in the filter implementation. The covariance matrix of residuals $R(t_k)$ can be obtained without any additional computation by making use of the relation

$$R(t_k) = C(t_k)P(t_k)C^T(t_k) + GG^T \quad (5.72)$$

where the right-hand side is already evaluated in Eq. (5.65). In this section hitherto it has been assumed that the state noise and measurement noise distribution matrices, F and G , are known, and that the filter algorithm of Eqs. (5.61)–(5.70) is formulated accordingly in terms of these F and G matrices and not in terms of R , the covariance matrix of the residuals. It becomes possible to include the elements of the F matrix in the unknown parameter vector Θ , but we do not estimate the elements of G as unknown parameters Θ applying the Gauss–Newton step, for the reasons of convergence which we have discussed several times now.

For the filter error method using a steady-state filter, the filter implementation was done in terms of R (as already discussed in the foregoing section). This was found to be convenient because the cost function in Eq. (5.4) is defined in terms

of R and also the steady-state R can be estimated using a closed-form solution. It is also possible to implement the steady-state filter in terms of G . From the explicitly estimated steady-state R , the measurement noise covariance matrix GG^T can be derived with only minor additional computations, but was not necessary, because, as just mentioned, the filter is implemented in terms of R .

For a time-varying filter, the maximum likelihood estimate of $R(t_k)$ can be obtained by setting the partial derivative of Eq. (5.57) with respect to $R(t_k)$ to zero. This yields:

$$R(t_k) = [z(t_k) - \tilde{y}(t_k)][z(t_k) - \tilde{y}(t_k)]^T \quad (5.73)$$

The time-varying filter implementation, however, requires knowledge of the G matrix. Although Eq. (5.73) provides the maximum likelihood estimate of $R(t_k)$ at each time point, it appears that a simple procedure to obtain updated G estimates from the updated $R(t_k)$, similar to the one in the steady-state case of Sec. V.A, is not feasible for a time-varying filter.

In view of this restriction, we do not consider estimation of GG^T using a time-varying filter. This limitation is a drawback of the time-varying filter method. In this sense, this approach does not lead to a complete, integrated solution yielding maximum likelihood estimates of all relevant quantities affecting the system. Nevertheless, reasonable information about the G matrix can be obtained from the laboratory calibration of the various measurement sensors or applying Fourier smoothing techniques.²² Although this is a feasible approach, we do not go into detail, because such approaches have been used in the past only on sample test cases and have not yet proved applicable on a routine basis. Moreover, any inaccuracies in these estimates of G will affect the estimates of the system parameters Θ . It is from this aspect that we tend to prefer the well established filter error method using the steady-state filter.

VI. Initial Noise Covariance Matrix

In the case of the output error method accounting for measurement noise only, there is no state update equation, that is, in other words the Kalman gain matrix K is zero. The predicted and the corrected states are one and the same, which are obtained by integration of the system equations. The residuals are not functions of R . This allows for a given Θ (either specified arbitrarily or obtained through some startup procedure) to compute the covariance matrix R using Eq. (4.15). This is then followed by the Gauss–Newton step to compute the new Θ . Thus, the two-step procedure in this case amounts to:

- 1) Choose suitable initial values for Θ .
- 2) Estimate R using Eq. (4.15).
- 3) Update parameter vector Θ applying the Gauss–Newton method, Eqs. (4.29) and (4.30).
- 4) Iterate on step 2 until convergence.

In the case of the filter error method accounting for both process and measurement noise, a slightly different approach is necessary. In order to compute the states and the residuals, it is first required to obtain the state noise prediction

error matrix P , and the Kalman gain matrix K . This in turn requires the matrix R to be specified, see Eq. (5.8). The matrix cannot be assumed to be a null matrix, because it will then lead to numerical difficulties in the solution of the Riccati equation for P . However, once an initial guess for R is provided, the two-step procedure can be implemented as:

- 1) Choose suitable initial values for Θ .
- 2) Generate or specify initial values for R .
- 3) Update parameter Θ by the Gauss–Newton method, Eqs. (5.18)–(5.20).
- 4) Estimate R using Eq. (5.11), compensate for the estimates of F , Eq. (5.41).
- 5) Iterate on step 3 until convergence.

How to specify the starting values for R may not be obvious in each case. Such information may come from laboratory calibration, from Fourier smoothing or from prior estimation results. Just a rough order of magnitude is generally good enough. It also becomes possible to automatically generate this information by incorporating an option to compute the matrix R in the first pass, if R is not specified.¹⁶ In such a case, we initially treat the complete noise as measurement noise, that is, basically running the algorithm in the output error mode. This can be easily done for a given Θ , assuming F and in turn K to be zero. In the subsequent iterations, F gets estimated and R automatically adjusted in the maximum likelihood sense through the details provided. This easy and pragmatic approach is found to work very well in several of the aircraft applications, and is hence recommended to be followed for the filter error method based on the steady-state filter. It is also recommended to carry out the a few initial iterations in which R is kept constant.

VII. Extension of Filter Error Method to Multiple Experiments

We briefly discussed in Chapter 3 the necessity of performing separate flight maneuvers (experiments) to excite different modes of aircraft motion, and in Chapter 4 we have shown that the output error method can be readily extended to analyze multiple maneuvers simultaneously, yielding one set of parameters. In the case of the filter error method, however, the analysis of multiple experiments simultaneously requires some considerations.

Recall from Chapter 3, Sec. III that the option of multiple experiments treats the initial conditions and biases separately for each of the maneuvers. In such a case, dropping the process noise matrix F , we represented in Eq. (3.12) the single set of unknown parameters Θ as

$$\Theta = [\beta^T \ \alpha_1^T \ \alpha_2^T \dots \alpha_{nE}^T \ \gamma_1^T \ \gamma_2^T \dots \gamma_{nE}^T \ \delta_1^T \ \delta_2^T \dots \delta_{nE}^T]^T \quad (5.74)$$

where n_E is the number of experiments to be analyzed simultaneously. Now, when we include the elements of the process noise distribution matrix in the unknown parameters Θ , Eq. (5.74) leads to

$$\Theta = [\beta^T \ \lambda^T \ \alpha_1^T \alpha_2^T \dots \alpha_{nE}^T \ \gamma_1^T \gamma_2^T \dots \gamma_{nE}^T \ \delta_1^T \delta_2^T \dots \delta_{nE}^T]^T \quad (5.75)$$

where λ denotes the elements of the process noise distribution matrix F . The reader may note that Eq. (5.75) is based on the assumption that the process noise matrix remains the same for all the experiments, that is, we have just one set of λ , like the system parameters Θ . The steady-state Riccati equation

$$AP + PA^T - \frac{1}{\Delta t} PC^T R^{-1} CP + FF^T = 0 \quad (5.76)$$

which is the same as Eq. (5.14), can be solved once to yield a single steady-state P , and subsequently from Eq. (5.8) or (5.48) a single gain matrix K , common to all the time segments, can be obtained. In such a case, we can apply the filter error algorithm that we have discussed in the foregoing sections without any difficulty. The unconstrained optimization and the constrained optimization to ensure KC to be less than 1 do not pose any difficulties.

The appropriateness of this assumption of the same process noise for all the experiments is, however, questionable. If the multiple experiments are carried out under similar atmospheric conditions with low magnitude turbulence, this assumption, although not strictly correct, might still be considered acceptable, because the estimates are somewhat insensitive to the process noise distribution matrix. In many cases this might be the case, because we usually carry out several maneuvers in a sequence during a single flight, or different flights might have been performed when atmospheric conditions were stable. In the other case, that is, when turbulence level is medium to high and different from experiment to experiment, then multiple experiment analysis based on the parameter vector of Eq. (5.75) poses some practical and some fundamental difficulties.

For the sake of argument, consider that we wish to estimate matrix F separately for each experiment, then the unknown parameter can be represented as:

$$\Theta = [\beta^T \ \lambda_1^T \ \lambda_2^T \dots \lambda_{nE}^T \ \alpha_1^T \ \alpha_2^T \dots \alpha_{nE}^T \ \gamma_1^T \ \gamma_2^T \dots \gamma_{nE}^T \ \delta_1^T \ \delta_2^T \dots \delta_{nE}^T]^T \quad (5.77)$$

Treating elements of F separately for each experiment leads to an increased number of parameters, assuming F to be diagonal, $n_q \cdot n_E$ instead of n_q for the single F common to all experiments. However, this is just an additional nuisance, requiring increased memory and computational time. It would be possible to compute the Kalman gain matrices for each time segment appropriately, compute the system responses from Eq. (5.46) and compute the covariance matrix of the innovations from Eq. (5.11). The first step of the Gauss–Newton method, Eqs. (5.18)–(5.20) including the perturbed gain matrices, Eq. (5.56), could be computed. After the unconstrained optimization, the inequality constraints on KC have to be checked, and, if violated, a constrained optimization according to Eqs. (5.35)–(5.39) performed.

The gain matrix K , which is a function of F , will be different for each time segment being analyzed. This difference is further aggravated in the case of non-linear systems, because we will have to compute the observation matrix C , linearized at the corresponding initial conditions [α in Eq. (5.77)], which will also be different from experiment to experiment. Thus, KC will be different due to different F as well as C . The constrained optimization given in Eqs. (5.35)–(5.39) does

not cater to such a case. Constraining the solution of KC for any one particular time segment, may not guarantee that constraints are satisfied for all the n_E experiments. This is a fundamental difficulty, and how to account for variable multiple constraints is not yet resolved. Owing to these practical and theoretical difficulties, the only pragmatic approach would be to deal with a single process noise distribution matrix by judiciously selecting the experiments to be analyzed simultaneously. In several cases this should be adequate, because, as suggested in Chapter 3, Sec. III, we analyze multiple experiments carried out specifically for system identification purposes, which are from the same flight or from flights under similar atmospheric conditions.

VIII. Explicit Modeling of Gust Spectrum

The filter error method as presented in the foregoing sections is a generic approach in which the atmospheric turbulence, $w(t)$, was treated as a white noise process, and we identified the process noise distribution matrix F , affecting the system state variables, along with the other unknown system parameters. It is sometimes argued that the characterization of atmospheric turbulence as a white noise is a poor representation and that the generic approach does not capitalize on the possible a priori knowledge about the model structures for the turbulence. Therefore, first we briefly indicate such an alternative approach based on the “state vector augmentation” and then evaluate the pros and cons of both of them.

For illustration purposes, the approach of state-vector augmentation is at best demonstrated by considering a simplified model pertaining to the longitudinal motion valid for small perturbations about the trim:

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -g \\ Z_u/U_0 & Z_\alpha/U_0 & 1 & 0 \\ 0 & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_{\delta_e} \\ Z_{\delta_e}/U_0 \\ M_{\delta_e} \\ 0 \end{bmatrix} \delta_e \quad (5.78)$$

where the state vector is given by $x = [u \ \alpha \ q \ \theta]^T$, δ_e is the elevator input, U_0 the velocity component of the trim speed in the x -direction. $X_{(.)}$, $Z_{(.)}$, and $M_{(.)}$ denote the dimensional derivatives pertaining to the longitudinal force, vertical force, and pitching moment coefficients respectively. The observation variables are given by $[a_x, a_z, u, \alpha, q, \theta]$.

Now, in order to account for the turbulence in the longitudinal and vertical directions, we model these stochastic disturbances using a Dryden spectrum assuming stationary and homogenous gusts.^{23,24} In order to incorporate such an explicit model in the parameter estimation procedure, we have to arrive at a model form compatible with the state space model of our system, for example Eq. (5.78). It can be shown that the Dryden spectrum can be described through a first order Gauss–Markov process of the form:

$$\dot{x} = -ax + b\xi \quad (5.79)$$

where x is an arbitrary variable, representing either longitudinal or vertical gust component, and $\xi(t) \sim \mathcal{N}(0, q_x)$ is the white noise (Gaussian distributed with

zero mean and variance q_x). For a stationary Gauss–Markov process, the power spectral density is given by⁷

$$S_x(\omega) = \frac{b^2 q_x}{a^2} \frac{1}{1 + (\omega/a)^2} \quad (5.80)$$

Equation (5.80) yields typical spectrum for $a = \omega_E U_0 / L$, $b = U_0 \sqrt{(2\omega_E/\pi)L}$, where L is the characteristic length in meter and ω_E the corner frequency in radians per second; q_x is the mean square intensity of the gust. For longitudinal and vertical gusts, typically, ω_E is 1 and 2.4 rad/s respectively.⁷ Thus, they are characterized through the mean square intensities and scale (characteristic) lengths (\bar{u}_g^2, L_u) and (\bar{w}_g^2, L_w). Combining Eqs. (5.79) and (5.80), and introducing them in Eq. (5.78), leads to an extended model representation of the form:

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{u}_g \\ \dot{w}_g \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -g & -X_u & -X_\alpha/U_0 \\ Z_u/U_0 & Z_\alpha/U_0 & 1 & 0 & -Z_u/U_0 & -Z_\alpha/U_0^2 \\ 0 & M_\alpha & M_q & 0 & 0 & -M_\alpha/U_0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -U_0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-U_0}{L_u} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-\omega_E U_0}{L_w} \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \\ u_g \\ w_g \end{bmatrix} \quad (5.81)$$

$$+ \begin{bmatrix} X_{\delta_e} \\ Z_{\delta_e}/U_0 \\ M_{\delta_e} \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta_e + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ U_0 \sqrt{2/\pi L_u} & 0 \\ 0 & U_0 \sqrt{2\omega_E/\pi L_w} \end{bmatrix} \begin{bmatrix} \xi_{ug} \\ \xi_{wg} \end{bmatrix}$$

Thus, we have now two additional state variables u_g and w_g representing the longitudinal and vertical gust velocities. The unknown parameters are the intensities and scale lengths of the gusts, (\bar{u}_g^2, L_u) and (\bar{w}_g^2, L_w). The inputs ξ_{ug} and ξ_{wg} are the white noise processes, which are generated using the random number generator. Thus, through state variable extension, it becomes possible to include longitudinal and vertical gusts based on the knowledge about the form of the gust spectrum as described here or using other forms of representation.²⁵

Parameter estimation based on such an extended model is by no means simpler than that using the generic approach. We still have to apply the complex filter error method incorporating a suitable state estimator. Moreover, if we assume that not only the longitudinal motion is affected by the gusts, but also the lateral-directional motion as well, extensions to coupled motion would require further additional states. Even when we restrict our attention to the longitudinal motion, it has been found that the estimation of the scale lengths and gust

intensities poses convergence problems, and provides inconsistent estimates compared with the expected values based on physical understanding of the atmosphere. Therefore, in some cases attempts to estimate statistical properties of turbulence were considered futile.⁸

To summarize, we can say that, based on a Dryden spectrum, it is possible to incorporate explicit gust models in parameter estimation. This approach of state-vector augmentation is, however, not flexible enough, because it is coupled to the form of the model chosen to represent the system. The system representation should be pertinent to the physical phenomenon being investigated and may vary from time to time and case to case. Moreover it leads to a larger size state vector. The number of additional parameters required in the state-augmentation and the generic approach remains typically of the same order. However, the generic approach has the advantage that no additional states are required to account for the turbulence in the parameter estimation, and is hence computationally faster. It is flexible with good convergence properties. Besides these advantages, because our major interest is to identify aircraft parameters, the form of the turbulence model is of secondary importance, and as such it is of little importance which approach we follow. Based on our past experience that clearly showed the generic approach to be the better one, we recommend that the generic approach based on Sec. V.A be applied for more reliable and accurate estimation of aerodynamic characteristics from flight data in turbulence.

IX. On the Equivalence of Output Error and Filter Error Methods

Having studied the output error and filter error methods, let us go more deeply into the exact differences between the two and try to understand precisely when the two methods are mathematically equivalent. We discuss the case when there is no process noise ($F = 0$). Besides the assumption of control inputs u being known exactly, if we further assume that the initial conditions are also known precisely, then there would be no (state) estimation problem.²⁶ In the case of linear systems, the state prediction given by Eq. (5.5), which is repeated here for convenience,

$$\tilde{x}(t_{k+1}) = \Phi\hat{x}(t_k) + \Psi B\bar{u}(t_k) + \Psi b_x, \quad x(t_0) = 0 \quad (5.82)$$

would provide the exact solution.

We come across loosely made statements that the output error and filter error methods are the same when there is no process noise. Mathematically, however, this is true only in the exercises that deal with simulated data, because in such a case we know the initial conditions of the states exactly, and we would have switched off the process noise while generating the data to be analyzed. However, in any exercise dealing with flight measured data, the assumptions of initial conditions x_0 being known exactly and the process noise, F , being zero are never met in practice. For the sake of argument, if we presume that the process noise might be neglected, even then we never know the true values of the initial conditions for the data being analyzed, because of the measurement errors and measurement noise. The two methods treat the initial conditions statistically differently.

In the sequel, when we consider the filter error method with process noise distribution matrix $F = 0$, it is not strictly equivalent to the output error method, because we still account for the uncertainties in the initial conditions. For this reason we will get some small differences in the numerical estimates of the parameters when we set $F = 0$ and apply the filter error method, compared with the pure output error method (as discussed in Chapter 4, which completely neglects the statistics of w and x_0). The differences are usually very small, and within an acceptable order of the numerical accuracy. In fact, from this subtle aspect, it can be argued that the estimates using the filter error algorithm with $F = 0$ are more likely to be reliable and better, at least from the statistical point of view, since the uncertainties in the state are accounted for.

X. FEM Software

As in the case of the output error method, we provide in this book a simple version of the software for the filter error method using a steady-state filter that runs under Matlab®. It caters for estimating the system parameters, process noise distribution matrix and measurement noise distribution matrix from a single maneuver. We provide here software packages for both types of system model implementations, namely for linear models in terms of system matrices and for general nonlinear models in terms of state and observation equations. A detailed README is found under the main directory, /FVSysID/chapter05/. The software packages help to understand better the various algorithmic steps that we have so far studied in this chapter. Furthermore, they allow us to generate the results that we will discuss here.

Figure 5.2 shows a flow chart of the computational procedure for the filter error method catering for linear systems. It is based on the details provided in Secs. II and IV. The function names are provided at the bottom of each block to trace the programming details. The main program is called “ml_fem_lin.” The model definition is in terms of system matrices A , B , C , D , and F , and bias vectors b_x and b_y . It suffices here to mention that “ml_fem_lin” is provided to gain deeper understanding of the complex computational procedure followed in the past;^{16,17} for practical purposes, we recommend the use of the software “ml_fem” that is capable of handling both linear and nonlinear systems. It is for this reason that we defer the detailed description of the software limited to linear systems to /FVSysID/chapter05/fem_linear/README_Linear.

Figure 5.3 shows a flow chart of the computational procedure for the filter error method applicable to general nonlinear systems. It is based on the details provided in Sec. V.A. In the case of nonlinear systems, we estimate the system parameters and noise covariances, but pre-specify the initial conditions and keep them fixed. Extension of the software to estimate x_0 for nonlinear systems is, of course, possible, but it is left to the readers.²⁷ While applying this software to linear systems, bias parameters of state and observation equations can be included and estimated. The software provides an option to selectively keep parameters included in the postulated model fixed, and estimate the remaining. A comparison of Figs. 5.2 and 5.3 reflects the differences that were highlighted in Table 5.1. It turns out that implementation of the software “ml_fem,” based on Sec. V.A, is somewhat more straightforward

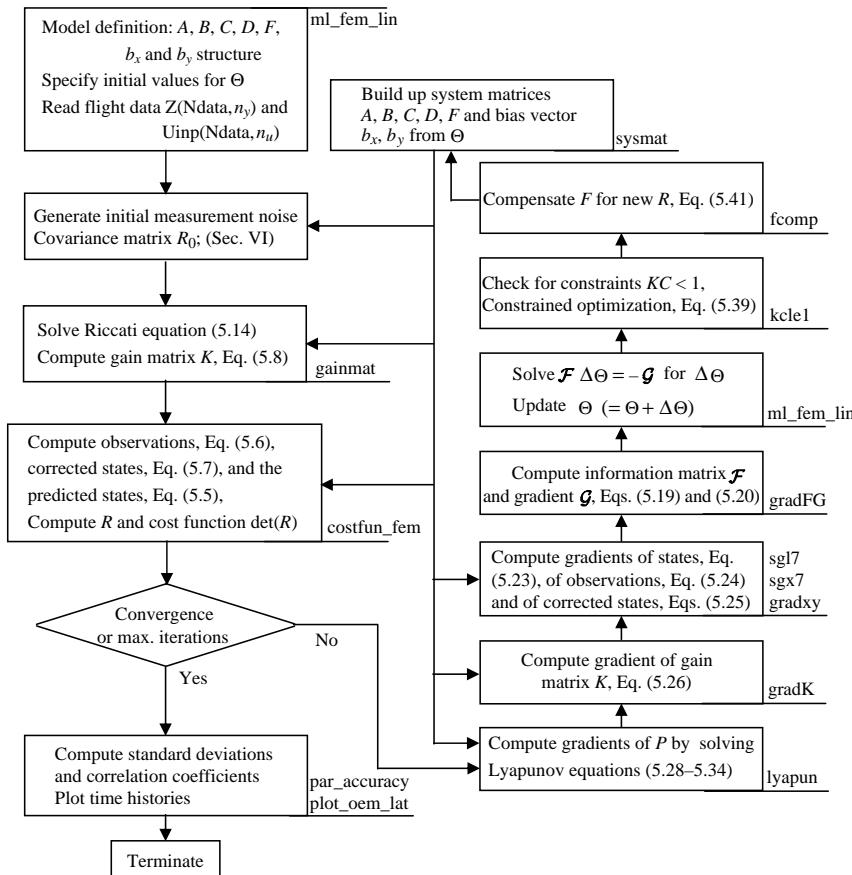


Fig. 5.2 Details of implemented filter error method for linear systems.

than that of the “ml_fem_lin,” particularly for computation of various gradients.

From Fig. 5.3 it can be seen that we code the postulated model in functions “state_eq” and “obser_eq.” The gain matrix K is computed in “gainmat.” The function “costfun_fem” propagates the states, computes the observations and the cost function. The perturbed gain matrices are computed in “gainmatPert,” the function “gradFG” propagates the perturbed state equations, computes the perturbed system responses, the gradient G , and the information matrix F ; the parameter updates $\Delta\Theta$ are computed in the main program “ml_fem” applying the Gauss–Newton method. We prefer to use Cholesky factorization to solve for $\Delta\Theta$. Then we check for the constraints in “kcle1” and perform constrained optimization, if necessary. The F -estimates are compensated for new R in “fcomp,” but only after iterFupR-iterations (recommended value 3). The convergence checking is performed in “ml_fem,” as is the step-size control, including

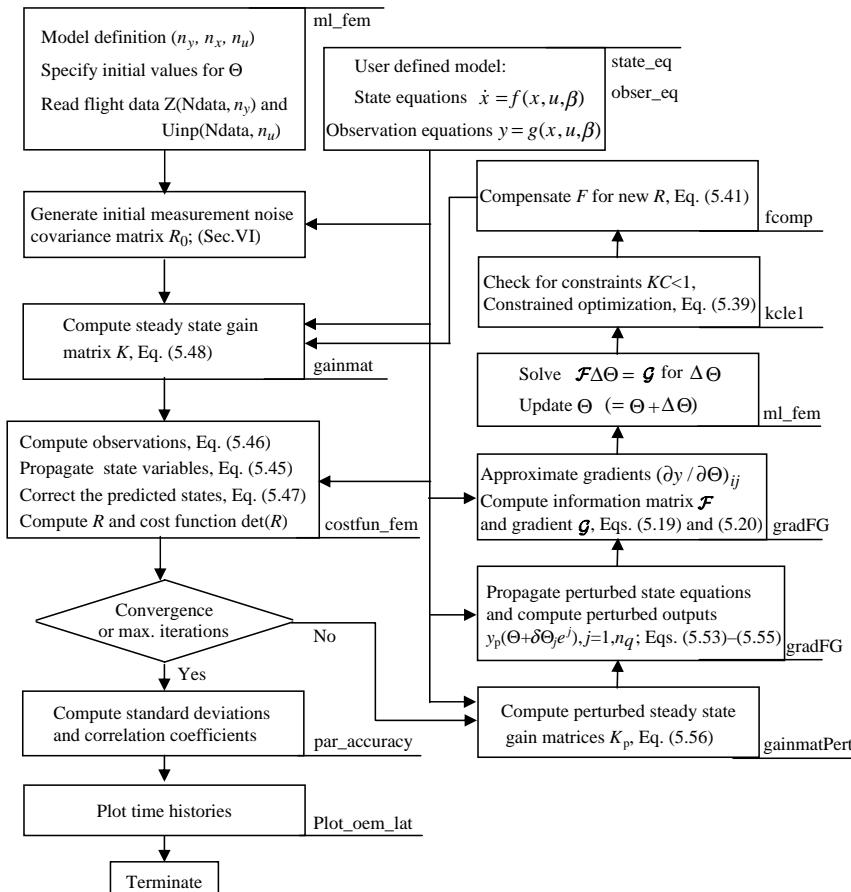


Fig. 5.3 Details of filter error method for general nonlinear systems.

halving of the parameter steps to overcome intermediate divergence. At the end, we compute the standard deviations and correlation coefficients in “par_accuracy” and finally make plots of the time histories and of the estimates. The source code (Matlab m-files) for the filter error method is provided in the directory/FVSysID/chapter05/.

The starting point is the main program called “ml_fem.” It provides the user interface to define the model and read the flight data. Different system models to be analyzed are denoted through the integer flag “test_case”.

test_case index for the test case

which is to be uniquely associated with the state space model and with the user provided interface in the form of a function defining the model parameters, flight data, and other relevant details. Different state space models, to be coded by the

user, can be invoked by defining the following flags and strings in a function `mDefCase**` called in “`ml_fem`,” where `**` denotes the test case index:

<code>state_eq</code>	function name (to code the right hand side of the state equations)
<code>obser_eq</code>	function name (to code the right hand side of the observation equations)
<code>Nx</code>	number of state variables (n_x)
<code code="" ny<=""></code>	number of output variables (n_y)
<code>Nu</code>	number of input variables (n_u)
<code>Nparsys</code>	number of system parameters
<code>Nparam</code>	total number of system and bias parameters

The variable names used in the program “`ml_fem`” and in the called functions are shown on the left-hand side, followed by the description and the notation used to denote these variables in the text. As in the case of OEM software elaborated in Chapter 4, Sec. XIX, default values are defined in the main program “`ml_fem`” for the maximum number of iterations (`niter_max`), the convergence criterion (`tolR`), and the step size for the parameter perturbations (`par_step`). The default value is also provided for “`iterFupR`,” the iteration count from which F -estimate will be corrected using the latest R -estimate according to Eq. (5.41). They can be suitably adapted, if necessary in a specific case. It is obvious that the model specification in terms of (`Nx`, `Ny`, `Nu`, `Nparam`) must match those coded in the above two user functions for the postulated model.

As discussed in Sec. VI, an option has been provided to generate the residual covariance matrix. If required, this can be specified by the user by setting the integer flag `iSD` and defining the standard deviations of the output errors. It is also necessary to specify the starting values for the parameters, and the integer flags to indicate whether the particular parameter is free (to be estimated) or fixed (to be kept fixed).

<code>iSD</code>	index for the initial residual covariance matrix = 0: Default (Recommended); = 1: User specified standard deviations of output errors
<code>param</code>	starting values for unknown parameters (Θ)
<code>parFlag</code>	flags for free and fixed parameters = 1: free parameters (to be estimated) = 0: fixed parameters (not to be estimated)
<code>NparID</code>	total number of unknown parameters, that is, nonzero elements of <code>parFlag</code> (n_q).
<code>x0</code>	initial conditions on state variables (kept fixed)
<code>SDyError</code>	initial standard deviations of output errors, if <code>iSD</code> = 1

The flight data to be analyzed is also to be loaded in the function for the model definition, and requires specification or assignment of the following information:

<code>Ndata</code>	number of data points
<code>dt</code>	sampling time

t	time vector [= 0 to (Ndata - 1)*dt]
$Z(N_{\text{data}}, N_y)$	flight data for measured outputs (N, n_y)
$U_{\text{inp}}(N_{\text{data}}, N_u)$	flight data for measured control inputs (N, n_u)

While loading the flight data in the arrays $Z(N_{\text{data}}, N_y)$ and $U_{\text{inp}}(N_{\text{data}}, N_u)$, it is obvious that the dimensions of the variables must match those used in the postulated model and programmed in the state and observation functions `state_eq` and `obser_eq`, respectively.

XI. Examples

We now turn our attention to the application of the filter error method to test cases. To enable critical evaluation of the filter error method and to compare the estimates with the nominal parameter values, we first consider simulated aircraft responses and a linear model. The second example pertains to estimation of the lift, drag, and pitching moment coefficients from a nonlinear model postulate and the third example analyzes flight data gathered from several flights under different atmospheric conditions.

A. Application of Filter Error Method to Data with Process Noise

In Chapter 4, Sec. XX.B we analyzed data with a moderate to high level of turbulence generated through simulation.^{10–12,16} The same test case, pertaining to the aircraft lateral-directional motion, is now analyzed here, applying the filter error method.

The model pertaining to the lateral-directional motion is postulated as follows:

State equations

$$\begin{aligned}\dot{p} &= L_p p + L_r r + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r + L_v v + b_{xp} \\ \dot{r} &= N_p p + N_r r + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r + N_v v + b_{xr}\end{aligned}\quad (5.83)$$

Observation equations

$$\begin{aligned}\dot{p}_m &= L_p p + L_r r + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r + L_v v + b_{yp} \\ \dot{r}_m &= N_p p + N_r r + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r + N_v v + b_{yr} \\ a_{ym} &= Y_p p + Y_r r + Y_{\delta_a} \delta_a + Y_{\delta_r} \delta_r + Y_v v + b_{yay} \\ p_m &= p + b_{yp} \\ r_m &= r + b_{yr}\end{aligned}\quad (5.84)$$

The unknown parameter vector Θ for the preceding model consists of the dimensional derivatives and the bias parameters. It is given by:

$$\begin{aligned}\Theta^T &= [L_p \ L_r \ L_{\delta_a} \ L_{\delta_r} \ L_v \ N_p \ N_r \ N_{\delta_a} \ N_{\delta_r} \ N_v \\ &\quad Y_p \ Y_r \ Y_{\delta_a} \ Y_{\delta_r} \ Y_v \ b_{xp} \ b_{xr} \ b_{yp} \ b_{yr} \ b_{yay} \ b_{yp} \ b_{yr} \ f_{pp} \ f_{rr}] \quad (5.85)\end{aligned}$$

The reader may notice that we have not, for convenience of notation, shown explicitly the process noise distribution matrix in Eq. (5.83), but it is implicitly assumed that it is considered here, as evident from the unknown parameter vector of Eq. (5.85) which shows, compared with Eq. (4.92) in Chapter 4, two additional parameters f_{pp} and f_{rr} corresponding to the diagonal process noise distribution matrix.

Thus, the postulated model is defined as follows:

	No. of variables	Function name
States	2 p, r	“xdot_TC02_attas_lat”
Outputs	5 $\dot{p}, \dot{r}, a_y, p, r$	“obs_TC02_attas_lat”
Inputs	3 δ_a, δ_r, v	

The right-hand sides of the state and observation equations (5.83) and (5.84) are programmed in the functions “xdot_TC02_attas_lat,” and “obs_TC02_attas_lat,” respectively. Thus, the model definition for this example provided by the function “/FVSysID/chapter05/mDefCase02.m,” which is called from the main program “ml_fem,” is as follows:

```

test_case = 2; % index for the test case
state_eq = 'xdot_TC02_attas_lat'; % function for state equations
obser_eq = 'obs_TC02_attas_lat'; % function for observation equations
Nx = 2; % number of states
Ny = 5; % number of observation variables
Nu = 3; % number of input (control) variables
NparSys = 15; % number of system parameters
Nparam = NparSys + Nx % total number of parameters
            + Ny + Nx;
dt = 0.04; % sampling time

```

The data to be analyzed, that is, the arrays Z(Ndata,Ny) and Uinp(Ndata,Nu), are loaded from the file /FVSysID/flt_data/y13aus_da1.asc, which is also provided. In short, we have the same model definition as in the case of the output-error method, Chapter 4, Sec. XX.B, except that we have two additional parameters for the process noise distribution matrix. Caveat, the model definition file “mDefCase02.m” for the filter error method is very similar to that for the output error method, but not exactly the same. We use the default option (iSD = 0) to generate automatically the initial residual covariance matrix. Some suitable starting values have been specified for the Nparam parameters and the corresponding integer flags parFlags set to one, because we estimate all of them.

The results of parameter estimation applying both the software packages, “ml_fem_lin” and “ml_fem,” are summarized in Table 5.2. Nominal values of the derivatives are also provided in the same table, and also those estimated in Sec. XX.B in Chapter 4, applying the output error method. The filter error methods are found to converge smoothly within 10 iterations and yield estimates which are close to the nominal values. No numerical or convergence problems were encountered. On the other hand, the output error method required 42

Table 5.2 Estimates applying filter error and output error methods, test_case = 2

Derivative	Nominal Value	Estimates (standard deviation, %) obtained by applying		
		Filter error method		Output error method "ml_oem"
		"ml_fem_lin"	"ml_fem"	
L_p	-5.820	-5.718 (6.6)	-5.724 (6.6)	-2.3294 (30.2)
L_r	1.782	1.720 (9.1)	1.722 (9.1)	0.6284 (35.6)
$L_{\delta a}$	-16.434	-14.925 (11.1)	-14.953 (11.1)	-0.0413 (1136)
$L_{\delta r}$	0.434	0.200 (216)	0.201 (216)	-1.1342 (11.9)
L_v	-0.097	-0.0883 (12.8)	-0.0884 (12.9)	0.0031 (37.6)
N_p	-0.665	-0.621 (9.4)	-0.621 (9.4)	-7.8203 (29.0)
N_r	-0.712	-0.722 (3.4)	-0.722 (3.4)	2.0246 (34.6)
$N_{\delta a}$	-0.428	-0.427 (58.9)	-0.431 (58.4)	0.4457 (147.8)
$N_{\delta r}$	-2.824	-2.828 (2.4)	-2.828 (2.4)	-2.6080 (6.7)
N_v	0.0084	0.0092 (19.0)	0.0092 (19.0)	0.0027 (109.3)
Y_p	-0.278	-0.297 (27.7)	-0.297 (27.7)	1.8696 (46.2)
Y_r	1.410	1.415 (2.5)	1.415 (2.5)	0.9587 (26.3)
$Y_{\delta a}$	-0.447	-0.514 (72.9)	-0.515 (72.8)	0.4415 (120.9)
$Y_{\delta r}$	2.657	2.688 (3.7)	2.688 (3.7)	3.1632 (4.74)
Y_v	-0.180	-0.180 (1.5)	-0.180 (1.5)	-0.1664 (3.0)
f_{pp}	0.200	0.167 (4.2)	0.167 (4.2)	—
f_{rr}	0.200	0.124 (3.7)	0.124 (3.7)	—
$ R $		2.146×10^{-12}	2.146×10^{-12}	2.3558×10^{-9}
Iterations		10	10	42

iterations, had severe convergence problems at each iteration and yielded estimates which are not anywhere near the nominal values. This is evident from the comparison of the fourth and fifth columns of Table 5.2. The two programs "ml_fem_lin" and "ml_fem" yield the same results, as evident from the third and the fourth columns. Figure 5.4 shows the time history plots of the control inputs, and the comparison of the measured data with the model estimated responses. Compared with Fig. 4.9, the agreement is now significantly improved. Thus, for data with an appreciable level of turbulence, the filter error method is far superior.

A critical reader may notice that in Table 5.2 the estimates provided by the filter error method agree much better with the nominal values for some of the derivatives than those for a few others. This is attributed to the fact that we have analyzed just a single maneuver, and the estimates of the secondary derivatives are somewhat less accurate than the primary derivatives. In the present case the estimates of the stability derivatives, that is the derivatives associated with the state variables, namely L_p , L_r , N_p , and N_r , are significantly improved by the filter error method. This directly confirms the statement that we made in Section II, saying the process noise and the covariance matrix of the state prediction error P improve the model. Also the smooth progress of

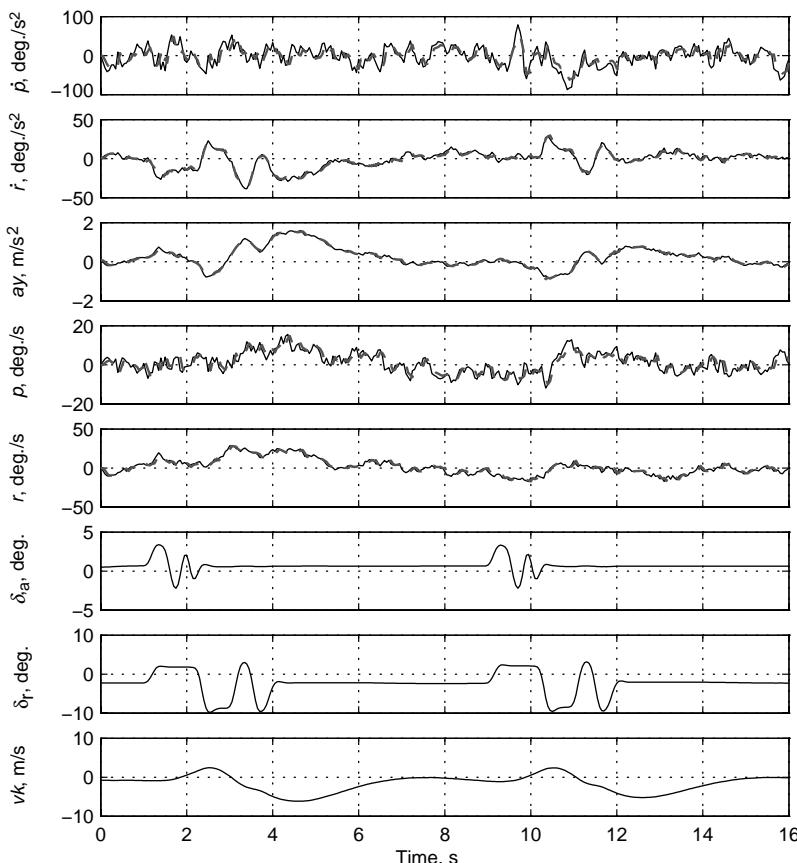


Fig. 5.4 Filter error method applied to data with moderate to high level of turbulence, test case 2 (—, flight measured; - - -, model predicted).

the optimizations with significantly fewer iterations confirms that the filter error method is less sensitive to stochastic disturbances and leads to a more nearly linear optimization problem, which has fewer local minima and a faster convergence rate.

B. Estimation of Longitudinal Derivatives

The second example pertains to determination of the lift, drag, and pitching moment coefficients of the research aircraft HFB-320. Flight tests were carried out to excite the longitudinal motion through a multi-step elevator input, resulting in short period motion and a pulse input leading to phugoid

motion.²⁸ The following model has been postulated to estimate the nondimensional derivatives:

State equations

$$\begin{aligned}\dot{V} &= -\frac{\bar{q}S}{m}C_D + g \sin(\alpha - \theta) + \frac{F_e}{m} \cos(\alpha + \sigma_T) \\ \dot{\alpha} &= -\frac{\bar{q}S}{mV}C_L + q + \frac{g}{V} \cos(\alpha - \theta) - \frac{F_e}{mV} \sin(\alpha + \sigma_T) \\ \dot{\theta} &= q \\ \dot{q} &= \frac{\bar{q}S\bar{c}}{I_y}C_m + \frac{F_e}{I_y}(\ell_{tx} \sin \sigma_T + \ell_{tz} \cos \sigma_T)\end{aligned}\quad (5.86)$$

where the lift, drag, and pitching moment coefficients are modeled as

$$\begin{aligned}C_D &= C_{D0} + C_{DV}\frac{V}{V_0} + C_{D\alpha}\alpha \\ C_L &= C_{L0} + C_{LV}\frac{V}{V_0} + C_{L\alpha}\alpha \\ C_m &= C_{m0} + C_{mV}\frac{V}{V_0} + C_{m\alpha}\alpha + C_{mq}\frac{q\bar{c}}{2V_o} + C_{m\delta_e}\delta_e\end{aligned}\quad (5.87)$$

Observation equations

$$\begin{aligned}V_m &= V \\ \alpha_m &= \alpha \\ \theta_m &= \theta \\ q_m &= q \\ \dot{q}_m &= \frac{\bar{q}S\bar{c}}{I_y}C_m + \frac{F_e}{I_y}(\ell_{tx} \sin \sigma_T + \ell_{tz} \cos \sigma_T) \\ a_{xm} &= \frac{\bar{q}S}{m}C_X + \frac{F_e}{m} \cos \sigma_T \\ a_{zm} &= \frac{\bar{q}S}{m}C_Z - \frac{F_e}{m} \sin \sigma_T\end{aligned}\quad (5.88)$$

where the longitudinal and vertical force coefficients C_X and C_Z are given by:

$$\begin{aligned}C_X &= C_L \sin \alpha - C_D \cos \alpha \\ C_Z &= -C_L \cos \alpha - C_D \sin \alpha\end{aligned}\quad (5.89)$$

where V is the true airspeed, α the angle of attack, θ the pitch attitude, q the pitch rate, δ_e the elevator deflection, F_e the thrust, σ_T the inclination angle of the

engines, $\bar{q}(=1/2\rho V^2)$ the dynamic pressure, m the mass, S the wing area, \bar{c} the wing chord, I_y the moment of inertia, and ρ the density of air. The subscript m on the left-hand side of Eq. (5.88) denotes the measured quantities.

The preceding system equations in terms of variables in the stability axes (V, α) contain not only the common trigonometric and multiplicative nonlinearities, but in addition, the variable dynamic pressure $\bar{q}(=1/2\rho V^2)$, which multiplies all of the aerodynamic derivatives, introducing an additional nonlinearity. Furthermore, inversion of the state variable V leads to further nonlinearities in the state equation for α .

The unknown parameter vector Θ consisting of the nondimensional derivatives is given by

$$\Theta = [C_{D0} \ C_{DV} \ C_{D\alpha} \ C_{L0} \ C_{LV} \ C_{L\alpha} \ C_{m0} \ C_{mV} \ C_{ma} \ C_{mq} \ C_{m\delta_e}]^T \quad (5.90)$$

From a set of flight maneuvers carried out with small inputs, a particular record that appears to contain atmospheric turbulence is analyzed here. A record length of 60 s with a sampling time of 0.1 s is used.

Thus, the postulated model for the program “ml_fem” is defined as follows:

	No. of variables	Function name
States	4 V, α, θ, q	“xdot_TC04_hfb_lon”
Outputs	7 $V, \alpha, \theta, q, \dot{q}, a_x, a_z$	“obs_TC04_hfb_lon”
Inputs	2 δ_e, F_e	

Since the variations in the altitude during the maneuver were small, the density of air ρ is assumed to be constant for the duration of the maneuver. We use a value of $\rho = 0.792 \text{ kg/m}^3$, corresponding to a nominal altitude of roughly 4200 m. The values for various geometrical quantities (mass, moments of inertia, reference area, chord, etc.) for the present case are found in the above two functions. The right-hand sides of the state equation (5.86) are programmed in the function “xdot_TC04_hfb_lon,” and those of the observation equation (5.87)–(5.89) in “obs_TC04_hfb_lon.” Thus, the model definition for this example provided by the function “/FVSysID/chapter05/mDefCase04.m,” called from the main program “ml_fem,” is as follows:

```

test_case = 4; % integer flag for the test case
state_eq = 'xdot_TC04_hfb_lon'; % function for state equations
obser_eq = 'obs_TC04_hfb_lon'; % function for observation equations
Nx      = 4; % number of states
Ny      = 7; % number of observation variables
Nu      = 2; % number of input (control) variables
NparSys = 11; % number of system parameters
Nparam = NparSys + Nx; % total number of parameters
dt      = 0.1; % sampling time

```

Once again, we use the default value for iSD = 0, specify suitable parameter values, and set the integer flags parFlag to one for all parameters.

To this case we once again apply both the output error method and the filter error method. Several different initial starting values for the parameters were tried out. It turns out that the output error method (/FVSySID/chapter04/ml_oem with test_case = 4) either did not converge or had convergence problems when the starting values were far from the optimum. However, starting from parameters somewhat closer to the optimum, the output error method converged, yielding the plots shown in Fig. 5.5. The match is reasonable, showing clear indications of turbulence seen from the plots for α and a_z , particularly for the first few seconds, and for V over the complete record.

On the other hand, the filter error method (/FVSySID/chapter05/ml_fem with test_case = 4) worked quite well for several sets of starting values. Starting from the same set of values as used while applying the output error method, the filter

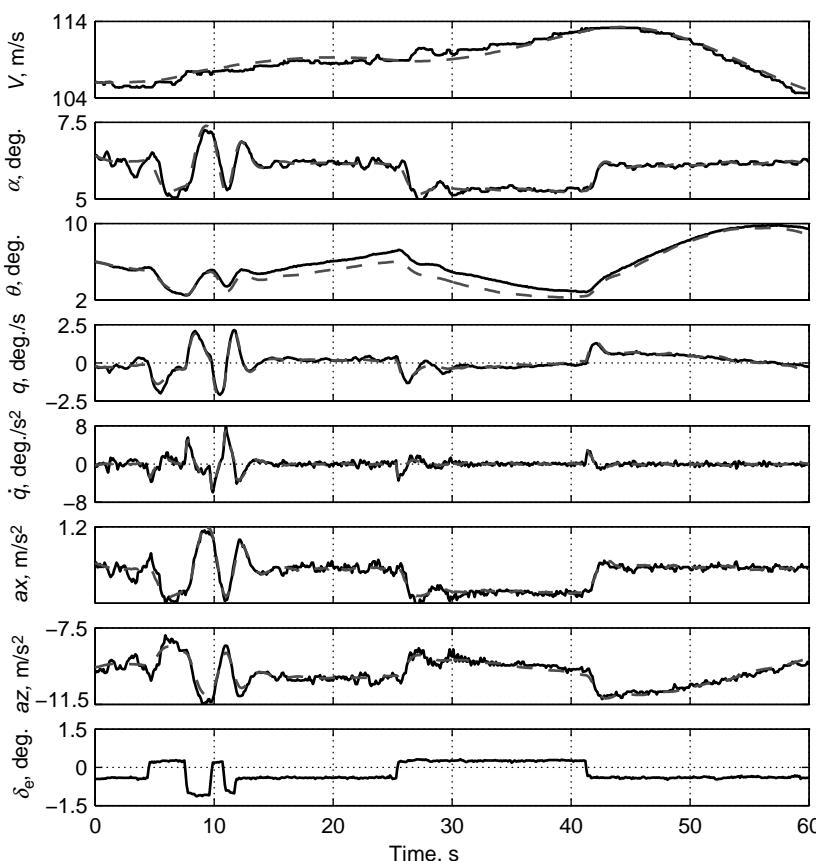


Fig. 5.5 Estimation of nondimensional derivatives by output error method (—, flight measured; - - - - -, model predicted).

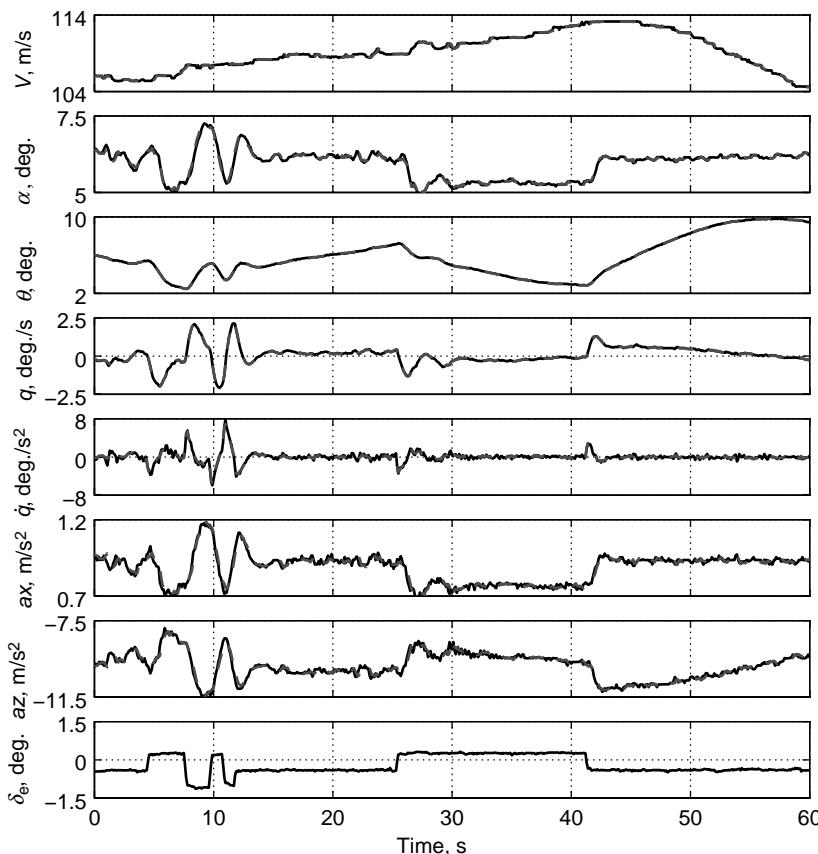


Fig. 5.6 Estimation of nondimensional derivatives by filter error method (—, flight measured; - - -, model predicted).

error method yields the response match shown in Fig. 5.6. The agreement between the flight measured and model estimated responses is found to be very good, and clearly brings out the improvement obtained by accounting for the process noise.

We now analyze the spectra of residuals from the two methods, shown in Fig. 5.7 for the four state variables. From Fig. 5.7a it is found that they are not flat for the output error method. The residuals show greater energy content at lower frequencies. On the other hand, Fig. 5.7b shows for all the four residuals flat spectra, thus indicating that the filter error method is better suited to account for the atmospheric disturbances and unavoidable modeling errors.

C. Estimation from Flight Data in Seemingly Steady Atmosphere

For flight data gathered in turbulence the filter error methods are necessary, because the output error method is known to yield biased estimates in the presence of atmospheric turbulence.^{6,13} Even in the case of flight maneuvers in

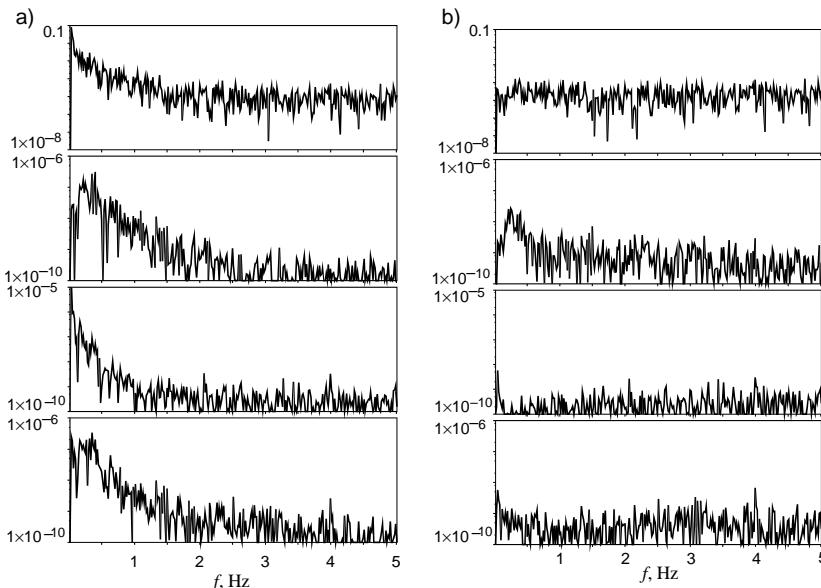


Fig. 5.7 Power spectral densities of residuals. a) Output error method, b) filter error methods.

smooth air, the filter error method could lead to better estimation results, since some of the unavoidable modeling errors are then treated as process noise characterized by low frequency contents rather than as measurement noise. Moreover, although it is generally argued that the flight tests for aircraft parameter estimation should be carried out in calm air, in any practical exercise one has no control over the prevailing atmospheric conditions or, owing to very tight time schedules and cost factors involved in a time-bound project, very little choice about waiting for steady atmospheric conditions.²⁹

As a typical example, the estimates of the weathercock stability derivative, $C_{n\beta}$, obtained by applying the output error and the filter error method to the same set of flight data are provided in Fig. 5.8a.³⁰ The C-160 data analyzed here were gathered from eight flights carried out during a span of less than two weeks, seven of them being in a seemingly steady atmosphere, whereas one encountered a moderate amount of turbulence. It is clear that the estimates provided by the output error method, particularly those for the flight 223 during which moderate turbulence was encountered, differ greatly from those of other flights at the same nominal flight conditions. Moreover, a fair amount of scatter is observed in the estimates from other flights in a seemingly steady atmosphere, making a final conclusion regarding the nature of the nonlinearity or fairing of data difficult. On the other hand, the filter error method yields clearly grouped estimates with much less scatter and the estimates from the flight 223 match the other estimates well. The nonlinear dependency of the weathercock stability on the angle of attack can now be observed much better.

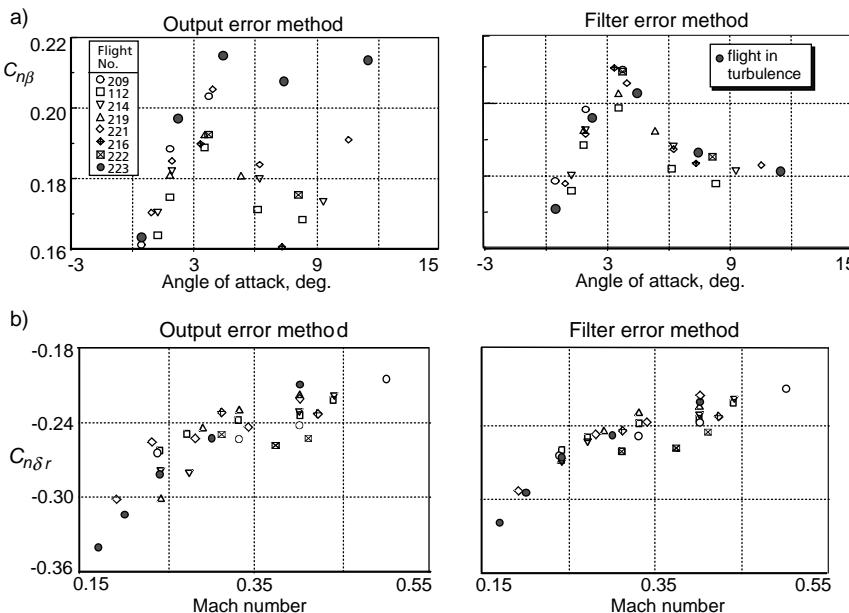


Fig. 5.8 Output error and filter error method applied to flight data in seemingly steady atmospheric conditions. a) State derivative, weathercock stability; b) control derivative, rudder effectiveness.

Figure 5.8b shows the estimates for the rudder effectiveness derivative $C_{n\delta r}$. The estimates of the control derivatives applying the output error method even for flight in moderate turbulence were found to be of similar magnitude to those from other flights for corresponding flight conditions, subject to a small scatter, the scatter being somewhat smaller for the filter error method. However, in an overall sense, the two methods provided comparable estimates of the control derivatives.

The improvements obtained in the estimates of the stability derivatives applying the filter error method were significant (see Fig. 5.8a), whereas they were marginal in the case of control derivatives, as seen in Fig. 5.8b. The better performance of the filter error method compared with the output error method in estimating the stability derivatives is attributed to the fact that these derivatives depend on the aircraft motion variables, that is, on the estimated states. The estimated states are affected by stochastic inputs such as atmospheric turbulence. On the other hand, the comparable performance of the two methods in the case of control derivatives is attributed to the fact that the control derivatives are identified from the deterministic inputs applied to the aircraft. The aircraft response to these control inputs is adequate to enable better identification, from flights both in seemingly steady air and with moderate turbulence. Somewhat lower scatter among the estimates from the filter error method could be partly due to the improved stability derivatives and partly due to accounting for the measurement

noise in the control inputs itself; see Fig. 5.1. The noise level in the control inputs is low; otherwise the output error method would not have yielded acceptable results in most of the flight conditions analyzed.³⁰

XII. Concluding Remarks

In this chapter we discussed the most complex stochastic approach to aircraft parameter estimation that provides capabilities to analyze flight data gathered under turbulent atmospheric conditions. Various process noise formulations were discussed, and it was argued that the combined natural cum innovation formulation is the best choice. Through a linear system representation, the various steps of the filter error method were elaborated. Subsequently, we discussed the extensions of the filter error method to general nonlinear systems. It has been demonstrated that a numerical approximation of gradients combined with a mixed state estimator using a nonlinear system model for state prediction and a correction based on a first-order approximation is a viable approach. We highlighted the exact differences between the algorithms for linear and nonlinear systems. As in the case of the output error method, we adopt here the relaxation strategy leading to a two-step algorithm to optimize the maximum likelihood cost function. The system parameters and process noise distribution matrix were updated by the Gauss–Newton algorithm. For state estimation both steady-state and time-varying filters were presented. In the majority of the cases, we prefer to use the steady-state filter implementation. Finally it is shown that the generic approach to estimate the noise distribution matrix is preferable over the other approach based on state augmentation using an explicit gust spectrum model.

Basic software for the filter error method using a steady-state (constant gain) estimator implemented under Matlab® has been provided to follow the various computational steps of the algorithm. It has been applied to three typical aircraft examples which were presented to demonstrate the performance of the filter error method. Samples of data with process noise have been supplied. The comparison of the estimates obtained by applying the two methods indicates that the filter error method yields significantly improved results compared with the output error method for flight data with atmospheric turbulence. Even in the case of flight data in seemingly smooth air, the filter error method provided estimates of the stability derivatives with a much smaller scatter than that from the output error method. The estimates of the control derivatives provided by the two methods agreed fairly well.

The examples provide an answer to the question often raised regarding the practical utility of the filter error method. It can be pragmatically concluded that these methods can yield better estimates, are no longer limited to linear systems, and are indispensable for many applications. These advantages outweigh the disadvantage of higher computational overheads. The limitations of these methods are mainly twofold: 1) they allow estimation of a single set of process noise distribution matrix, which implies they are mostly applicable to a single flight maneuver or to multiple maneuvers carried out under similar atmospheric conditions; as elaborated in this chapter, the difficulties in extending the filter error method to multiple experiments treating process noise distribution

matrix separately for each maneuver have not yet been resolved; 2) the filter error method always yields a good match for the responses, which is a necessary condition for parameter estimation, but not a sufficient one for parameters and model structure to be correct. The good response match tends to mask the discrepancies which may result from unaccounted aerodynamic or other, for example aeroelastic, effects, and as a consequence the model improvements may not be very obvious. In the worst case, we may obtain a good response match even when an important parameter has been inadvertently omitted. Thus, some care is necessary while applying the advanced filter error method. To this end, some practical tips have been provided. It is for the reasons just mentioned that the filter error method is used in practice for special cases only; the majority of applications, including those covered later in a separate chapter on selected advanced examples, are based on the use of output error or least squares methods.

References

- ¹Balakrishnan, A. V., "Stochastic System Identification Techniques," in *Stochastic Optimization and Control*, edited by H. F. Karreman, John Wiley & Sons, London, 1968.
- ²Mehra, R. K., "Maximum Likelihood Identification of Aircraft Parameters," *Proceedings of the 11th Joint Automatic Control Conference*, Atlanta, GA, 1970, pp. 442–444.
- ³Mehra, R. K., "Identification of Stochastic Linear Dynamic Systems Using Kalman Filter Representation," *AIAA Journal*, Vol. 9, Jan. 1971, pp. 28–31.
- ⁴Tyler, J. S., Powell, J. D., and Mehra, R. K., "The Use of Smoothing and Other Advanced Techniques for VTOL Aircraft Parameter Identification," Final report, N0019-69-C0534, Naval Air Systems Command Contract Systems Control Inc., Palo Alto, CA, June 1970.
- ⁵Ilfiff, K. W., "Identification and Stochastic Control with Applications to Flight Control in Turbulence," Ph.D. Dissertation, University of California, Los Angeles, CA, May 1973.
- ⁶Maine, R. E. and Iliff, K. W., "Formulation and Implementation of a Practical Algorithm for Parameter Estimation with Process and Measurement Noise," *SIAM Journal of Applied Mathematics*, Vol. 41, No. 3, 1981, pp. 558–579.
- ⁷Schulz, G., "Maximum-Likelihood-Identifizierung mittels Kalman-Filterung—Kleinste Quadrate Schätzung. Ein Vergleich bei der Bestimmung von Stabilitätsderivativa unter Berücksichtigung von Böenstörungen," DLR-FB 75–54, August 1975 (in German).
- ⁸Foster, G. W., "The Identification of Aircraft Stability and Control Parameters in Turbulence," RAE TR-83025, March 1983.
- ⁹Yazawa, K., "Identification of Aircraft Stability and Control Derivatives in the Presence of Turbulence," AIAA Paper 77-1134, Aug. 1977.
- ¹⁰Jategaonkar, R. V. and Plaetschke, E., "Estimation of Aircraft Parameters Using Filter Error Methods and Extended Kalman Filter," DFVLR-FB 88-15, March 1988.
- ¹¹Jategaonkar, R. V. and Plaetschke, E., "Identification of Moderately Nonlinear Flight Mechanics Systems with Additive Process and Measurement Noise," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 2, 1990, pp. 277–285.
- ¹²Jategaonkar, R. V. and Plaetschke, E., "Algorithms for Aircraft Parameter Estimation Accounting for Process and Measurement Noise," *Journal of Aircraft*, Vol. 26, No. 4, 1989, pp. 360–372.
- ¹³Maine, R. E. and Iliff, K. W., "Identification of Dynamic Systems," AGARD AG-300 Vol. 2, Jan. 1985.

¹⁴Gelb, A., *Applied Optimal Control*, The MIT Press, Cambridge, MA, 1974.

¹⁵Grewal, M. S. and Andrews, A. P., *Kalman Filtering Theory and Practice*, Prentice Hall, Upper Saddle River, NJ, 1993.

¹⁶Jategaonkar, R. V. and Plaetschke, E., "Maximum Likelihood Estimation of Parameters in Linear Systems with Process and Measurement Noise," DFVLR-FB 87-20, June 1987.

¹⁷Maine, R. E. and Iliff, K. W., "User's Manual for MMLE3, a General FORTRAN Program for Maximum Likelihood Parameter Estimation," NASA TP-1563, Nov. 1980.

¹⁸Vaughan, D. R., "A Nonrecursive Algebraic Solution for the Discrete Riccati Equation," *IEEE Transactions on Automatic Control*, Vol. AC-15, Oct. 1970, pp. 597–599.

¹⁹Potter, J. E., "Matrix Quadratic Solutions," *SIAM Journal of Applied Mathematics*, Vol. 14, 1966, pp. 496–501.

²⁰Holley, W. E. and Wei, S. Y., "Improved Method for Solving the Algebraic Riccati Equation," *Journal of Guidance and Control*, Vol. 3, No. 2, 1980, pp. 190–192.

²¹Bazaraa, M. S. and Shetty, C. M., *Nonlinear Programming: Theory and Algorithm*, John Wiley & Sons, New York, 1979.

²²Morelli, E. A., "Estimating Noise Characteristics from Flight Data Using Optimal Fourier Smoothing," *Journal of Aircraft*, Vol. 32, No. 4, 1995, pp. 689–695.

²³Dryden, H. L., *A Review of the Statistical Theory of Turbulence*, Interscience Publishers, New York, 1961.

²⁴Houbolt, J. C., "Atmospheric Turbulence," *AIAA Journal*, Vol. 11, No. 4, 1973, pp. 421–437.

²⁵Iliff, K. W., "Identification and Stochastic Control of an Aircraft Flying in Turbulence," *Journal of Guidance and Control*, Vol. 1, No. 2, 1978, pp. 101–108.

²⁶Sorenson, H. W., "Kalman Filtering Techniques," in *Advances in Control Systems*, Vol. 3, edited by C. T. Leondes, Academic Press, New York, 1966, pp. 219–239.

²⁷Jategaonkar, R. V., "A Modular and Integrated Software Tool for Parameter Estimation and Simulation of Dynamic Systems—User's Manual, Version 1.0," DLR-IB 111–2001/29, July 2001.

²⁸Mackie, D. B., "A Comparison of Parameter Estimation Results from Flight Test Data Using Linear and Nonlinear Maximum Likelihood Methods," DFVLR-FB 84–06, Dec. 1983.

²⁹Hamel, P. G. and Jategaonkar, R. V., "Evolution of Flight Vehicle System Identification," *Journal of Aircraft*, Vol. 33, No. 1, 1996, pp. 9–28.

³⁰Jategaonkar, R. V., "A Comparison of Output Error and Filter Error Methods from Aircraft Parameter Estimation Results," *Proceedings of the NAL-DLR Symposium on System Identification*, Nov. 1993, DLR-Mitt. 93-14, Dec. 1993, pp. 63–87.

Chapter 6

Equation Error Methods

I. Introduction

L EAST SQUARES estimation, originally stimulated by astronomical studies, is one of the oldest problems in estimation theory with numerous engineering applications, including flight vehicles. It was invented and applied during the period 1795–1806 independently by A.-M. Legendre and C.-F. Gauss to describe planetary motion.^{1,2} Historians have, however, acknowledged Gauss as the inventor, who applied his method to a meager set of three well-chosen observations to obtain a preliminary estimate, and used further observations made over a period of 41 days to improve the prediction of the orbital position of the asteroid “Ceres” discovered by the astronomer Piazzi in January 1801, and thereby helped to precisely relocate Ceres after a period of several months in December 1801.³ Application of his approach to several other planetary bodies located by astronomers proved the generality of the technique and marked the beginning of estimation theory.

Least squares (LS) techniques, also called regression analysis, belong to a class of methods called the equation-error methods, because they minimize a cost function defined directly in terms of an input–output equation. The cost function is not based on probability theory, as was the case with the maximum-likelihood method discussed in Chapter 4. The linear least squares technique is characterized by its mathematical simplicity, in the sense that the estimates are obtained by applying matrix algebra operations in a one-shot computational procedure. The regression techniques can be applied to nonlinear models as well, because the basic principle is to minimize the sum of squares of the errors between the measurements and model response. As for the cost function, it is immaterial whether the errors result from a linear or nonlinear model. Only the minimization procedure is model type-dependent. Nonlinear least squares problems can be solved only iteratively.

As the least squares method characterizes the cause–effect relation solely in terms of independent and dependent variables, its performance and applicability is strongly influenced by the data quality, assumptions made regarding the noise and availability of the necessary variables. The classical ordinary least squares (OLS) method assumes the independent variables to be error and noise free, and dependent variables corrupted by uniformly distributed noise. An extension called weighted least squares (WLS) accounts for nonconstant scatter in the

residuals. The major limitation of these methods is that they yield asymptotically biased and inconsistent estimates in the presence of measurement errors and noise in the independent variables.^{4–7} Nevertheless, this method has found several applications in aircraft parameter estimation, providing acceptable results compared with more complex methods, mainly for two reasons. First, high-quality sensors and instrumentation systems minimize these errors. Secondly, prior to applying the regression method, more reliable signals can be generated through a data preprocessing step.^{7–13} An extension called total least squares (TLS) is also possible to account for noise in the independent variables.

As covered in Chapters 3–5, for several applications dealing with dynamical systems including flight vehicles, the models are postulated in state space. The least-squares method can be applied to such models provided that all the states and their time derivatives are measured. The use of measured states and state derivatives eliminates the state estimation problem, which had to be solved by integration in the case of the output error method and by a suitable state estimator incorporating a Kalman filter or an extended Kalman filter in the case of the filter error methods. However, the requirement that all states and state-derivatives be measured directly, as well as the assumption of only the state-derivatives being corrupted by noise may, in general, be restrictive in a practical application. On the other hand, since the LS method does not rely on the temporal relation defining explicit sequential ordering in time of states/variables between the data points, it facilitates analysis of large amplitude maneuvers applying linear regression to several smaller portions through a process called “data partitioning.” Furthermore, the least squares method is directly amenable to a step-by-step procedure of selecting the best independent variables in the regression equation.

In this chapter we derive the least-squares estimator from basic principles, and study its properties under theoretical assumptions based on practical considerations. Its applicability to state space models is presented, which allows direct comparison with the approaches covered in the previous chapters. The weighted least squares method, which accounts for variability in the errors in the dependent variables, and the total least squares, which helps to eliminate the effects of noise in the independent variables, are presented. Nonlinear and multi-output regression approaches have been elaborated and their similarity with the output error method has been brought out. A related procedure of instrumental variable is illustrated briefly. Stepwise regression has been elaborated, looking pragmatically at its relevance in the presence of a priori knowledge about the model structure. A simple example of estimating aerodynamic characteristics from flight data has been presented to bring out the necessary steps to compute force and moment coefficients from flight measured data. Application of a one-shot procedure as well as an iterative procedure considering linear and nonlinear equations have been demonstrated using the software and flight data provided.

II. Least Squares Method

In general, the least squares estimation problem can be stated as follows. We have been given N discrete data samples $[y_i(1); y_i(2); \dots, y_i(N)]$ of an i th

dependent variable, and $[x(1); x(2); \dots, x(N)]$ of independent variables, where $x = (x_1 \ x_2 \ \dots \ x_{nq})^T$ is an $n_q \times 1$ vector. We assume that the measured dependent variable, $y_i(k)$, at each discrete time point k depends linearly on the independent variables:

$$y_i(k) = \theta_1 x_1(k) + \theta_2 x_2(k) + \dots + \theta_n x_n(k) + \varepsilon(k); \quad k = 1, 2, \dots, N \quad (6.1)$$

where $\theta = (\theta_1 \ \theta_2 \ \dots \ \theta_{nq})^T$ denotes the vector of unknown parameters and ε is the equation error representing modeling discrepancies and/or noise in the dependent variable y . It is not possible to measure the parameters θ directly; hence, it is required to estimate the system parameters θ . In our specific case of flight vehicle system identification, θ are the unknown stability and control derivatives to be estimated from the measured (flight data derived) aerodynamic force and moment coefficients denoted as y . The independent variables x comprise of aircraft motion variables like angular rates, flow angles and control surface deflections. The θ are assumed constant over all data samples N , and x is assumed to be measured exactly and error-free. A block schematic of the least-squares approach is shown in Fig. 6.1. Although not included explicitly in Eq. (6.1), the procedure allows estimation of bias terms (e.g., aerodynamic zero parameters) by simply adding a vector of all ones as an independent variable.

The above formulated problem deals with a dependent variable which is a function of more than one independent variable, and as such represents a problem called “multiple linear regression.” In our specific case of flight vehicle system identification, as a rule, this is the case. The single dimensional case with just one independent variable is of theoretical interest only, and hence we do not elaborate on it here. Accordingly, whenever we say least-squares or regression analysis, it is automatically implied that we are talking about multiple linear regression analysis. In the literature on estimation theory, different terminologies are used to characterize the least squares problem. The dependent variable y , as it will be called in this book, is also termed as “response

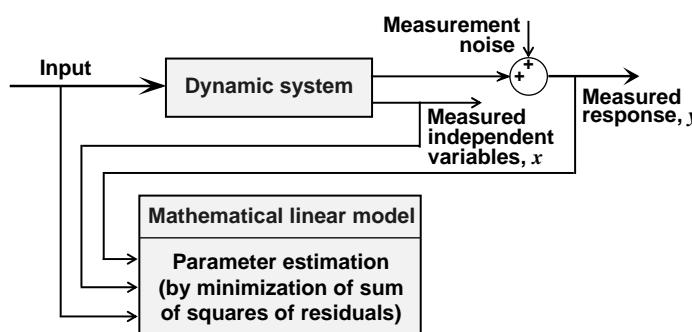


Fig. 6.1 Block schematic of least-squares method.

variable" or "observation." The independent variables x are alternatively termed "regressors," "explanatory variables," or "predictor variables." The deterministic part of Eq. (6.1), defining the relationship between the dependent and independent variables, is commonly called a "regression equation"; sometimes it is also called a "smoothing function." The OLS method is also sometimes called, particularly in the neural-network literature, a least mean squares estimator (LMS).

In Eq. (6.1) we have denoted the independent variable as y_i , that is, y with subscript i , implying that we may have more than one dependent variable. For the moment, however, to develop the estimation procedure and to study their properties, we restrict our attention to a single dependent variable, because each is usually modeled separately. For such a one-dimensional case, we drop for convenience of notation the subscript i from the above equation.

In Eq. (6.1) the y and x are provided (as measured data) and hence known; it contains n_q unknowns, namely $(\theta_i, i = 1, n_q)$, which can be solved in an ideal case from exactly n_q pairs of dependent and independent variables, provided $\varepsilon = 0$, that is, in the absence of noise and error. This is the minimum number of data samples that are required for the determination of the unknown parameters. However, due to the presence of errors in the measurements of dependent variables, that is, $\varepsilon \neq 0$, we cannot find exact values of θ . We rather try to find the best approximations based on some suitable criterion, which requires more than n_q samples of data to estimate the parameters and leads to an overdetermined problem; loosely speaking, to average out the noise effects on the parameter estimates. Intuitively, the larger the number of such redundant data, the better will be the averaging out process, yielding more reliable estimates.

Equation (6.1) can be rewritten in matrix notation as

$$y(k) = x^T(k)\theta + \varepsilon(k) \quad (6.2)$$

Thus, for each of the N discrete time points at which the measurements are available, Eq. (6.2) can be written as

$$\begin{aligned} y(1) &= x^T(1)\theta + \varepsilon(1) \\ y(2) &= x^T(2)\theta + \varepsilon(2) \\ &\dots \\ y(N) &= x^T(N)\theta + \varepsilon(N) \end{aligned} \quad (6.3)$$

Equation (6.3) can be rewritten in matrix notation as

$$Y = X\theta + \varepsilon \quad (6.4)$$

where $Y = [y(1) \ y(2) \dots \ y(N)]^T$ and $\varepsilon = [\varepsilon(1) \ \varepsilon(2) \dots \ \varepsilon(N)]^T$ are $N \times 1$ size vectors and X is the $(N \times n_q)$ matrix of independent variables:

$$X = \begin{bmatrix} x_1(1) & x_2(1) & \dots & x_{n_q}(1) \\ x_1(2) & x_2(2) & \dots & x_{n_q}(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(N) & x_2(N) & \dots & x_{n_q}(N) \end{bmatrix} \quad (6.5)$$

The errors (also called residuals), ε , are assumed to be uncorrelated with the independent variables. Furthermore, the scatter in the residuals is assumed to be constant across the values of independent variables, that is, the standard deviation of the random errors in y is constant. The residuals are assumed to be a white noise process with zero mean and variance σ^2 , that is, $E\{\varepsilon\} = 0$ and $E\{\varepsilon\varepsilon^T\} = \sigma^2 I$, where I is the identity matrix.

From Eq. (6.2), the equation error at the discrete time point k is given by

$$\varepsilon(k) = y(k) - x^T(k)\theta \quad (6.6)$$

The errors at N discrete time points can be written from Eq. (6.6) or from Eq. (6.4) in vector form as

$$\begin{aligned} \varepsilon &= [\varepsilon(1) \ \varepsilon(2) \dots \varepsilon(N)]^T \\ &= Y - X\theta \end{aligned} \quad (6.7)$$

The least squares estimates of the unknown parameters θ are obtained by minimizing the sum of the squares of the residuals, or alternatively the weighted sum of the squares of the residuals which we will discuss in the next section. In both the cases the basic principle is, however, not based on the probability theory, as was the case with the maximum-likelihood method discussed in Chapter 4. Accordingly, the cost function for the least-squares estimation is defined as

$$\begin{aligned} J(\theta) &= \frac{1}{2} \sum_{k=1}^N \varepsilon^2(k) = \frac{1}{2} \varepsilon^T \varepsilon \\ &= \frac{1}{2} [Y - X\theta]^T [Y - X\theta] = \frac{1}{2} [Y^T - \theta^T X^T][Y - X\theta] \end{aligned} \quad (6.8)$$

The gradient of the cost function with respect to the parameters is given by

$$\frac{\partial J(\theta)}{\partial \theta} = -Y^T X + \theta^T (X^T X) \quad (6.9)$$

Since the error $\varepsilon(k)$ is a linear function of the parameters θ , see Eq. (6.6), the minimum of the cost function $J(\theta)$ defined in Eq. (6.8) is obtained

by setting the gradient of $J(\theta)$ with respect to θ to zero. Accordingly, equating Eq. (6.9) to zero, taking the transpose, and denoting $\hat{\theta}$ the estimates of the true parameter values θ , we arrive at the so-called *normal equation*:

$$(X^T X)^T \hat{\theta} = X^T Y \quad (6.10)$$

Assuming $(X^T X)$ to be invertible, a unique solution to Eq. (6.10) is given by

$$\hat{\theta} = (X^T X)^{-1} X^T Y \quad (6.11)$$

which are the least squares estimates of θ . Where $(X^T X)$ happens to be singular, then there will be multiple solutions. For parameter estimation purposes from specially conducted flight maneuvers, the data matrix is, in general, nonsingular.

The matrix $X^T X$ is called the information matrix, because it is a measure of the information content in the data being analyzed. Thus, based on the postulated model of Eq. (6.1) the estimates of the system parameters are obtained through matrix algebra operations given by Eq. (6.11) with the data matrix X of the independent variables and Y the vector of dependent variables. We do not go into the various techniques of matrix-inversion, except to say that a procedure based on Cholesky factorization or singular value-decomposition is numerically more efficient. Under the Matlab® environment, the job is even simpler because standard procedures are readily available. In particular, the use of the backslash operator is recommended, mainly because of better error detection properties.

A. Properties of Least Squares Estimates

As in the case of the maximum likelihood method, the accuracy of the least squares estimates can be studied in terms of bias, covariance of the parameter estimation error, and consistency. For the assumptions made, namely 1) residuals are uncorrelated with the independent variables and represented by a white noise process with $E\{\varepsilon\} = 0$ and $E\{\varepsilon\varepsilon^T\} = \sigma^2 I$, and 2) the independent variables are error free, the least squares estimates are unbiased, efficient and consistent.

The bias in the least-squares estimates is given by $E\{\hat{\theta}\}$, the expected values of the estimates $\hat{\theta}$. Substituting Eq. (6.11) for $\hat{\theta}$, we get

$$E\{\hat{\theta}\} = E\{(X^T X)^{-1} X^T Y\} \quad (6.12)$$

Now substitution of Eq. (6.4) for Y in terms of the true value θ , and the use of linearity of expectations (namely, expected value of a sum or difference is the

sum or difference of the expected values) leads to

$$\begin{aligned} E\{\hat{\theta}\} &= E\{(X^T X)^{-1} X^T (X\theta + \varepsilon)\} \\ &= E\{(X^T X)^{-1} X^T X\theta\} + E\{(X^T X)^{-1} X^T \varepsilon\} \\ &= E\{\theta\} + E\{(X^T X)^{-1} X^T \varepsilon\} \end{aligned} \quad (6.13)$$

However, by definition we know that $E\{\theta\}$ is nothing but its true value θ . Since X is assumed to be noise-free, the matrix of independent variables is known exactly. Equation (6.13) then simplifies to

$$E\{\hat{\theta}\} = \theta + (X^T X)^{-1} X^T E\{\varepsilon\} \quad (6.14)$$

Now, making use of the assumptions made regarding the noise statistics, we know that $E\{\varepsilon\}$ is zero. Accordingly, the last term on the right-hand side of Eq. (6.14) vanishes, yielding

$$E\{\hat{\theta}\} = \theta \quad (6.15)$$

Equation (6.15) implies that the least squares estimation yields unbiased estimates, subject to the assumptions made.

To verify whether the least-squares procedure is an efficient (i.e., minimum-variance) estimator or not, we need to derive an expression for the parameter error covariance matrix P , that is $E\{(\hat{\theta} - \theta)(\hat{\theta} - \theta)^T\}$, which can be written by expanding the terms within the brackets as

$$\begin{aligned} P &= E\{(\hat{\theta} - \theta)(\hat{\theta} - \theta)^T\} = E\{\hat{\theta}\hat{\theta}^T - \theta\hat{\theta}^T - \hat{\theta}\theta^T + \theta\theta^T\} \\ &= E\{\hat{\theta}\hat{\theta}^T\} - \theta E\{\hat{\theta}^T\} - E\{\hat{\theta}\}\theta^T + \theta\theta^T \end{aligned} \quad (6.16)$$

Now making use of the property that the least-squares estimates are unbiased, substituting $E\{\hat{\theta}\} = \theta$ from Eq. (6.15) in the above equation leads to:

$$P = E\{\hat{\theta}\hat{\theta}^T\} - \theta\theta^T \quad (6.17)$$

Substituting the least squares solution of Eq. (6.11) in Eq. (6.17), making use of the matrix algebra result $(AB)^T = B^T A^T$ successively, and knowing that the data matrix X is known exactly from the measurements, leads to

$$\begin{aligned} P &= E\{[(X^T X)^{-1} X^T Y][(X^T X)^{-1} X^T Y]^T\} - \theta\theta^T \\ &= E\{(X^T X)^{-1} X^T Y Y^T X (X^T X)^{-1}\} - \theta\theta^T \\ &= (X^T X)^{-1} X^T E\{Y Y^T\} X (X^T X)^{-1} - \theta\theta^T \end{aligned} \quad (6.18)$$

As in the case of the derivation for unbiased estimates, we now substitute the regression equation (6.4) for Y , which results in

$$\begin{aligned}
 P &= (X^T X)^{-1} X^T E\{(X\theta + \varepsilon)(X\theta + \varepsilon)^T\} X(X^T X)^{-1} - \theta\theta^T \\
 &= (X^T X)^{-1} X^T E\{(X\theta\theta^T X^T + \varepsilon\theta^T X^T \\
 &\quad + X\theta\varepsilon^T + \varepsilon\varepsilon^T\} X(X^T X)^{-1} - \theta\theta^T \\
 &= (X^T X)^{-1} X^T [X\theta\theta^T X^T + E\{\varepsilon\}\theta^T X^T \\
 &\quad + X\theta E\{\varepsilon^T\} + E\{\varepsilon\varepsilon^T\}] X(X^T X)^{-1} - \theta\theta^T
 \end{aligned} \tag{6.19}$$

Since the errors ε are assumed to have zero mean value, $E\{\varepsilon\} = 0$, and constant covariance, $E\{\varepsilon\varepsilon^T\} = \sigma^2 I$, it leads to

$$\begin{aligned}
 P &= (X^T X)^{-1} X^T [X\theta\theta^T X^T + \sigma^2 I] X(X^T X)^{-1} - \theta\theta^T \\
 &= (X^T X)^{-1} X^T X\theta\theta^T X^T X(X^T X)^{-1} \\
 &\quad + \sigma^2 (X^T X)^{-1} X^T X(X^T X)^{-1} - \theta\theta^T \\
 &= \theta\theta^T + \sigma^2 (X^T X)^{-1} - \theta\theta^T \\
 &= \sigma^2 (X^T X)^{-1}
 \end{aligned} \tag{6.20}$$

Thus, the inverse of the information matrix is the covariance matrix, representing the expected error in the parameter estimates. Usually σ^2 is unknown. Its estimate is given by

$$\hat{\sigma}^2 = \frac{1}{N - n_q} \sum_{k=1}^N [y(k) - X(k)\hat{\theta}]^2 \tag{6.21}$$

To investigate the asymptotic consistency of the estimates, we have to check the behavior of P for a large number of data samples. For this purpose we rewrite the covariance matrix given by Eq. (6.20) as:

$$P = \frac{1}{N} \sigma^2 \left(\frac{1}{N} X^T X \right)^{-1} \tag{6.22}$$

Now, taking the limit as $N \rightarrow \infty$, it is obvious that the first term on the right-hand side, $1/N$, vanishes asymptotically, yielding:

$$\begin{aligned}
 \lim_{N \rightarrow \infty} P &= \lim_{N \rightarrow \infty} \left\{ \frac{1}{N} \sigma^2 \left(\frac{1}{N} X^T X \right)^{-1} \right\} \\
 &= 0
 \end{aligned} \tag{6.23}$$

where it is assumed that limit of $(X^T X)^{-1}$ exists. Equation (6.23) implies that the parameter error covariance matrix P asymptotically converges to zero, which in turn implies that there is no error in the estimates $\hat{\theta}$, and hence $\hat{\theta} = \theta$, the true values. Thus, the least squares estimator is asymptotically consistent.

B. Practical Considerations

The least squares procedure developed in Sec. II and the properties derived in Sec. II.A are based on the assumption that the measurements of the independent variables x are exact and noise-free, and accordingly the variables x and matrix X represent the true values. In reality, this is never the case; every recorded variable contains unavoidable measurement noise as well as systematic errors like bias or scale factor; recall our discussion on systematic errors in Chapter 3. Let us denote the measured independent variables x_m , the matrix of x_m as X_m (note that the subscript m is used in this section to denote explicitly the measurements, and not the m th variable), the systematic errors as Δx , and the noise in the independent variable as μ , which we also assume to be additive and independent of ε . The true value of the independent variables is then given by

$$X = X_m - (\Delta X + \mu) \quad (6.24)$$

where ΔX is the matrix of Δx , which is assumed constant over the N data samples. Substituting Eq. (6.24) into Eq. (6.4) leads to

$$Y = (X_m - \Delta X - \mu)\theta + \varepsilon \quad (6.25)$$

Pre-multiplying Eq. (6.25) with $(X_m^T X_m)^{-1} X_m^T$ yields

$$(X_m^T X_m)^{-1} X_m^T Y = (X_m^T X_m)^{-1} X_m^T [(X_m - \Delta X - \mu)\theta + \varepsilon] \quad (6.26)$$

Except for the change of notation, the left-hand side of Eq. (6.26) is nothing but $\hat{\theta}$, the estimate of unknown parameters. Expanding the terms on the right-hand side of Eq. (6.26) gives:

$$\begin{aligned} \hat{\theta} &= (X_m^T X_m)^{-1} X_m^T X_m \theta - (X_m^T X_m)^{-1} X_m^T (\Delta X + \mu) \theta \\ &\quad + (X_m^T X_m)^{-1} X_m^T \varepsilon \\ &= \theta + (X_m^T X_m)^{-1} X_m^T (\varepsilon - \Delta X \theta - \mu \theta) \end{aligned} \quad (6.27)$$

Based on Eq. (6.27) we can now check the properties of the least squares estimator in the presence of systematic errors and noise in the independent variables. From Eq. (6.27), the expected value of the estimates is obtained as

$$E\{\hat{\theta}\} = \theta + E\{(X_m^T X_m)^{-1} X_m^T (\varepsilon - \Delta X \theta - \mu \theta)\} \quad (6.28)$$

Now applying the linearity of expectation and the multiplicative rule to the expectation of a product (namely, expected value of a product of two independent variables is the product of the expected values), based on the assumptions of noise ε in the dependent variables being zero mean, and μ independent of ε and θ , Eq. (6.28) simplifies to

$$E\{\hat{\theta}\} = \theta - E\{(X_m^T X_m)^{-1} X_m^T (\Delta X + \mu)\} \theta \quad (6.29)$$

Following a procedure similar to that used in Eqs. (6.18)–(6.20), in the present case, it can be shown that the covariance of the parameter estimation error P is given by

$$\begin{aligned} P = & E\{(X^T X)^{-1} X^T \varepsilon \varepsilon^T X (X^T X)^{-1}\} \\ & + E\{(X^T X)^{-1} X^T (\Delta X + \mu) \theta \theta^T (\Delta X + \mu)^T X (X^T X)^{-1}\} \end{aligned} \quad (6.30)$$

Comparing Eq. (6.29) with Eq. (6.15), it becomes clear that the estimates are now affected by the combined errors $(\Delta x + \mu)$, and in general yield biased estimates. The bias depends only on the errors in the measurement of the independent variables, and not on the noise in the dependent variable. The variance is affected by the noise levels of dependent as well as independent variables.

Let us further analyze Eq. (6.29) in terms of the two components of possible corruptions, namely μ (noise) and Δx (systematic errors/bias or scale factor). For argument sake, if we assume that there are no systematic errors, that is, $\Delta x = 0$, then the estimates will be unbiased, provided the noise μ in the independent variables is zero mean; such a case is dealt with in Sec. V. In the other case, that is, in the presence of systematic errors Δx , the estimates will always be biased. It is this undesirable property that needs consideration while applying the least squares method to flight data, and in turn necessitates critical checking of recorded data to eliminate such errors, before the data can be used for regression purposes. Thus, in practice, application of the least squares procedure calls for a data preprocessing step to eliminate the errors, and thereby leads to a two-step procedure for aerodynamic parameter estimation. As already mentioned in Chapter 2 and will be pursued in Chapter 10, it becomes possible at some additional computational costs to check for the kinematic consistency of data applying the flight path reconstruction technique based either on filtering techniques or maximum-likelihood method and to eliminate such errors from the measured data.

In an analogous manner, the presence of atmospheric turbulence (process noise) can be investigated. In such cases it turns out that the covariance of the parameter estimation error will be affected by process noise as well. As in the case of measurement noise μ in the independent variables, if we assume that the process noise is a white noise process having zero mean, then the estimates will be unbiased, provided systematic errors are eliminated. It is this property that implies that the least squares estimator accounts for the process noise, because the measured variables contain responses due to deterministic inputs as well as those due to nonmeasurable stochastic inputs.

C. Applicability to State Space Models

To bring out the applicability of the least squares principle to models represented in state space, consider the following linear system representation:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (6.31)$$

Now assuming that the measurements of all the states x and of their time derivatives \dot{x} are available and denoted as x_m and \dot{x}_m , respectively, the equation error can be expressed as

$$\varepsilon_{EE} = \dot{x}_m - Ax_m(t) - Bu(t) \quad (6.32)$$

where the subscript m is used to denote measured variables; note that we have not used subscript m for the control inputs, because the control inputs have to be measured in any case, whether we apply the equation error method or any other and as such the subscript m is implicit for the u . Furthermore, note that \dot{x}_m is not a differential operation of the state variables, but the measurements. This implies that the least squares criterion does not involve solution of the system differential equations, that is, there is no state estimation required. Instead, the system equations are used with the measured data.

For N discrete measurements, the cost function to be minimized can be written as:

$$J_{LS}(\theta) = \sum_{k=1}^N [\dot{x}_m(k) - Ax_m(k) - Bu(k)]^T [\dot{x}_m(k) - Ax_m(k) - Bu(k)] \quad (6.33)$$

Combining the control vector u with the measured state vector x_m , and also the matrices B and A into a single partitioned matrix as

$$\theta = \{B : A\} \quad (6.34)$$

Equation (6.33) can be rewritten as

$$J_{LS} = \sum_{k=1}^N \left[\dot{x}_m(k) - \theta \begin{pmatrix} u \\ x_m \end{pmatrix} \right]^T \left[\dot{x}_m(k) - \theta \begin{pmatrix} u \\ x_m \end{pmatrix} \right] \quad (6.35)$$

Setting the gradient of Eq. (6.35) with respect to θ to zero yields

$$\theta^T = \left\{ \sum_{k=1}^N \begin{pmatrix} u \\ x_m \end{pmatrix} \begin{pmatrix} u \\ x_m \end{pmatrix}^T \right\}^{-1} \left\{ \sum_{k=1}^N \begin{pmatrix} u \\ x_m \end{pmatrix} \dot{x}_m^T \right\} \quad (6.36)$$

Equation (6.36) is an alternate form of the least squares solution of Eq. (6.11). It is now apparent that the least squares procedure can be applied to the state-space model provided the measurements of all the states and the derivatives of all the states with respect to time are available.

III. Weighted Least Squares Method

The least squares method dealt hitherto yields unbiased estimates when, as pointed out in Sec. II, the standard deviations of the random errors ε in the dependent variables are constant over all the values of the independent variables; see constant scatter band on the left-hand side of Fig. 6.2 for a single-dimensional case. In some cases the least squares fit would yield residuals with nonconstant variability, which may, as shown on the right-hand side of Fig. 6.2, increase with the amplitude (level) of the independent variable or may not be randomly distributed. In such cases the weighted least squares method, which is an extension of the ordinary least squares method, is preferable. In this extension, weights are associated with each data point, reflecting the degree of confidence in (or relative precision of) the individual measurements. Instead of minimizing the error $\varepsilon^T \varepsilon$ in Eq. (6.8), we now minimize $\varepsilon^T W \varepsilon$. Except for this change leading to the weighted sum of squares, we have the same set-up as in the previous case:

$$\begin{aligned} J(\theta) &= \frac{1}{2} \sum_{k=1}^N w_k \varepsilon^2(k) = \frac{1}{2} \varepsilon^T W \varepsilon \\ &= \frac{1}{2} [Y - X\theta]^T W [Y - X\theta] \end{aligned} \quad (6.37)$$

where W is the given weighting matrix, assumed symmetric positive definite. The gradient of the cost function with respect to the parameters is given by

$$\frac{\partial J(\theta)}{\partial \theta} = -Y^T W X + \theta^T X^T W X \quad (6.38)$$

Setting the gradient of $J(\theta)$ with respect to θ to zero, and taking the transpose we arrive at the modified *normal equation*:

$$(X^T W X) \hat{\theta} = X^T W Y \quad (6.39)$$

where $\hat{\theta}$ denotes the estimates of the true parameters θ . Assuming $(X^T W X)$ invertible and after simple rearrangement the estimates are obtained as

$$\hat{\theta} = (X^T W X)^{-1} X^T W Y \quad (6.40)$$

which can be solved in one shot through simple matrix operations.

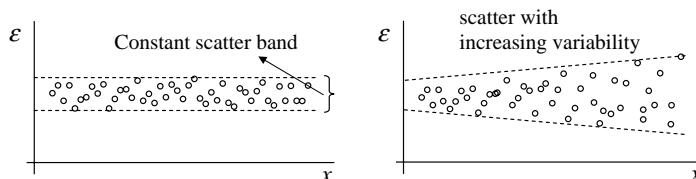


Fig. 6.2 Residuals from simple linear regression.

Another way to arrive at the WLS solution is through the transformation $\tilde{Y} = C^T Y$, where C is the Cholesky square root, that is, the lower triangular matrix, such that $W = CC^T$. Now, substituting Eq. (6.4) for Y in $\tilde{Y} = C^T Y$, we obtain⁵

$$\tilde{Y} = C^T X \theta + C^T \varepsilon \quad (6.41)$$

which can be rewritten in terms of the transformed variable $\tilde{X} = C^T X$ and transformed error $\tilde{\varepsilon} = C^T \varepsilon$ as

$$\tilde{Y} = \tilde{X} \theta + \tilde{\varepsilon} \quad (6.42)$$

Equation (6.42) pertains to the transformed variables, but otherwise is exactly in the same form as Eq. (6.4). Applying the ordinary least squares method, that is, Eq. (6.11), studied in Sec. II, the solution to Eq. (6.42) can be readily written down as

$$\hat{\theta} = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T \tilde{Y} \quad (6.43)$$

Now re-substitution of $\tilde{X} = C^T X$ and $\tilde{Y} = C^T Y$ in Eq. (6.43) and further minor simplification leads to the solution

$$\hat{\theta} = (X^T W X)^{-1} X^T W Y \quad (6.44)$$

which is the same as Eq. (6.40).

There is a minor issue that needs to be considered while applying the weighted least squares technique, because such an estimator is derived based on the basic assumption that the weights are known exactly. Based on the practical considerations presented in Sec. II.B, we are aware that such an assumption represents an ideal case, and in reality it is hard to meet. Therefore, the practical approach is to estimate the weights by applying some suitable procedure, hoping that the performance of the estimator is not significantly affected. The influence will be reduced when such weights are estimated from a large number of data samples. A common procedure is to choose the weights inversely proportional to the variance at each level of the independent variables.

The weighted least squares method is often called generalized least squares (GLS) method. We prefer the explicit terminology of WLS, because the other term GLS has also been used in other connotations such as total least squares, which we will address in Sec. V.

IV. Nonlinear and Multi-output Regression

In the regression equation that we had so far considered, the dependent variable y was linearly related to the independent variables x . Linear models are usually valid over a smaller range of excursions around an operating point. Models valid over a wider range may be necessary in some applications and

may involve a nonlinear response relationship. There are two approaches to handle such cases depending upon the type of nonlinearity in the postulated model. They are: 1) model being nonlinear in the independent variables, but linear in parameters; for example, $y = \theta_1 x_1 + \theta_2 x_2^2 + \dots$, and 2) model being nonlinear in the parameter itself, for example represented as $y = f(x, \theta)$, where f is a general nonlinear function.

In the first case, the problem can be transformed into a linear least squares problem by pre-computing the nonlinear terms and treating them as pseudo-independent variables. This is possible because the independent variables are available. To this problem in terms of new independent variables, the model then being linear in them, we can apply the classical least squares or weighted least squares methods we have already covered in Secs. II and III, respectively. This approach is found to work fairly well.

The second case of $y = f(x, \theta)$ is more complex and leads to a nonlinear regression problem. Although we have indicated the possibility of handling the first case through classical linear least squares approach, in its basic form it also really falls under the category of nonlinear regression problems. In such cases, making the same assumptions regarding the noise ε , the following cost function is minimized:

$$J_{\text{LS}} = \sum_{k=1}^N \{y(k) - f[x(k), \theta]\}^2 \quad (6.45)$$

Minimization of the cost function in Eq. (6.45) can be performed only iteratively. Thus, instead of the one-shot solution of Eq. (6.11), we now have to apply some iterative optimization procedure. Before going into the detail of such an algorithm we discuss yet another case, namely applying the equation error method to several dependent variables, which can also be more efficiently solved through an iterative optimization algorithm.

We recall from Eq. (6.1) that the dependent variable y is actually the i th variable, but we dropped the subscript i because each dependent variable was hitherto treated separately. Where we have more than one dependent variable, then for each of them it is necessary to form the vector Y and matrix X , and to solve Eq. (6.11) each time. In our specific case of flight vehicle system identification, we would in general wish to model the three force and three moment coefficients, which are the dependent variables. They are functions of the independent variables x comprising the aircraft motion variables like angular rates and flow angles as well as control surface deflections.

Since the aerodynamic force and moment coefficients depend on some variables from a set of independent variables, the six equations can be written in matrix form as

$$y(k) = \Theta x(k) + e(k) \quad k = 1, 2, \dots, N \quad (6.46)$$

where y is now a $n_y \times 1$ vector (e.g., $n_y = 6$ for aerodynamic modeling) and Θ contains the unknown parameters $[\theta_{11} \dots \theta_{1nq}; \dots; \theta_{ny1} \dots, \theta_{nynq}]^T$. The problem formulated in Eq. (6.46) is called the “multivariate multiple regression”

(MMR), which is typical of engineering applications. For the linear MMR case of Eq. (6.46) the matrix Y of the dependent variables is of the size $N \times n_y$. Assuming the same set of independent variables for each of the dependent variables, the matrix of the independent variables X becomes of the size $N \times n_q$. The $n_q \times n_y$ dimensional matrix of unknown parameters Θ can be estimated applying the one-shot procedure of Eq. (6.11). Although this is a viable procedure, it is seldom practiced, because treating the same set of independent variables for all the y introduces cross-coupling derivatives. Their estimation may not be reliable if adequate information is not available in X and Y . Furthermore, some of the other coefficients may not be physically realistic. The least squares estimates are commonly obtained by treating each of them separately, because the one-shot procedure is simple enough and can be easily repeated by appropriately choosing different independent variables for each dependent variable. The iterative procedure can be effectively applied to such a case and also to the nonlinear MMR, as discussed next.

In the case of nonlinear MMR, that is, when the model is nonlinear either in the parameters Θ or in the independent variables x , Eq. (6.46) can be rewritten as

$$y(k) = f[x(k), \Theta] + e(k) \quad k = 1, 2, \dots, N \quad (6.47)$$

Equation (6.47) is exactly in the form of an observation equation of the state space model postulated in Chapter 4, Eqs. (4.11) and (4.12). As such, we can apply the same iterative algorithm that was developed there for the output error method.¹⁴ The iterative procedure described in Chapter 4, Sec. V.B for the output error method caters for estimation of the covariance matrix of the residuals, which is used in the parameter-update step as a weighting matrix. Thus, application of the same algorithm to Eq. (6.47) consequently yields weighted least squares estimates. In other words, as already pointed out in Sec. II.C, algorithmically the only difference between the output error and weighted least squares estimation is the way we obtain the information about the states (independent variables); in one case it is by incorporating a suitable state estimator and in the other by using the measured states directly. The block schematic of the weighted least squares procedure is not explicitly shown, since it is very similar to that of the output error method, Fig. 4.1, with the exception of using the measured states.

V. Total Least Squares

It is pointed out in Sec. II.B that the ordinary least squares technique yields biased estimates in the presence of systematic errors and noise in the independent variables. Although there is no general procedure within the framework of regression techniques to account for systematic errors, the total least squares method allows the noise in the independent variables to be accounted for. It is also termed “orthogonal distance regression” (ODR) or “error-in-variable” (EIV) modeling. The differences between OLS and TLS are brought out best by considering the dependent variable as a function of a single independent variable. The OLS amounts to minimizing the difference along the y -axis (ordinate) between the model (regression equation) and the measured values of the

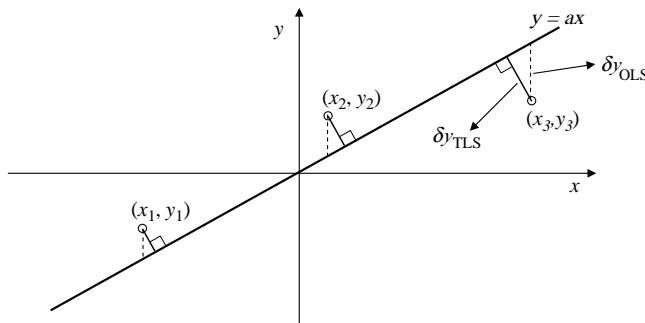


Fig. 6.3 Schematic of OLS and TLS principle.

dependent variables, that is, δy_{OLS} in Fig. 6.3. On the other hand, while considering errors in both, that is, in the dependent variables (along the y -axis) and the independent variables (along the x -axis), a clear assignment between them becomes difficult, and hence minimization of the orthogonal distance, which contains errors in both x and y , is more appropriate; see Fig. 6.3.

To account for noise in X , we rewrite the regression equation, Eq. (6.4), as:

$$Y = (X - \mu)\theta + \varepsilon \quad (6.48)$$

By multiplying out the terms on the right-hand side, by bringing all of them to one side, and by regrouping, Eq. (6.48) can be rewritten as

$$[[X \ Y] \ -[\mu \ \varepsilon]] \begin{bmatrix} \theta \\ -1 \end{bmatrix} = [\tilde{X} \ -\tilde{\Delta}] \begin{bmatrix} \theta \\ -1 \end{bmatrix} = 0 \quad (6.49)$$

where $\tilde{X} = [X \ Y]$ is the compounded data matrix of size $(N \times n_q + 1)$ formed by adjoining the column vector Y to the columns of X on the right, and $\tilde{\Delta} = [\mu \ \varepsilon]$ is the compounded noise vector. In the literature on TLS, the compounded matrix has been denoted as $[X|Y]$; we denote the same as $[X \ Y]$ which is more compatible with the computational implementation. The TLS solution to θ is obtained by solving Eq. (6.49).

Without going into detailed development, we look at the computational steps of obtaining TLS estimates, for which a simple linear solution as in Eq. (6.11) is not possible. The solution proposed by Golub and Van Loan based on the singular value decomposition is the most commonly applied approach to solving the TLS problem.¹⁵ Considerable advancements have been carried out by Van Huffel and Vanderwalle¹⁶ and others. The compounded matrix $[X \ Y]$ can be written as

$$[X \ Y] = U\Sigma V^T \quad (6.50)$$

where $\Sigma = [\text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{nq+1})]$ is the $(n_q + 1 \times n_q + 1)$ diagonal matrix of singular values such that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{nq+1}$; U and V are the left and right

singular matrices. Note that $[X \ Y]$ is of size $(N \times n_q + 1)$, but it is sufficient to compute only the first $n_q + 1$ columns of U corresponding to the first $n_q + 1$ nonzero singular values.

It can be shown that the smallest of the singular values $(\sigma_1, \sigma_2, \dots, \sigma_{nq+1})$, which happens to be the $n_q + 1$ th value (i.e., the last in the sorted list), corresponds to the minimum. The total least squares solution is then obtained from the column corresponding to the smallest singular value, that is, the last column of V and is given by:

$$\theta_{\text{TLS}} = -v/\lambda \quad (6.51)$$

where λ is the last element and v is the vector of the first n_q elements of the last column of V . Equation (6.51) corresponds alternatively to¹⁶

$$\theta_{\text{TLS}} = (X^T X - \sigma_{n+1}^2 I)^{-1} X^T Y \quad (6.52)$$

where σ_{nq+1} is the smallest singular value. Comparing Eq. (6.52) with Eq. (6.11), it becomes evident that the total least squares method incorporates a correction term to account for the noise in the independent variables. It has already been pointed out in Sec. II.B that the TLS estimates are unbiased, provided the noise μ is zero mean.

Although little more complex than the OLS solution, the classical TLS problem, as presented in Eqs. (6.49)–(6.52), can be solved analytically. Both the OLS and TLS assume $N > n_q$, that is, an overdetermined set of linear equations. TLS caters for noise in the data matrix X as well as in the observation vector Y , minimizing the sum of squares of residuals on all the (independent and dependent) variables in the equations and not just on the dependent variable.

The analytical solution of TLS can be extended to MMR having n_y observation (dependent) variables; see Sec. IV. In such a case, Y will be a matrix of size $N \times n_y$ instead of a column vector and the compounded matrix is $(N \times n_q + n_y)$ sized instead of $(N \times n_q + 1)$ sized for the case with a single dependent variable. It is assumed that the same set of independent variables X is used. The singular value decomposition of the compounded matrix $[X \ Y]$ is then given by:

$$[X \ Y] = U \Sigma V^T = [U_S \ U_N] \begin{bmatrix} \Sigma_S & 0 \\ 0 & \Sigma_N \end{bmatrix} \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}^T \quad (6.53)$$

where the matrices are partitioned into what is termed system subspace (subscript S) and a null subspace (subscript N). As in the previous case, it is sufficient to compute only the first $n_q + n_y$ columns of U corresponding to the first $n_q + n_y$ nonzero singular values. It turns out that the TLS solution is given by^{15,16}

$$\theta_{\text{TLS}} = -V_{12} V_{22}^{-1} \quad \text{if } n_q \geq n_y; \quad \text{else } \theta_{\text{TLS}} = (V_{11}^T)^{-1} V_{21} \quad (6.54)$$

For the commonly addressed problem with $n_q \geq n_y$, the partitioned matrices V_{12} and V_{22} required in the solution are of the sizes $(n_q \times n_y)$ and $(n_y \times n_y)$, respectively, corresponding to the singular values in the null subspace. Following the Matlab® notation, they are given by $V_{12} = V(1:nq, nq + 1:nq + ny)$ and $V_{22} = V(nq + 1:nq + ny, nq + 1:nq + ny)$. It is obvious that Eq. (6.51) is a special case of Eq. (6.54) for $n_y = 1$.

TLS is not to be confused with the WLS, which was discussed in Sec. III and assumes error-free independent variables and minimizes the distance along the y -coordinate as the weighted sum of squares of the errors to account for variations in the reliability of the dependent variables. Several extensions of TLS are possible, for example weighted TLS to include a weighting matrix based on the square root of the covariance matrix,^{17,18} and generalized TLS, but we do not go into the detail of such techniques, which can be found in Ref. 16. The TLS problem can also be solved using iterative techniques.^{18,19}

Applications of the TLS to flight vehicle system identification have been very few.^{17,20} Nevertheless, it offers an option to obtain better least squares estimates compared with the ordinary least squares procedure in the presence of noise. Since the procedure does not cater for systematic errors in the variables, it once again requires rigorous data preprocessing to eliminate such errors. In this sense, it has the same practical limitations as OLS. On the other hand the methods discussed in Chapters 4 and 5 are more general and allow simultaneous estimation of system parameters and systematic errors as well as noise statistics.

VI. Instrumental Variable Method

The ordinary least squares method developed in Sec. II assumes the equation errors (residuals) to be a white noise process, that is, random and uncorrelated. If the residuals are correlated, the least squares procedure yields biased estimates. In such cases, an instrumental variable (IV) method is applied to account for correlated noise. In this method the so-called instrument variables are introduced

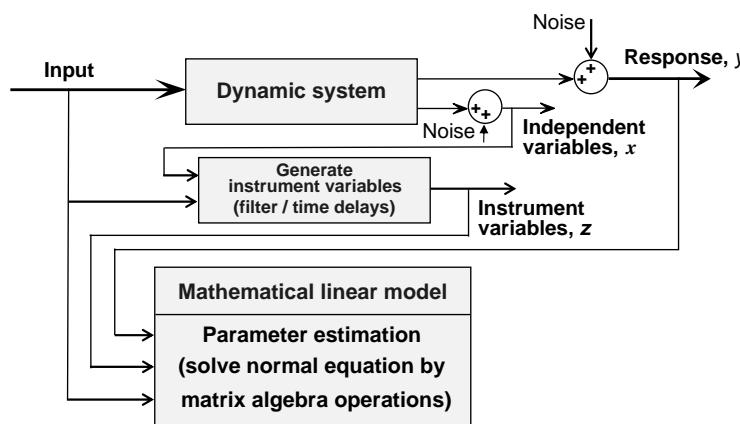


Fig. 6.4 Block schematic of instrumental variable method.

to cancel out the effect of correlated noise. Figure 6.4 shows a schematic of this method. It is necessary to choose signals $z = (z_1, \dots, z_{nq})^T$ such that $z(k)$ are strongly correlated with the independent variables $x(k)$, but are uncorrelated with an equation error $\varepsilon(k)$, that is

$$E\{z(k)x^T(k)\} \text{ is a positive definite matrix; } E\{z(k)\varepsilon(k)\} = 0 \quad (6.55)$$

Now, multiplying Eq. (6.6) by $z(k)$ and summing and averaging over N data samples yields

$$\frac{1}{N} \sum_{k=1}^N z(k)\varepsilon(k) = \frac{1}{N} \sum_{k=1}^N z(k)y(k) - \frac{1}{N} \sum_{k=1}^N z(k)x^T(k)\theta \quad (6.56)$$

From the assumption of $z(k)$ being uncorrelated with $\varepsilon(k)$, for large N we know that

$$\frac{1}{N} \sum_{k=1}^N z(k)\varepsilon(k) = 0 \quad (6.57)$$

which is the left-hand side of Eq. (6.56). Combining Eqs. (6.56) and (6.57), that is, equating the right-hand side of Eq. (6.56) to zero, and solving for θ yields the estimates

$$\hat{\theta} = \left[\frac{1}{N} \sum_{k=1}^N z(k)x^T(k) \right]^{-1} \left[\frac{1}{N} \sum_{k=1}^N z(k)y(k) \right] \quad (6.58)$$

which can be conveniently written in an equivalent matrix form as

$$\hat{\theta} = (Z^T X)^{-1} Z^T Y \quad (6.59)$$

where X and Y are as defined as before by Eqs. (6.4) and (6.5) and the matrix of instrument variables Z :

$$Z = \begin{bmatrix} z_1(1) & z_2(1) & \dots & z_{nq}(1) \\ z_1(2) & z_2(2) & \dots & z_{nq}(2) \\ \vdots & \vdots & \ddots & \vdots \\ z_1(N) & z_2(N) & \dots & z_{nq}(N) \end{bmatrix} \quad (6.60)$$

Equation (6.59) for estimates applying the instrumental variable method is similar in form to that of the least-squares solution of Eq. (6.11). The estimates from Eq. (6.59) can be obtained once again through simple matrix algebra operations. It can be shown that these estimates are asymptotically unbiased,²¹ but we do not go into these details. The major difficulty associated with the

instrumental variable method is, however, in finding an appropriate set of such variables z . Any choice that satisfies the conditions specified in Eq. (6.55) will yield unbiased estimates. Most of the procedures^{21,22} to generate z are based on some form of filter incorporating either a priori values of the unknown parameters θ or estimates obtained through a least squares procedure. Another possibility is to use time delayed values of the independent variables as instrument variables. Although the method yields unbiased estimates, it has not found widespread use in flight vehicle system identification, mainly because of the difficulty just discussed.

VII. Data Partitioning

Modeling of aerodynamic characteristics over a large range of values of the influencing independent variable, for example angle of attack, may be of interest in specific cases, for example fighter aircraft with high angle-of-attack capabilities. Estimation of aerodynamic parameters of linear models and from small amplitude maneuvers at such flight conditions applying the classical output error method may require a large number of flight maneuvers at several trim conditions. Moreover, it may not always be possible to trim the aircraft at some of these extreme flight conditions. On the other hand estimation of aerodynamic characteristics from a single large amplitude maneuver, such as sweeping the angle-of attack, will require complex nonlinear model postulates, whose form may not be known or anticipated. The estimation may pose several difficulties owing to an inadequate model structure. It is also likely that, at high angles of attack, stochastic vortex shedding at the aircraft nose may be dominant, necessitating an estimation method capable of accounting for process noise thereby further aggravating the difficulties. In these cases the least squares technique combined with "data partitioning," which will be discussed in the following, provides a simpler alternative approach.

The linear regression analysis, being based on linear models, may not be directly applicable to large-amplitude maneuvers. In such cases, the data partitioning is adopted to divide a maneuver that covers a large range of some variable into several portions, each of which spans a smaller range of that variable.^{10,23,24} To each segment the regression analysis can be applied to estimate the unknown derivatives. A schematic explanation of data partitioning is shown in Fig. 6.5. Typically, the angle of attack is chosen as the variable for data partitioning to select those data points from the complete maneuver for which the measured angle of attack is in the specified range, $\alpha_{\min} \leq \alpha \leq \alpha_{\max}$.

For the case depicted in Fig. 6.5 the range from 50 to 55 deg. of angle of attack is chosen to demonstrate the procedure. The upper part shows the complete flight maneuver covering the angle of attack from 20 deg. to roughly 65 deg. It is observed that there are six portions, marked 1–6, that span the chosen range of 50–55 deg. These six data segments are cut from the complete maneuver, see the middle part of the figure, and concatenated for evaluation purposes, as shown in the bottom part of Fig. 6.5. Principally, it would be possible to apply the output error or filter error method to these six time slices, because these methods provide a multirun evaluation capability. The disadvantage is, however, that the initial conditions will have to be estimated for each time

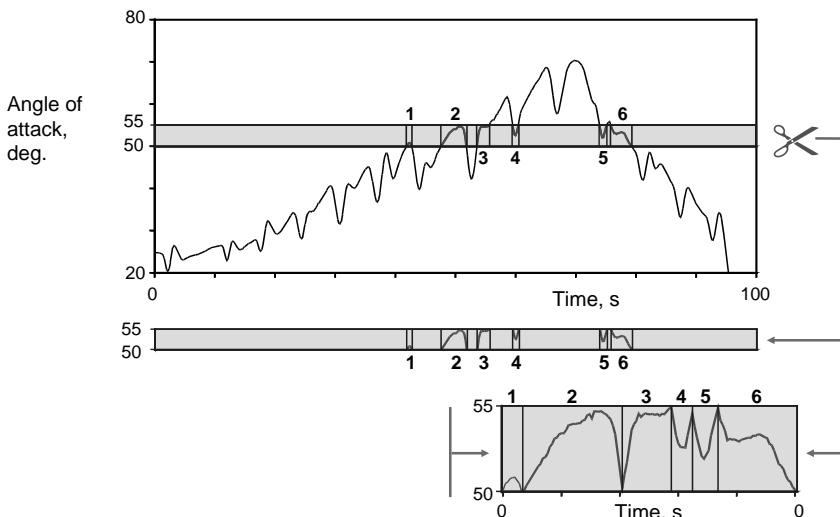


Fig. 6.5 Schematic of data partitioning.

segment. The short record lengths and large number of parameters pose convergence problems and mostly result in correlated estimates. On the other hand least squares techniques can be readily applied to such partitioned data sets, estimating the aerodynamic parameters directly without the need to estimate the initial conditions. This is particularly useful when different model structures and different data partitioning will have to be tried out. The procedure usually gives improved parameter estimates and reduced correlations.

Although the procedure of data partitioning has been illustrated based on a single parameter, in the present case the angle of attack, it is possible to partition the data considering multiple parameters simultaneously. Angle of attack and angle of sideslip is one such combination that may be useful in estimating linear models successively to arrive at a model valid for large ranges of these variables from large amplitude maneuvers.

Basically, "data partitioning" is a preprocessing step followed by parameter estimation applying the conventional linear regression method. This approach is feasible because the regression analysis does not rely on the temporal relation between the data points. The temporal relation defines the explicit sequential ordering in time of states/variables, defining the current states as some function of the previous states. It is fundamental to the state space model representation, in which the states evolve over time, but not for the least squares estimation. Processing of N data points in any arbitrary order of sequence will lead to the same least squares estimates.

VIII. Model Structure Determination

Model structure determination is one of the most widely addressed topics in the abundant literature on the regression analysis. There are several dedicated

publications addressing just this aspect. What we cover in this section is by no means exhaustive. Rather, we take a cursory glance at the general philosophy and methodology as well as at some of the evaluation criteria and various issues involved. Finally, we will take a pragmatic look at this approach as applicable to our interest of flight vehicle system identification, particularly in light of the fact that considerable a priori knowledge about the aerodynamic model structure is available in many of the cases.

The two typical problems encountered in the application of the hitherto discussed regression approach are: 1) some of the independent variables x might be slightly correlated with the dependent variable y , and 2) some of the independent variables may be redundant because they may be correlated with other independent variables (collinearity).

Such independent variables need to be detected to avoid their inclusion in the regression model, because the resulting inaccuracies due to their inclusion outweigh the increase in the model fidelity.²⁵ Collinearity is very often the cause of poor regression results, because while considering two variables, it shows strong influence on each when considered separately (as may be indicated by a large t -ratio, which is simply the ratio of estimated coefficient to its standard deviation), but a small t -ratio when both variables are included, implying large uncertainty in the estimates. Such estimates may show poor predictive capability when applied to other data sets. Hence, some procedure is necessary to overcome these practical difficulties, in order to include the most significant variables, so that the overall model has as few variables as possible, yet has adequate ability to effectively predict the dependent variable y .

Three basic approaches to model structure determination are: 1) forward selection, 2) backward elimination, and 3) stepwise regression. In the forward selection procedure, starting from the simplest of the models, containing just the bias (aerodynamic zero term), we add to the regression equation first an independent variable having highest correlation with the dependent variables. Subsequently, at each step contributions of each of the remaining variables are evaluated, and added to the model one at a time provided the corresponding t -ratio is greater than the specified threshold value. When none of the remaining variables have a t -ratio greater than a specified threshold, the process is terminated. Other statistical test criteria, for example partial F -statistics, which we will cover later in this section, have also been used instead of the t -ratio to determine the significance of parameter being entered. The forward selection is a unidirectional procedure, based purely on evaluating the t -ratio. Once a particular variable has been selected, the choice is final in the sense that the selection procedure does not evaluate the degradation that may result owing to subsequent entry of other variables.

The backward elimination, as the name implies, is the reverse process of forward selection. Here, we start with an oversized model, including all the potential variables in the model. At each step, the effect of dropping one variable is evaluated. If the corresponding t -ratio is smaller than the specified threshold and is the smallest of the t -ratios of all variables still in the model, the corresponding variable is then removed. If none of these two conditions are satisfied, the model retains the particular variable.

Stepwise regression combines forward selection and backward elimination. It consists of starting from a simplified model with just one or two primary independent variables, and then introducing other independent variables sequentially, evaluating the significance of the new addition by suitable statistical criteria. At each step, just like in the backward elimination procedure, all the parameters included are reassessed. It could happen that, owing to correlation among the variables, some of the variables introduced earlier in the model development process drop out later. Thus, the stepwise regression procedure enables determination of an appropriate ("best") subset of independent variables from the potential variables that have to be included in the regression equation, Eq. (6.1), to adequately model the observations. In other words, it helps to arrive at the parsimonious representation. In general, the stepwise regression may lead to a subset of variables that is different from the one obtained by forward or backward elimination procedure. Since stepwise regression is more commonly used for aerodynamic model structure determination, we study this procedure in more detail in the next subsection.

In the case of multivariate multiple regression, that is, when y is an $n_y \times 1$ vector, there are two possible approaches, 1) we treat the n_y -dependent variables one at a time, and perform the model selection procedure n_y times separately, and 2) perform the multivariate model selection procedure to arrive at a common set of predictors giving the best combination over all the n_y dependent variables. In the case of aerodynamic modeling, we know from the physics of the phenomenon that, in general, we come across the problem of the first type, because the six aerodynamic coefficients are functions of different sets of independent variables. As such, the simpler approach leading to n_y different subsets of predictors is more appropriate to our specific case.

A. Stepwise Regression

The stepwise regression procedure of variable selection can be summarized as follows:

- 1) To start with, define a set of potential independent variables; compute the correlations between each of the independent variables x_i and the output y ; choose the variable with the highest correlation as the first entry.
- 2) Add another variable from the remaining set with highest partial correlation coefficient.
- 3) Check partial F -values of all included variables; eliminate those variables having F -values below the prespecified threshold.
- 4) Jump back to step 2 to repeat the procedure until none of the remaining variables lead to model improvement.

Mathematically, the above four steps lead to the following computational procedure:

Step 1 Compute the correlation coefficients between y (dependent variable) and each independent variable x_i from the set of n_p potential variables:

$$\rho_i = r_{yx_i} = \frac{\sum_{k=1}^N (y(k) - \bar{y})(x_i(k) - \bar{x}_i)}{\sqrt{\sum_{k=1}^N [y(k) - \bar{y}]^2 \sum_{k=1}^N [x_i(k) - \bar{x}_i]^2}} \quad (6.61)$$

where the subscript i corresponds to the i th independent variable, and $\bar{y} = 1/N[\sum_{k=1}^N y(k)]$ and $\bar{x}_i = 1/N[\sum_{k=1}^N x_i(k)]$ denote the respective mean values.

Check whether the correlation coefficients ρ_i are significant or not (i.e., greater than a preselected threshold). If not, none of the potential variables can enter the model, yielding the mean as the best fit. Otherwise, select the independent variable, say the j th variable, with the highest correlation coefficient as the first entry and postulate the probable regression model as

$$y = \theta_0 + \theta_j x_j + \varepsilon \quad (6.62)$$

and proceed with the model of Eq. (6.62).

Step 2 Compute the partial correlations $r_{yx_i*x_j}$, $i \neq j$, for each of the remaining independent variables x_i , ($i = 1, \dots, j-1, j+1, \dots, n_p$). The procedure to compute the partial correlations with one or more variables held fixed is described subsequently.

Choose an independent variable with the largest partial correlation, say the k th independent variable. Then fit the appended model:

$$y = \theta_0 + \theta_j x_j + \theta_k x_k + \varepsilon \quad (6.63)$$

Step 3 Now, compute the partial F -values F_j and F_k corresponding to the j th and k th variable. Comparing the smallest of F_j and F_k with the preselected threshold F_{Th} leads to two possibilities: 1) $\min(F_j, F_k) < F_{Th}$, suggesting that the contribution due to the corresponding variable is not significant; accordingly, delete either the j th or k th independent variable corresponding to the smaller of them; jump back to step 2 and continue with the remaining variables; 2) $\min(F_j, F_k) > F_{Th}$, suggesting that the k th variable has a significant contribution to y , and hence can be retained in the extended model of Eq. (6.63); jump back to step 2, and proceed to examine the partial correlations of y on the remaining variables, given j and k are in the model.

At each step, these two sub-steps may lead to either increasing or decreasing the variables included in the model. The partial correlations will have to be computed keeping multiple variables already entered in the model held fixed. The procedure automatically stops when the variables once entered in the model cannot be removed and when addition of any new variables does not lead to improvement in the model fit.

B. Statistical Test Criteria

The correlation coefficient defined in Eq. (6.61) lies in the range $[-1, 1]$; it is an indicator of the degree of influence of the particular independent variable on the dependent variables (observation). The higher the correlation coefficient is, the larger the dependence of y on that particular independent variable. The correlation coefficient defined in Eq. (6.61) is, however, not to be confused with the correlation coefficient between the parameters that we had discussed in connection with the properties of parameter estimates in Sec. XIV in Chapter 4.

Partial correlation determines, in general, the correlation between any two of the variables, if the other variables are held constant. In our case, it is an indicator of the degree of influence of the particular independent variable, say x_i , on the dependent variables y , when all of the x except x_i are held fixed. In other words, it predicts the influence of the variables which are held fixed on the other two variables, that is, on y and x_i . If partial correlation approaches zero, it implies that there is no significant link between the two variables being tested, because the control variables (i.e., variables held fixed) are the primary or intervening variables. Hence, the partial correlation can be used as a measure of relative importance of each independent variable in the regression equation.

Given any three variables y , x_i , and x_j , computation of the partial correlation keeping just one variable fixed is fairly straightforward. In this case the partial correlation can be computed from the simple correlation coefficient:

$$r_{yx_i * x_j} = \frac{r_{yx_i} - r_{yx_j} r_{x_i x_j}}{\sqrt{(1 - r_{yx_j}^2)(1 - r_{x_i x_j}^2)}} \quad (6.64)$$

where the correlation coefficients r_{yx_i} , r_{yx_j} , and $r_{x_i x_j}$ are computed from the expression given in Eq. (6.61) replacing y and x_i with appropriate combinations of y , x_i , and x_j . However, when there are multiple variables entered in the model, computation of the partial correlation for the rest, keeping those already in the model fixed, is somewhat tedious. Basically, it is based on calculating 1) the residuals after fitting y using entered variables (representing the part that has not been so far explained), and 2) the residuals after fitting the candidate variable x_i using the same set of entered variables (representing the part of x_i that cannot be predicted by the entered variables). The correlation between these two residuals gives the desired partial correlation. To describe the process mathematically, for the general model postulated in Eq. (6.1), the partial correlation of x_j on y , denoted by $r_{yx_j * (x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_{nq})}$, is computed as follows. First we fit the model by dropping the j th independent variable:

$$\begin{aligned} y_{(-j)} &= \theta_1 x_1(k) + \dots + \theta_{j-1} x_{j-1}(k) \\ &\quad + \theta_{j+1} x_{j+1}(k) + \dots + \theta_{nq} x_{nq}(k) + \varepsilon(k); \quad k = 1, 2, \dots, N \end{aligned} \quad (6.65)$$

where $y_{(-j)}$ denotes the independent variable y , but without the contribution due to the j th independent variable, as evident from the term $\theta_j x_j(k)$ missing on the

right-hand side of Eq. (6.65). Note specifically the notation $y_{(-j)}$; it is not the j th element of y ; we are dealing here with a single dependent variable.

Now fit a similar model to the dropped, that is, j th, independent variable using the same set of independent variables as in Eq. (6.65):

$$x_j = \theta_1 x_1(k) + \cdots + \theta_{j-1} x_{j-1}(k) + \theta_{j+1} x_{j+1}(k) + \cdots + \theta_{nq} x_{nq}(k) + \varepsilon(k); \quad k = 1, 2, \dots, N \quad (6.66)$$

Denoting the fit errors in the two cases resulting from the model fit of Eqs. (6.65) and (6.64) as $e^{y_{(-j)}}$ and e^{x_j} , respectively; the partial correlation is then given by

$$r_{yx_j*}(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_{nq}) = \frac{\sum_{k=1}^N [e^{y_{(-j)}}(k) - \bar{e}^{y_{(-j)}}][e^{x_j}(k) - \bar{e}^{x_j}]}{\sqrt{\sum_{k=1}^N [e^{y_{(-j)}}(k) - \bar{e}^{y_{(-j)}}]^2 \sum_{k=1}^N [e^{x_j}(k) - \bar{e}^{x_j}]^2}} \quad (6.67)$$

where $\bar{e}^{y_{(-j)}} = 1/N \sum_{k=1}^N e^{y_{(-j)}}(k)$ and $\bar{e}^{x_j} = 1/N \sum_{k=1}^N e^{x_j}(k)$ are the respective mean values of the residuals.

The F -value used in the stepwise regression procedure is a measure of the significance of the independent variables and relates the goodness of fit (i.e., the value of the cost function) to the fit error, weighted by the degrees of freedom in the data (i.e., the number of independent samples from which the variances is calculated). It can be computed from the partial correlations as follows:

$$F = \frac{r_{yx_j*} N - n_q}{(1 - r_{yx_j*}) n_q - 1} \quad (6.68)$$

If more than one independent variable is held fixed, then the partial correlation $r_{yx_i*}(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_{nq})$ is to be used to compute the corresponding F -values.

Some other useful criteria are the coefficient of determination (R^2), the adjusted coefficient of determination, Mallow's C_m statistics, and Akaike's information criterion. We give a brief statement on the first two of these additional tests. The coefficient of determination (R^2) is given by

$$R^2 = \frac{\sum_{k=1}^N [\hat{y}(k) - \bar{y}]^2}{\sum_{k=1}^N [y(k) - \bar{y}]^2} = \frac{\hat{\theta}^T X^T Y - N\bar{y}^2}{Y^T Y - N\bar{y}^2} \quad (6.69)$$

where \hat{y} represents the predicted output (dependent variable) using the estimated model parameters $\hat{\theta}$ and y the measured data samples, and, as already defined, \bar{y} is the mean value. As the number of estimated parameters (elements of $\hat{\theta}$) will vary

from step to step, R^2 statistics give the proportion of the variation in the dependent variable y that is explained by the regression model, and has a value between 0 and 1. The larger the value of R^2 , the better the model fit is. However, it tends to be overestimated when the number of samples N is not large compared with n_q , the number of independent variables. Furthermore, for the same number of data samples, R^2 always increases with increasing the number of independent variables, even when some of them may not have sufficient influence to justify their inclusion. This may not lead to a parsimonious model representation.

The adjusted coefficient of determination, $\text{Adj}R^2$, is a modified version that corrects Eq. (6.69) for the number of parameters being estimated and does not necessarily increase when we add extra variables:

$$\text{Adj}R^2 = 1 - (1 - R^2) \frac{N - 1}{N - n_q - 1} \quad (6.70)$$

C. Practical Aspects

Although we have covered in some depth the stepwise regression procedure as a possible means of determining model structure, the actual procedure is plagued by some of the common problems that we encounter in system identification. It may look quite easy and promising at a first glance, but for complex problems with a large number of independent variables and complex nonlinear relations, it requires some skill and judgment to arrive at the final model. Amongst the difficulties associated with the model structure determination using stepwise regression, the four most important ones are:

- 1) In the presence of data collinearity the variable selection process is inefficient and inconsistent.
- 2) The R^2 statistics are usually over estimated.
- 3) The threshold values used to judge the statistical significance are situational, that is, they may vary from case to case.
- 4) If two data sets at almost similar operating conditions differ in variability, the selected variable may not turn out to be a unique set.

The choice of the potential independent variables, including all possible cross combinations, has to be pre-specified. An inadvertent omission of a particular primary influencing variable may not be detected, leading to a good match of the data fitted through combinations of other variables. Most likely, such models will not show good predictive capabilities for data not used in fitting the model. The specification of a partial F -statistics threshold to accept a particular variable will also affect the final model.

Furthermore, the best choice of variables is usually guided by physical understanding and good judgment of prime variables of the process. From this aspect we quote Judd and McClelland's observation on the stepwise regression procedure:²⁶

It is our experience and strong belief that better models and a better understanding of one's data result from focused data analysis, guided by substantive theory.

Although the observation pertains to a completely different phenomenon being analyzed, it brings out some concern that one should keep in mind while applying such procedures. For flight vehicles, the a priori information about the typical structure of aerodynamic forces and moments is usually available, particularly when we try to estimate linear models valid for small excursions around the trim point. In such cases the advantages of stepwise regression are minimal. Furthermore, the commonly used set of potential independent variables, like $\alpha, \beta, p, q, r, \delta_e, \delta_a, \delta_r$ and their various combinations such as $\beta\alpha, \alpha\delta_e$, may not be quite adequate when the model is nonlinear within the range of variable excursions. We will come across such a case in Chapter 12, Sec. VIII.A showing $C_{n\beta}$ to be nonlinearly dependent on the angle of attack. The efficacy of straightforward model structure determination, without additional considerations such as data partitioning, is not evident in such cases. In addition, the least squares method and the stepwise regression approach have certain limitations which have already been discussed.

Based on some of these practical considerations, we generally prefer to use a more general approach to modeling based on 1) incorporation of the basic a priori knowledge, which we mostly possess, about the modes of aircraft motion, aerodynamic effects, and associated model structure, and 2) estimation of parameters applying more powerful methods such as the output error method described in Chapter 4. In special cases like modeling at high angles of attack and unstable aircraft, however, stepwise regression combined with the option of data partitioning can be useful. In such cases, techniques to detect collinearity, and procedures to account for them in the least squares estimation, are necessary. Such procedures will be covered in Chapter 9, dealing specifically with unstable aircraft identification.

IX. Examples

Since we are concerned in this book with aircraft parameter estimation, we demonstrate the least squares method to estimate the stability and control derivatives from flight data. To demonstrate the various aspects studied in this chapter, it is sufficient to analyze a single operating point based on simple model. A more complex application is presented in Chapter 12.

A. Flight-derived Aerodynamic Force and Moment Coefficients

Although the main advantage of the regression analysis is its simplicity, a fair amount of computation during a preprocessing step is required in the case of aircraft parameter estimation, because the aerodynamic forces and moments (i.e., the dependent variables) are not directly measured, but have to be obtained from the measurements of the related variables, namely linear accelerations and angular rates. Actually, one needs angular accelerations, but in practice in most of the cases angular rates are measured and not the angular accelerations. It is also possible to treat the measured linear and angular accelerations as dependent variables, in which case the coefficients pertaining to the aerodynamic forces and moments will have to be obtained through proper transformations after the estimation. Generally the first approach of precomputing the force and moment coefficients is preferred. A block schematic of the computational procedure presented is depicted in Fig. 6.6.

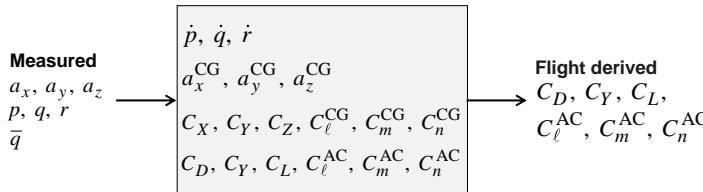


Fig. 6.6 Schematic of data preprocessing to compute aerodynamic force and moment coefficients from flight measured data.

Considering the flight vehicle as a point mass, we know that the resultant aerodynamic forces and moments act at the center-of-gravity (CG). Accordingly, to apply the regression method, one needs to know the linear acceleration at the CG. Owing to the restrictions of the mechanical mountings, it is not always possible to mount the sensors exactly at the CG; they are usually located at some convenient location, away from the CG. Moreover, the CG location depends on the loading conditions and may vary from flight to flight. The actual CG location is, however, computed from the flight loading manifest, and thus the actual distances from the CG to the sensor assembly can be determined accurately. If the measurement of actual fuel quantity (or alternatively of fuel consumption rate) is available, the effects of variations in the total fuel quantity during the flight on the CG location can also be accounted for. Depending upon the specific application, it may also be necessary to account for fuel slosh effects. Since the accelerometers mounted away from the CG measure local accelerations which contain the components due to the angular rates and accelerations, it is necessary to obtain the linear accelerations at CG.

The body-fixed accelerations ($a_x^{CG}, a_y^{CG}, a_z^{CG}$) at the CG are computed from the translational accelerations ($a_{xm}^{AS}, a_{ym}^{AS}, a_{zm}^{AS}$) measured at the accelerometer sensor position (AS) through the following transformation:

$$\begin{aligned} a_x^{CG} &= a_{xm}^{AS} + (q^2 + r^2)x_{ASCG} - (pq - i)y_{ASCG} - (pr + \dot{q})z_{ASCG} \\ a_y^{CG} &= a_{ym}^{AS} - (pq + \dot{r})x_{ASCG} + (p^2 + r^2)y_{ASCG} - (qr - \dot{p})z_{ASCG} \\ a_z^{CG} &= a_{zm}^{AS} - (pr - \dot{q})x_{ASCG} - (qr + \dot{p})y_{ASCG} + (p^2 + q^2)z_{ASCG} \end{aligned} \quad (6.71)$$

where (p, q, r) are the measured angular rates, $(\dot{p}, \dot{q}, \dot{r})$ the angular accelerations obtained by numerical differentiation of the measured angular rates, and $(x_{ASCG}, y_{ASCG}, z_{ASCG})$ the position of the AS with respect to the CG along the three body-fixed coordinate axes, respectively.

Having obtained the linear accelerations at the CG, the three body axes aerodynamic force coefficients are given by:

$$\begin{aligned} C_X &= (ma_x^{CG} - F_{eng} \cos \sigma_{eng}) / (\bar{q}S) \\ C_Y &= (ma_y^{CG}) / (\bar{q}S) \\ C_Z &= (ma_z^{CG} + F_{eng} \sin \sigma_{eng}) / (\bar{q}S) \end{aligned} \quad (6.72)$$

where m is the aircraft mass, F_{eng} the total thrust, σ_{eng} the inclination angle of the engines, \bar{q} the measured dynamic pressure, and S the reference area.

The lift, drag and side force coefficients are derived from the body axes force coefficients C_X, C_Y, C_Z as follows:

$$\begin{aligned} C_L &= C_X \sin \alpha - C_Z \cos \alpha \\ C_D &= -C_X \cos \alpha - C_Z \sin \alpha \\ C_Y &= C_Y \end{aligned} \quad (6.73)$$

Computation of the three moment coefficients is a little more involved and requires the angular accelerations $(\dot{p}, \dot{q}, \dot{r})$. As already pointed out, since angular accelerations are not directly measured, they have to be obtained by numerical differentiation of the measured angular rates (p, q, r) . Numerical differentiation may introduce some additional noise due to discretization errors and noise in the measured rates. Several techniques of numerical differentiation are readily available, and hence not elaborated here further.

The body-axes rolling, pitching, and yawing coefficients, C_ℓ, C_m, C_n referred to CG are given by:

$$\begin{aligned} C_\ell^{\text{CG}} &= [I_X \dot{p} - I_{XZ} \dot{r} - I_{XZ} pq - (I_Y - I_Z)qr \\ &\quad - \Delta F_{\text{eng}} \sin \sigma_{\text{eng}} y_{\text{ENCG}}]/(\bar{q}Sb) \\ C_m^{\text{CG}} &= I_Y \dot{q} + I_{XZ}(p^2 - r^2) - (I_Z - I_X)pr \\ &\quad - F_{\text{eng}} \cos \sigma_{\text{eng}} z_{\text{ENCG}} - F_{\text{eng}} \sin \sigma_{\text{eng}} x_{\text{ENCG}}]/(\bar{q}Sl) \\ C_n^{\text{CG}} &= [I_Z \dot{r} - I_{XZ} \dot{p} + I_{XZ} qr - (I_X - I_Y)pq \\ &\quad - \Delta F_{\text{eng}} \cos \sigma_{\text{eng}} y_{\text{ENCG}}]/(\bar{q}Sb) \end{aligned} \quad (6.74)$$

where I_X, I_Y , and I_Z are the moments of inertia about the x, y, z axes, respectively, I_{XZ} is the cross moment of inertia, ΔF_{eng} is the differential thrust (left minus right engine thrust), y_{ENCG} the lever arm (distance of the right engine from the centerline along the y -axis), $(x_{\text{ENCG}}, z_{\text{ENCG}})$ the position of the engines with respect to the CG along the x - and z -axes, b the reference length for the lateral-directional coefficient, and l the reference length for the longitudinal coefficient. All other quantities have been previously defined. The reference lengths for the lateral-directional and longitudinal coefficients are usually the wing span (or half the wing span) and the wing mean aerodynamic chord, respectively. In a few specific cases the mean aerodynamic chord is used for both the modes. What is actually used as the reference length is not really important, but has to be kept in mind when comparing derivatives and when using the identified aerodynamic model. The aerodynamic model contains not only the equations for the coefficients, but also the application rules (containing the numerical values of the reference areas and lengths) to calculate the forces and moments. This is one of the common sources of error that is often overlooked, when results from different sources are compared or combined together.

The moment coefficients referred to the aerodynamic center (AC), also some time called moment reference point, are now computed from Eqs. (6.74) and (6.72). They are given by

$$\begin{aligned} C_{\ell}^{\text{AC}} &= C_{\ell}^{\text{CG}} - C_Z \frac{y_{\text{ACCG}}}{b} + C_Y \frac{z_{\text{ACCG}}}{b} \\ C_m^{\text{AC}} &= C_m^{\text{CG}} - C_X \frac{z_{\text{ACCG}}}{l} + C_Z \frac{x_{\text{ACCG}}}{l} \\ C_n^{\text{AC}} &= C_n^{\text{CG}} - C_Y \frac{x_{\text{ACCG}}}{b} + C_X \frac{y_{\text{ACCG}}}{b} \end{aligned} \quad (6.75)$$

where (C_X, C_Y, C_Z) are the force coefficients given by Eq. (6.72), and $(x_{\text{ACCG}}, y_{\text{ACCG}}, z_{\text{ACCG}})$ is the position of the AC with respect to the CG.

It is obvious that the above data preprocessing step involves several important aspects, like computation of exact CG location, accounting for sensor offsets, the engine thrust, and so on. An astute reader will realize that these steps are mostly in the reverse order of those which are required while identifying the model applying the output error method. Thus, we do not really avoid the computations, except that they are performed at different stages of the algorithm being applied. In the output error method the aerodynamic model is embedded in the equations of motion and constitutes a part of the postulated state space model and the computations carried out using the integrated states. Depending upon the numerical integration method, they will be repeated at each data point a number of times. This contributes significantly to the computational burden of the output error method. On the other hand, while using the regression technique, there are no state equations and no integrations; the forces and moments are computed *a priori* from the measurements of the corresponding variables. This data preprocessing step needs to be done only once. Combined with the use of measured states in the estimation, it leads to a significant reduction in the computational burden.

Although not considered explicitly in Eqs. (6.71)–(6.75), it is straightforward to see that any errors in the measurements of the linear accelerations, angular rates, and dynamic pressure will directly affect the computed force and moment coefficients. These flight-derived coefficients $C_L, C_D, C_Y, C_{\ell}, C_m, C_n$ are the dependent variables to be modeled to estimate the stability and control derivatives. Any errors in these flight-derived coefficients will directly affect the estimates of the unknown aerodynamic parameters. Although the LS estimation accounts for noise in the dependent variables, we have to eliminate the systematic errors in them, such as bias, scale factor, or time delays. Elimination of such errors in these dependent variables as well as those in the independent variables (comprising angular rates, flow angles, and control surface deflections) is critical to the next step of parameter estimation.

B. Estimation of Aerodynamic Parameters of a Linear Model

For demonstration purposes we consider here analysis of flight data recorded with the research aircraft ATTAS. Three flight maneuvers performed at a

nominal altitude of 16,000 ft and at 200 kts are considered. The first maneuver is a multistep elevator input exciting the short period motion, the second is an aileron input resulting in the bank-to-bank motion, and the third is a rudder doublet input exciting the Dutch roll motion. The three maneuvers are 25, 30, and 30 s long, respectively. The same set of three maneuvers is analyzed in Chapter 10, Sec. V to check the kinematic consistency of the recorded flight data. The results of data consistency checking generated in that section (test_case = 22) showed the following biases in the measurements of linear accelerations and angular rates:

$$\begin{aligned}\Delta a_x &= 0.0092 \text{ m/s}^2 & \Delta a_y &= 0.00021 \text{ m/s}^2 & \Delta a_z &= 0.0960 \text{ m/s}^2 \\ \Delta p &= 0.0011 \text{ rad/s} & \Delta q &= 0.0033 \text{ rad/s} & \Delta r &= 0.0010 \text{ rad/s}\end{aligned}$$

The measurements of these variables are corrected for the corresponding biases before they are used in the data preprocessing step described in Sec. IX.A and for parameter estimation in the following. The engine thrusts F_{eL} and F_{eR} due to the left and right engines were computed from the measured engine parameters and stored in the flight data files along with the other measured data.

The models for aerodynamic force and moment coefficients are now postulated as follows:

$$\begin{aligned}C_D &= C_{D0} + C_{D\alpha}\alpha \\ C_L &= C_{L0} + C_{L\alpha}\alpha \\ C_m &= C_{m0} + C_{m\alpha}\alpha + C_{mq}q^* + C_{m\delta_e}\delta_e \\ C_Y &= C_{Y0} + C_{Y\beta}\beta \\ C_\ell &= C_{\ell0} + C_{\ell\beta}\beta + C_{\ell p}p^* + C_{\ell r}r^* + C_{\ell\delta_a}\delta_a \\ C_n &= C_{n0} + C_{n\beta}\beta + C_{np}p^* + C_{nr}r^* + C_{n\delta_r}\delta_r\end{aligned}\tag{6.76}$$

where δ_e is the elevator deflection, δ_a the aileron deflection, δ_r the rudder deflection, α the angle of attack, β the angle of sideslip, (p^*, q^*, r^*) the normalized angular rates, and $(C_{D0}, C_{L0}, C_{m0}, C_{Y0}, C_{\ell0}, C_{n0})$ the unknown aerodynamic parameters. The normalized angular rates are given by

$$p^* = pb/V \quad q^* = ql/V \quad r^* = rb/V\tag{6.77}$$

where V is the true airspeed; other quantities have been defined previously. We identify the unknown aerodynamic derivatives applying the LS method in both the forms, namely 1) iterative solution elaborated in Sec. IV, using the OEM software developed in Chapter 4, and 2) classical one-shot OLS and TLS solutions using matrix algebra operations, discussed in Secs. II and V, respectively.

To apply the iterative OEM program “/FVSysID/chapter04/ml_oem.m,” we first carry out the data preprocessing step described in Sec. IX.A and then provide the details for parameter estimation in the model definition function “mDefCase23.m” and set the integer flag test_case = 23.

The input and output variables for the data preprocessing step to compute the aerodynamic force and moment coefficients are given by:

Input variables

ZAccel	a_x, a_y, a_z, p, q, r
Uinp	$\delta_e, \delta_a, \delta_r, p, q, r, \alpha, \beta, V, \rho, F_{eL}, F_{eR}, \dot{\alpha}, Ma$

Output variables

Z	$C_D, C_L, C_m, C_Y, C_\ell, C_n$
---	-----------------------------------

where ρ is the density of air, F_{eL} the left engine thrust, and F_{eR} the right engine thrust. The dynamic pressure required in the precomputation of the force and moment coefficients is given by $0.5 \rho V^2$. The angular accelerations $\dot{p}, \dot{q}, \dot{r}$ required to compute the moment coefficients C_m, C_ℓ, C_n are obtained by numerical differentiation of measured angular rates p, q, r . The input array Uinp contains variables that are required during both the steps of preprocessing and parameter estimation. Not all signals from Uinp are required in the present case, but for the sake of similarity with the other cases addressed in other chapters it is made oversized.

The data to be preprocessed and the input variables, that is, the arrays ZAccel(Ndata,Ny) and Uinp(Ndata,Nu), are loaded from three data files:

```
load ..\f1t_data\fAttasElv1;
load ..\f1t_data\fAttasAil1;
load ..\f1t_data\fAttasRud1;
```

It is of course necessary to provide the input and output variables in proper units. Since we are analyzing three maneuvers simultaneously, we additionally define:

Nzi = 3	number of time segments
izhf = [Nts1; Nts1 + Nts2; Nts1 + Nts2 + Nts3]	cumulative index

where Nts1, Nts2, and Nts3 are the number of data points in the three individual time segments and izhf is the cumulative index at which the maneuvers end when they are concatenated.

Having defined the necessary details, the aerodynamic force and moment coefficients are computed according to Eqs. (6.71)–(6.75) by calling the data preprocessing function “umr_reg_attas.m”:

```
[Z, Uinp] = umr_reg_attas(ZAccel, Uinp, Nzi, izhf, dt, test_case);
```

Using the flight derived aerodynamic coefficients now available from the foregoing pre-processing step in array Z, and the input in the array Uinp, we now formulate the model for parameter using OEM method as follows:

	No. of variables	Function name
States	0	—
Outputs	6	$C_D, C_L, C_m, C_Y, C_\ell, C_n$
Inputs	14	$\delta_e, \delta_a, \delta_r, p, q, r, \alpha, \beta, V, \rho, F_{eL}, F_{eR}, \dot{\alpha}, Ma$

As there are no state equations, we do not actually need a function to code the right-hand sides of the state equations, but since we propose to use here the same algorithm that was developed for the output error method, to avoid problems associated with the number of arguments being passed, we either supply a dummy function, called “*xdot_attas_reg*,” in which there are no computations, or alternatively we define the function name for state equations as an empty string, as will be done here. The right-hand sides of the regression equation, namely Eq. (6.76), are coded in “*obs_attas_regLonLat.m*.” Thus, the model definition for this example provided by the function “*/FVSysID/chapter04/mDefCase23.m*,” which is called from the main program “*/FVSysID/chapter04/ml_oem*,” is as follows:

```
Test_case = 23; % test case number
state_eq = [ ]; % empty string for state equations
obser_eq = 'obs_TC23_attas_regLonLat'; % function for observation equations
Nx = 0; % number of states
Ny = 6; % number of observation variables
Nu = 14; % number of input variables
Nparam = 20; % total number of parameters
dt = 0.04; % sampling time
```

We specify zero starting values for all 20 parameters and run the program “*/FVSysID/chapter04/ml_oem.m*.” The determinant of the covariance matrix of the residuals (cost function) corresponding to these starting parameter values is 2.3022×10^{-20} . The iterative procedure is terminated after two iterations with the cost function value of 5.7047×10^{-33} . Actually, the model being linear in parameters and in the independent variables, the minimum is reached after the first iteration, but the iterative procedure needs another iteration to check whether or not convergence has been achieved.

The control surface deflections ($\delta_e, \delta_a, \delta_r$), angular rates (p, q, r) and flow angles (α, β), which are inputs to the model (independent variables), are shown in Fig. 6.7. The variation in the flow angles about the trim values is of the order of ± 3.3 and ± 4 deg., respectively. A comparison of the flight derived (i.e., measured) force and moment coefficients with those estimated by the model postulated in Eq. (6.76) is shown in Fig. 6.8; it is observed that the linear model captures the major cause-effect relations nicely. Little correlation was observed between the parameter pairs ($C_{D0}, C_{D\alpha}$) and ($C_{L0}, C_{L\alpha}$), which is attributed to the fact that we analyzed just one time record pertaining to the longitudinal motion. Moreover, the drag derivatives are better estimated from longer duration maneuvers, including phugoid motion. Nevertheless, the example is adequate to demonstrate the procedure of applying the iterative regression method to flight data, to show the use of the software that was developed in Chapter 4, and to demonstrate the two-step approach consisting of data pre-processing and followed by LS parameter estimation.

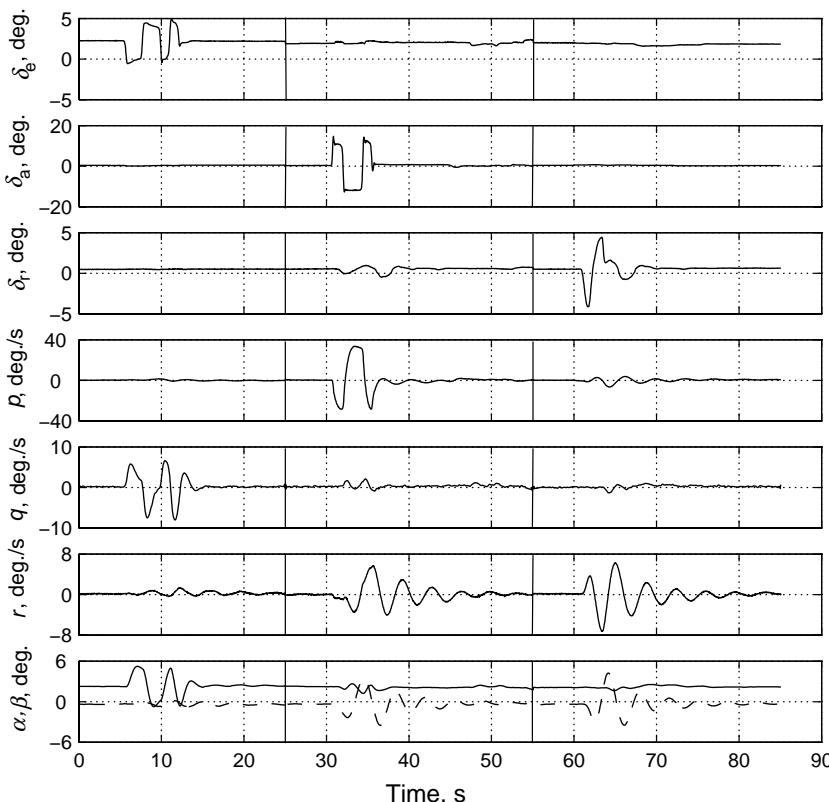


Fig. 6.7 Time histories of independent variables.

To this linear regression model we also apply the classical one-shot least squares method. The data preprocessing step is the same, but the estimates are now obtained from Eq. (6.11) using matrix algebra operations, that is, by solving the normal equation. The one-shot procedure is applied six times considering the dependent variables one at a time; it requires the output vector Y and data matrix X to be built appropriately each time, constituting different sets of variables. As summarized in Table 6.1, the data matrix is of different size each time, as it depends upon the number of independent variables appearing in each regression equation being estimated; see Eq. (6.76).

As already pointed out in Sec. II, the procedure allows estimation of aerodynamic zero terms by simply associating a vector of all ones as an additional independent variable. This is denoted by “1” in Table 6.1 and it is sometimes called a bias vector.

We run this case by calling the program “/FVSysID/chapter06/attas_regLS,” which is a very rudimentary implementation of the OLS method. There are many

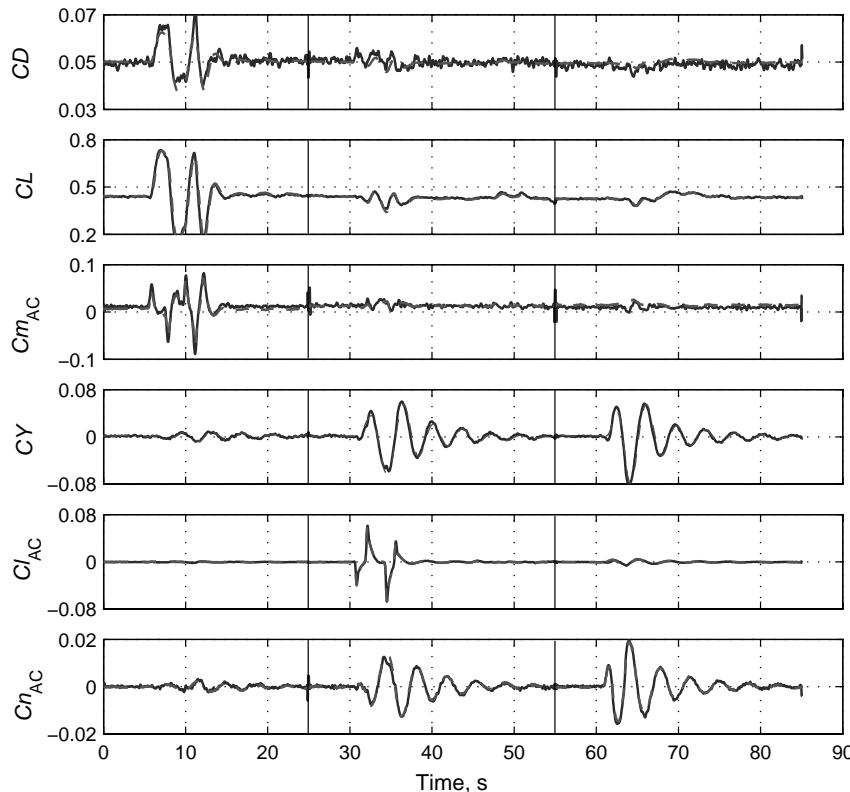


Fig. 6.8 Time histories of measured and estimated outputs (—, flight derived; - - -, model estimated).

elegant, sophisticated least squares programs available, providing more flexibility. The basic principle behind them, however, remains the same. The simple version is quite sufficient to trace the procedure elaborated in Sec. II and to compare the results with the iterative version. To run this program, we

Table 6.1 Summary of variables for one-shot least squares estimation

Coefficient of:	Dependent variable (Y)	Independent variables (X)	Parameters θ
Drag	C_D	$1, \alpha$	$C_{D0}, C_{D\alpha}$
Lift	C_L	$1, \alpha$	$C_{L0}, C_{L\alpha}$
Pitching moment	C_m	$1, \alpha, q^*, \delta_e$	$C_{m0}, C_{m\alpha}, C_{mq}, C_{m\delta e}$
Side force	C_Y	$1, \beta$	$C_{y0}, C_{y\beta}$
Rolling moment	C_ℓ	$1, \beta, p^*, r^*, \delta_a$	$C_{\ell 0}, C_{\ell \beta}, C_{\ell p}, C_{\ell r}, C_{\ell \delta a}$
Yawning moment	C_n	$1, \beta, p^*, r^*, \delta_r$	$C_{n0}, C_{n\beta}, C_{np}, C_{nr}, C_{n\delta r}$

load the three time segments, define Nzi and izhf and assign the arrays ZAccel (Ndata, Ny) and Uinp (Ndata, Nu), and then call the data preprocessing function “`umr_reg_attas`,” exactly in the same way as before. This is followed by building the output vector Y and data matrix X for each coefficient according to Table 6.1. The parameter estimates are obtained by solving Eq. (6.11) through simple matrix operations. The estimates of the parameters and their absolute and relative standard deviations are printed out. The time history plots of computed and estimated aerodynamic coefficients are generated, but not shown here.

As expected, the one-shot and the iterative procedures yield the same results. This is evident from the second and the third column of Table 6.2, providing estimates of a few selected parameters. The results for the other (pitching and yawing moment) coefficients can be readily verified by running the two cases. The iterative procedure is, however, more flexible as it can be applied to non-linear model postulates.

To the same problem, we also apply the TLS method by calling the program “/FVSysID/chapter06/attas_regTLS.” This basic version provided herewith helps to trace the computational steps elaborated in Sec. V, Eqs. (6.50) and (6.51) and to compare the results generated with OLS. As in previous case, the estimates of the parameters and their absolute and relative standard deviations are printed out, and the time history plots of computed and estimated aerodynamic coefficients are generated, but not shown here in the text. Comparing the second and the last column of Table 6.2, it is observed that the TLS estimates differ only slightly from the OLS estimates; in general, for the data analyzed, they are fairly comparable. This is attributed to the fact that the noise in the measurements of the dependent and independent variables is quite low and that the systematic errors were eliminated in the data preprocessing step. Under such conditions, the two methods tend to be equivalent. The low noise level is attributed to

Table 6.2 Least-squares estimates for few selected parameters

Parameter	One-shot LS; linear model: “attas_regLS”	Iterative LS; linear model: “ml_oem” Test_case = 23	Iterative LS; nonlinear model: “ml_oem” Test_case = 24	One-shot TLS; linear model: “attas_regTLS”
C_{D0}	0.03996	0.03996	0.03896	0.03973
$C_{D\alpha}$	0.26098	0.26098	NA	0.26698
e	NA	NA	0.77730	NA
C_{L0}	0.23545	0.23545	0.23542	0.23439
$C_{L\alpha}$	5.22550	5.22550	5.22648	5.25324
C_{Y0}	-0.00499	-0.00499	-0.00499	-0.00506
$C_{Y\beta}$	-1.00402	-1.00402	-1.00402	-1.01777
$C_{\ell 0}$	0.000184	0.000184	0.000184	0.000192
$C_{\ell \beta}$	-0.10268	-0.10268	-0.10268	-0.10244
$C_{\ell p}$	-0.75976	-0.75976	-0.75976	-0.76644
$C_{\ell r}$	0.23769	0.23769	0.23769	0.25193
$C_{\ell \delta a}$	-0.19247	-0.19247	-0.19247	-0.19365

NA, not applicable.

the high quality sensors and instrumentation used to gather the flight data. The data analyzed here is a typical sample of flight recordings and representative of noise levels generally present in the data used for parameter estimation. In the presence of low cost sensors which may introduce a higher level of noise, the estimates from the two methods may differ more.

The performance of the OLS and TLS methods depend critically on the adequacy of the postulated model. This aspect was not very apparent from the modeling of the aerodynamic coefficients C_D , C_L , C_Y , C_ℓ and C_n , for which, as just discussed, the estimates and the match from the OLS and TLS methods were comparable. In these cases the cross coupling effects were not very strong, and hence the postulated models in Eq. (6.76) were reasonable for the qualitative analysis performed here. However, the TLS method applied to the pitching moment coefficient C_m showed some very discernible discrepancies in the match for C_m compared with that from the OLS method. The estimates were also not comparable. This can be verified from the first plot obtained by running “/FVSysID/chapter06/attas_regTLS.” This anomaly is attributed to the fact that three time segments with elevator, aileron, and rudder inputs are analyzed simultaneously, whereas the model for C_m is based on longitudinal variables only, that is, α , q , δ_e as independent variables, see Eq. (6.76). It does not account for the cross-coupling effects owing to the lateral-directional motion variables β , p , r , δ_r into the pitching motion, which happen to be significant in the present case. The variations in these lateral-directional variables are large in the second and third maneuvers. Since they are missing in the model, they are treated as noise. These discrepancies in the postulated model lead to wrong TLS estimates from three maneuvers. To analyze this case correctly, it would require model augmentation (i.e., introduction of appropriate independent variables). Since our aim is restricted to demonstration of TLS principle, such model improvements are not pursued here. The above discussion can be verified in an alternative way by analyzing separately the first maneuver with an elevator input based on the simplified model of Eq. (6.76). This has been carried out in the same program, where we apply both OLS and TLS to this single maneuver, and plot the results in the second figure. It turns out that the OLS and TLS estimates and the match are now comparable. This aspect of model adequacy needs to be ascertained when applying the OLS or TLS methods to data from multiple maneuvers.

The foregoing OLS and TLS estimates were obtained by treating each dependent variable separately and by suitably choosing different independent variables each time (see Table 6.1). As discussed in Secs. IV and V, the multivariate OLS and multivariate TLS procedures are mathematically feasible, but not practiced because the same set of independent variables is usually not suitable for all the dependent variables. Nevertheless, to demonstrate these techniques, two script files, 1) “attasLonLat_mV_LS.m” and 2) “attasLonLat_mV_TLS.m” (both found in the directory /FVSysID/chapter06/) are provided. These functions are tailored to the combined longitudinal and lateral-directional motion with $(C_D, C_L, C_m, C_Y, C_\ell, C_n)$ as outputs, that is, $n_y = 6$, and $(1, \alpha, q, \delta_e, \beta, p, r, \delta_a, \delta_r)$ as inputs; $n_q = 9$. The unknown parameter Θ is an $(n_q \times n_y)$ matrix. Additional functions (attasLon_mV_LS.m, attasLat_mV_LS.m, attasLon_mV_TLS.m, and attasLat_mV_TLS.m) for the decoupled longitudinal

and lateral-directional motion are also provided for reference purposes. For the same set of independent variables the multivariate or univariate OLS provides the same results. This can be verified by running any one of the above multivariate cases. As a caveat, the same is not true in the case of TLS. To illustrate the point, let us consider the longitudinal motion. The independent variables are $(1, \alpha, q, \delta_e)$ and the three dependent variables are (C_D, C_L, C_m) . The multivariate TLS is based on the composed data matrix $[X \ C_D \ C_L \ C_m]$, where X is the data matrix of the above defined independent variables. Because of the presence of noise in the independent variables X , the TLS problem based on $[X \ C_D \ C_L \ C_m]$ is not the same as solving $[X \ C_D]$, $[X \ C_L]$, $[X \ C_m]$ separately. For the reasons pointed out in Sec. IV we prefer to treat the dependent variables separately.

C. Estimation of Aerodynamic Parameters from a Nonlinear Model

To demonstrate the nonlinear regression method, we consider the following improved modeling of the drag coefficient:

$$C_D = C_{D0} + \frac{C_L^2}{e\pi\Lambda} \quad (6.78)$$

where e is the Oswald factor and Λ the aspect ratio. The Oswald factor characterizes the increase in drag over the ideal condition caused by nonelliptical lift distribution and interference.²⁷ It is typically less than 1. The regression model is nonlinear in the parameters and in the independent variables as well. Equations for the rest of the coefficients, namely for C_L, C_Y, C_t, C_m, C_n , are exactly the same as those in Eq. (6.76). The model definition required for the program “ml_oem” is almost the same as that for the previous case presented in Sec. IX.B, except for the change of test_case number and the function for observation equation (see the function “/FVSysID/chapter04/mDefCase24.m”):

```
test_case = 24; % test case number
obser_eq = 'obs_TC24_attas_regNL'; % function for observation
equations
```

Except for the change in the model for the drag coefficient, the function “obs_TC24_attas_regNL.m” is the same as “obs_TC23_attas_regLonLat.m.” In the case of the linear least squares model estimated in the previous section, all the 20 parameters were initialized to zero. In the present case, to avoid the division by zero, it is necessary to specify a nonzero starting value for e . Furthermore, because of the nonlinearity in the postulated model, reasonable starting values are required for the lift curve slope $C_{L\alpha}$. By setting the flag for test cases to 24, we run the program “/FVSysID/chapter04/ml_oem” to estimate the 20 aerodynamic parameters.

It is found that, starting from the initial values of 1 for e , 5 for $C_{L\alpha}$, and zero for the other 18 parameters, the optimization of the cost function requires six iterations to converge, yielding a final cost function value of 4.272×10^{-33} . Additionally, specifying initial values of 0.03 for C_{D0} and 0.2 for C_{L0} results

in convergence to the same minimum value after four iterations. The results for this model for a few selected derivatives are summarized in Table 6.2. The estimates of side force and rolling moment coefficients, as well as those for the pitching and yawing moment, are exactly the same as those provided by the cases analyzed in the previous section. This should be the case, because these four output variables are treated independently of each other. On the other hand, the drag coefficient is now not independent, but a nonlinear function of the lift coefficient, which itself is one of the dependent variables. Although the lift coefficient is modeled linearly as in Eq. (6.76), the coupling between the drag and the lift coefficients in Eq. (6.78) influences the estimates of both these coefficients. Accordingly, some minor differences are found in the estimates of C_{L0} , C_{La} , and C_{D0} ; the estimates of $C_{D\alpha}$ in Eq. (6.76) and the Oswald factor e in Eq. (6.78) cannot be directly compared.

X. Concluding Remarks

In this chapter we have discussed various equation error approaches to parameter estimation, studied the properties of least squares estimates under ideal conditions and practical considerations. It is shown that the ordinary linear least squares estimation is a one-shot procedure, based on simple matrix algebra operations and does not require initial parameter values to be specified. The basic principle is applicable to nonlinear regression equations as well. It is also possible to extend the least squares procedure to multiple outputs and to models represented in state space, provided all the states and state-derivatives are measured. It has been argued that this may be a restriction in some of the practical cases, as all of them may not be directly measurable. Obtaining the state derivatives from the measured state variables by numerical differentiation is likely to introduce noise. Since the least squares procedure yields biased and inconsistent estimates in the presence of systematic errors and noise in the independent variables, a rigorous data preprocessing is generally necessary to eliminate such undesirable effects. From this aspect, the output error is more versatile, and hence the equation error method is generally used as a start-up procedure to generate good initial parameter values for the output-error method. Three aspects which favor least-squares estimation are 1) a somewhat simpler approach to analyze large amplitude maneuvers through a series of linear models using data partitioning, 2) a somewhat simpler mechanism to judge the appropriateness of individual parameters through stepwise regression, and 3) applicability to unstable aircraft (not covered in this chapter; it will be discussed in Chapter 9). These aspects may outweigh some of the limitations that have been mentioned, as is evident from several successful applications of aircraft parameter estimation using the least-squares method. Related topics of weighted and total least squares, and stepwise selection of variables have been discussed, bringing out each time the pros and cons from the viewpoint of practical applicability.

The least-squares theory and its practical application has been illustrated on a typical example of estimating stability and control derivatives based on linear as well as nonlinear regression models. Estimation of the stability and control derivatives applying the least-squares method requires a fairly elaborative data

preprocessing step to derive the aerodynamic force and moment coefficients from flight measurements of the linear accelerations and angular rates. These computations are mostly in the reverse order of the computations performed in the state equations postulated for the output error method. It has been demonstrated that the iterative algorithm developed in Chapter 4 can be applied to the multivariate multiple regression model.

References

- ¹Gauss, C.-F., *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium*, Perthes and Besser, Hamburg; Germany, 1809 (in Latin); also published as *Theory of the Motion of the Heavenly Bodies Moving About the Sun in Conic Section*, Dover, New York, 1963.
- ²Sorenson, H. W., "Least-Squares Estimation from Gauss to Kalman," *IEEE Spectrum*, Vol. 7, July 1970, pp. 63–68.
- ³Tennenbaum, J., "Gauss' Determination of the Orbit of Ceres, Parts 1–3," *The American Almanac*, 22 December 1997.
- ⁴Goodwin, G. C. and Payne, R. L., *Dynamic System Identification: Experiment Design and Data Analysis*, Academic Press, New York, 1977.
- ⁵Sorenson, H. W., *Parameter Estimation—Principles and Problems*, Marcel Dekker, New York, 1980.
- ⁶Mendel, J. M., *Lessons in Digital Estimation Theory*, Prentice Hall, Englewood Cliffs, NJ, 1987.
- ⁷Klein, V., "Identification Evaluation Methods," AGARD LS-104, Paper No. 2, Nov. 1979.
- ⁸Greenberg, H., "A Survey of Methods for Determining Stability Parameters of an Airplane from Dynamic Flight Measurements," NACA TN-2340, April 1951.
- ⁹Gerlach, O. H., "Determination of Performance, Stability and Control Characteristics from Measurements in Non-Steady Manoeuvres," AGARD-CP-17, Sept. 1966, pp. 499–523.
- ¹⁰Klein, V., "Estimation of Aircraft Aerodynamic Parameters from Flight Data," *Progress in Aerospace Sciences*, Pergamon, Oxford, Vol. 26, No. 1, 1989, pp. 1–77.
- ¹¹Mulder, J. A., Sridhar, J. K., and Breeman, J. H., "Identification of Dynamic Systems—Applications to Aircraft, Part 2: Nonlinear Analysis and Manoeuvre Design," AGARD AG-300, Vol. 3, Part 2, May 1994.
- ¹²Maine, R. E. and Iliff, K. W., "Identification of Dynamic Systems," AGARD AG-300, Vol. 2, Jan. 1985.
- ¹³Hamel, P. G. and Jategaonkar, R. V., "Evolution of Flight Vehicle System Identification," *Journal of Aircraft*, Vol. 33, No. 1, 1996, pp. 9–28.
- ¹⁴Plaetschke, E., Jategaonkar, R. V., Rohlf, D., and Weiss, S., "Methoden zur Schätzung der Parameter eines Flugzeugs im Post-Stall-Bereich," *Proceedings of the 4th Aerospace Symposium on "Safety in Air Traffic,"* Technical University of Braunschweig, Germany, 13–15 Sept. 1994, pp. 297–306 (in German).
- ¹⁵Golub, O. H. and Van Loan, C. F., "An Analysis of the Total Least Squares Problem," *SIAM Journal of Numerical Analysis*, Vol. 17, No. 6, 1980, pp. 883–893.
- ¹⁶Van Huffel, S. and Vanderwalle, J., *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM, Philadelphia, PA, 1991.

- ¹⁷Laban, M. and Masui, K., "Total Least Squares Estimation of Aerodynamic Model Parameters from Flight Data," *Journal of Aircraft*, Vol. 30, No. 1, 1993, pp. 150–152.
- ¹⁸Markovsky, I. and Van Huffel, S., "A Matlab Toolbox for Weighted Total Least Squares Approximation," Catholic University Leuven, Leuven-Heverlee, Belgium, Report No. ESAT-SISTA/TR 2004-220, Nov. 2004.
- ¹⁹Davis, T. G., "Total Least-Squares Spiral Curve Fitting," *Journal of Surveying Engineering*, Vol. 125, No. 4, 1999, pp. 159–176.
- ²⁰Pinkelman, J. K., Batill, S. M., and Kehoe, M. W., "Total Least-Squares Criteria in Parameter Identification for Flight Flutter Testing," *Journal of Aircraft*, Vol. 33, No. 4, 1996, pp. 784–792.
- ²¹Wong, K. Y. and Polak, E., "Identification of Linear Discrete Time Systems Using the Instrumental Variable Method," *IEEE Transactions on Automatic Control*, Vol. AC-12, No. 6, 1967, pp. 707–718.
- ²²Van den Boom, A. J. W., "On the Relation between Weighted Least-Squares Estimators and Instrumental Variable Estimators," *Proceedings of the IFAC 4th Symposium on Identification and System Parameter Estimation*, Sept. 1976, pp. 457–466.
- ²³Batterson, J. G. and Klein, V., "Partitioning of Flight Data for Aerodynamic Modeling of Aircraft at High Angles of Attack," *Journal of Aircraft*, Vol. 26, No. 4, 1989, pp. 334–339.
- ²⁴Weiss, S., Friehmelt, H., Plaetschke, E., and Rohlf, D., "X-31A System Identification Using Single Surface Excitation at High Angles of Attack," *Journal of Aircraft*, Vol. 33, No. 3, 1996, pp. 485–490.
- ²⁵Spark, R. S., Zucchini, W., and Coutsourides, D., "On Variable Selection in Multivariate Regression," *Communications in Statistics—Theory and Methods*, Vol. 14, No. 7, 1985, pp. 1569–1587.
- ²⁶Judd, C. M. and McClelland, G. H., *Data Analysis: A Model-Comparison Approach*, Harcourt Brace Jovanovich, San Diego, CA, 1989.
- ²⁷Oswald, W. B., "General Formulas and Charts for the Calculation of Airplane Performance," NACA Report No. 408, 1932.

Recursive Parameter Estimation

I. Introduction

IN GENERAL, the motion of a flight vehicle can be represented through well-defined equations of motion, whereby adequate nonlinear models with time-invariant parameters can be postulated and updated to characterize aerodynamic effects. Accordingly, the major applications of aircraft parameter estimation have been in an offline mode, applying methods discussed in Chapters 4–6, which assume the availability of data set over a fixed interval of time. Such methods are also called batch mode or post-flight analysis techniques, which implicitly assume that system parameters are constant over the period of observation. All data points are processed together at a time, yielding parameters representing average system behavior. As opposed to batch-processing, recursive estimation methods utilize the data point-by-point as they become available. Such methods are alternatively called sequential, adaptive, or real-time algorithms.^{1,2} We use the terminology of recursive methods which is commonly followed in the control applications.

Historically, it appears that C.-F. Gauss, the inventor of the least squares method (see Chapter 6, Sec. I), had also suggested the recursive least squares (RLS) approach.³ The RLS method was, however, not widely pursued until 1950, when R. L. Plackett rediscovered the algorithm.⁴ This was followed by widespread use of RLS and development of other recursive algorithms. The basic ideas of practical engineering applications of recursive parameter estimation (RPE) originate from control applications in chemical and thermal power-generating industrial processes, where online adaptation of model parameters is desired to increase the overall plant efficiency. In such applications, we encounter not only changing operating conditions but also several different types of system disturbances. Moreover, the control algorithms are usually based on a simplified representation of a highly complex process, for which exact mathematical representation may require prohibitive effort or may not even be possible. In the case of flight vehicles, the primary motivation for recursive estimation could be to obtain immediate knowledge about an aircraft model, which is essential for designing adaptive control, fault detection and reconfiguration of control systems following damage. These real-time applications necessarily call for recursive estimation methods. During the 1970s, initial attempts were undertaken to apply such techniques in flight. However,

owing to the limited computational power of on-board computers, this approach did not find widespread use then.^{5,6} During recent years, the research has once again been focused on the development of real-time parameter estimation techniques to address applications in the area of adaptive control, fault tolerant systems, and experiment design.^{7–9}

To some extent such methods can also be useful for on-board verification of recorded data and for iterative experiment design. If done onboard, the RPE results from a nonoptimal maneuver could be used to design the subsequent test maneuver and so on throughout the flight test until acceptable results are achieved. This adaptive experiment design and data analysis could provide a higher level of confidence and ensure that sufficient high-quality data are collected for the modeling task before landing. The flight testing in this way would, therefore, increase the efficiency and effectiveness since the complete identification cycle would be carried out in flight in an automated fashion. However, as discussed in Chapter 2, based on the existing a priori knowledge about the flight vehicles and using some simple guidelines, adequate experiments in most of the cases can be designed. Furthermore, near real-time processing capabilities could also serve the intended purposes in some of these cases. With powerful computers now being available, it becomes possible to gather data and apply the classical offline algorithms implemented on an onboard computer before proceeding to the next maneuver during the flight. Such techniques were applied for safer envelope expansion through real-time comparison of flight tests data with predicted responses,¹⁰ and for onboard estimation of aerodynamic characteristics.^{11–13}

RPE methods are characterized by the following features:

- 1) They are approximations of the more elaborate nonrecursive methods that we have already studied in the foregoing chapters.
- 2) By nature, they cater for systems with time-varying parameters.
- 3) Computer memory requirements are small, because storage of past data is not required and hence they are suitable for online implementation even on a small onboard computer or a chip.

The second attribute, namely the ability to track time-varying parameters, helps us indirectly in aerodynamic modeling. Estimates evolve over the time, that is, parameter estimates vary in time, even when they may represent a constant effect which is, as already argued, usually the case for flight vehicles. This leads to accounting for nonlinearities in a system model, at least to some extent, although the postulated model may be linear. Likewise, it also helps to account for the changes in the flight conditions, which implies that the dependencies of some of the aerodynamic derivatives on other motion variables like angle of attack or Mach number or configuration changes like flap setting are covered automatically by the same linear model.

Although the last two behaviors are in favor of RPE, there are certain limitations associated with these techniques. For example:

- 1) The convergence of standard RPE methods, like recursive least squares, is slow and may not be adequate for real-time fault detection or to detect sudden changes in the dynamics.

2) Convergence can be improved by incorporating a forgetting factor to discard older data and thereby rely more on recent data. This, however, leads to increased noise sensitivity. Shorter records are necessary for faster adaptation, whereas longer records are necessary to distinguish noise. A compromise between the rate of tracking parameter changes and noise sensitivity is necessary. A wrong choice of forgetting factor can result in estimates oscillating around the true values.

3) Uninterruptedly running a recursive estimation procedure may cause numerical problems, because of lack of or limited information content pertaining to dynamic motion. During parts of the flight phases, like steady level flight, the control and motion variables could be below the noise level.

4) Verification of data collinearity is not a part of RPE methods. The nearly correlated states and controls affect estimation of stability and control derivatives.

Determination of instrument errors such as biases, scale factors, and time delays is yet another issue that needs some consideration. We will address the techniques of flight-path reconstruction in Chapter 10; if needed, those methods can be implemented in real-time to perform flight-path reconstruction¹⁴ to estimate biases and scale factors, and to calibrate flow variables. Estimation of time delays using a recursive approach is basically feasible, but is usually not considered in the application pertaining to aircraft flight path reconstruction or parameter estimation from flight data applying such recursive algorithms. Offline estimation procedures based on the output error method, which can also be implemented onboard in near-real-time, can cope with such problems more easily. As will be demonstrated in Chapter 10, elaborative offline procedures are more powerful to address such nonlinear estimation problems. In general, the presence of such errors can adversely affect the estimates. For this reason we restrict ourselves here to recursive estimation of aerodynamic parameters assuming that measured data are of adequate quality and free of major instrument errors. Furthermore, owing to some of the aforementioned limitations, we prefer more accurate and powerful offline methods, unless there is a need for online estimation, for example in terms of fault detection and online control or reconfiguration. Accordingly, we further restrict ourselves mainly to models that are adequate to demonstrate the methodology behind recursive estimation methods.

In this chapter, we start with recursive least squares (RLS) method, which is a recursive version of the ordinary least squares method that was studied in Chapter 6. This is followed by a recursive weighted least squares methods based on a forgetting factor to de-emphasize older data. We then look at a so-called sequential batch procedure, based once again on weighted least squares, but incorporating a check to judge the information content in the new incoming data before discarding the old one to include the new one. This is followed by yet another method called Fourier transform regression (FTR) based on the least squares approach, but now implemented in the frequency domain using the discrete Fourier transform. We then turn our attention to a different class of techniques based on a filtering approach, which is widely used for state estimation in the presence of process and measurement noise. Augmenting the basic system

states with system parameters and applying a nonlinear filter to the augmented states allows recursive estimation of parameters. Two nonlinear filter algorithms are discussed. First one is the commonly applied extended Kalman filter (EKF) based on linearization of system dynamics, and the second one is the more recently introduced nonlinear filter, called unscented Kalman filter (UKF), having better performance than the EKF. Two different implementations of UKF are presented. This is followed by a state estimator without requiring knowledge of the noise covariances, called extended forgetting factor recursive least squares (EFRLS),¹⁵ which is an extension of the RLS to linear dynamic systems.

Finally, we test these recursive algorithms in an offline environment on flight data pertaining to the longitudinal and lateral-directional modes of research aircraft VFW-614 ATTAS and HFB-320. The RLS results are compared with the LS method discussed in Chapter 6. For state space models, the results from the standard iterative output error method (OEM) discussed in Chapter 4 or the filter error method (FEM) elaborated in Chapter 5 form the basis for evaluating the performance of these recursive methods. Their performance in terms of parameter convergence, robustness in the presence of noise, and the computational effort is evaluated.¹⁶ As in the case of previous chapters, the flight data and the software are appended to this book.

II. Least Squares-based Recursive Methods

There are many variants of recursive algorithms based on the least squares principle. First, we consider the ordinary least squares method to show how an offline algorithm can be modified to a recursive computational form and to bring out the fact that such methods are approximations of the more elaborate nonrecursive methods. In general, following a similar approach any offline method can be converted into an equivalent recursive form. For example, the output error method based on the maximum-likelihood principle, discussed in Chapter 4, can be translated into an equivalent recursive maximum likelihood (RML) method.¹⁴ Owing to certain limitations of RLS, which we will discuss in our treatment on the subject, more commonly applied algorithms are based on the use of a forgetting factor to de-emphasize the older data. Such methods have better adaptation (convergence) rates and allow parameters to be estimated in time-varying systems.

A. Recursive Least Squares

From Sec. II in Chapter 6, we recall that, given a set of k samples of independent variables $x = (x_1 \ x_2 \ \dots \ x_{nq})^T$ and of dependent variable y , the least squares estimates of θ are obtained as

$$\begin{aligned}\hat{\theta}(k) &= (X_k^T X_k)^{-1} X_k^T Y_k \\ &= P(k) X_k^T Y_k\end{aligned}\tag{7.1}$$

where $X_k = [x(1) \ x(2) \ \dots \ x(k)]^T$, $Y_k = [y(1) \ y(2) \ \dots \ y(k)]^T$, and the inverse of the information matrix $(X_k^T X_k)^{-1} = P(k)$. We note the change from N for the total number of data points used in Sec. II of Chapter 6 for offline algorithm to k in this chapter, because we wish to derive an algorithm to process data sample-by-sample as they become available.

Now, if we assume that one more sample point denoted by $[x(k+1), y(k+1)]$ becomes available, then similar to Eq. (7.1), the offline least squares estimates from the complete set of 1 to $(k+1)$ data points will be given by

$$\hat{\theta}(k+1) = P(k+1)X_{k+1}^T Y_{k+1} \quad (7.2)$$

where $X_{k+1} = [X_k; x^T(k+1)]$ and $Y_{k+1} = [Y_k; y(k+1)]$. Equation (7.2) can be rewritten as

$$\hat{\theta}(k+1) = P(k+1)[X_k^T Y_k + x(k+1)y(k+1)] \quad (7.3)$$

Using $X_{k+1} = [X_k; x^T(k+1)]$ and through simple matrix multiplications, we obtain $P(k+1)$ appearing in Eq. (7.3) in terms of $P(k)$:

$$\begin{aligned} P(k+1) &= [X_{k+1}^T \ X_{k+1}]^{-1} \\ &= [P^{-1}(k) + x(k+1)x^T(k+1)]^{-1} \end{aligned} \quad (7.4)$$

Now, applying the so-called matrix inversion lemma, namely for matrices A , B , C , and D of compatible dimensions

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[D^{-1}A^{-1}B + C^{-1}]^{-1}DA^{-1} \quad (7.5)$$

and substituting $A = P^{-1}(k)$, $B = x(k+1)$, $C = I$, and $D = x^T(k+1)$, Eq. (7.4) leads to

$$P(k+1) = P(k) - P(k)x(k+1)[x^T(k+1)P(k)x(k+1) + I]^{-1}x^T(k+1)P(k) \quad (7.6)$$

which can be expressed in terms of the gain matrix K as:

$$P(k+1) = P(k) - K(k+1)x^T(k+1)P(k) \quad (7.7)$$

where the gain matrix K is defined as

$$K(k+1) = P(k)x(k+1)[x^T(k+1)P(k)x(k+1) + I]^{-1} \quad (7.8)$$

Now, substituting Eq. (7.7) in Eq. (7.3) we derive the recursive formula for updating the parameters θ :

$$\begin{aligned}\hat{\theta}(k+1) &= P(k+1)[X_k^T Y_k + x(k+1)y(k+1)] \\ &= [P(k) - K(k+1)x^T(k+1)P(k)][X_k^T Y_k + x(k+1)y(k+1)] \\ &= P(k)X_k^T Y_k + P(k)x(k+1)y(k+1) - K(k+1)x^T(k+1)P(k)X_k^T Y_k \\ &\quad - K(k+1)x^T(k+1)P(k)x(k+1)y(k+1)\end{aligned}\quad (7.9)$$

However, from Eq. (7.1) we know that $P(k) X_k^T Y_k = \hat{\theta}(k)$. Accordingly, Eq. (7.9) can be further simplified as

$$\begin{aligned}\hat{\theta}(k+1) &= \hat{\theta}(k) + P(k)x(k+1)y(k+1) - K(k+1)x^T(k+1)\hat{\theta}(k) \\ &\quad - K(k+1)x^T(k+1)P(k)x(k+1)y(k+1)\end{aligned}\quad (7.10)$$

Post-multiplying Eq. (7.8) with $[x^T(k+1)P(k)x(k+1) + 1]$, and substituting the result for $P(k)x(k+1)$ in the second term of Eq. (7.10) and minor sorting of the terms leads to

$$\begin{aligned}\hat{\theta}(k+1) &= \hat{\theta}(k) - K(k+1)x^T(k+1)\hat{\theta}(k) \\ &\quad + \left\{ K(k+1)x^T(k+1)P(k)x(k+1) + K(k+1) \right. \\ &\quad \left. - K(k+1)x^T(k+1)P(k)x(k+1) \right\} y(k+1) \\ &= \hat{\theta}(k) - K(k+1)x^T(k+1)\hat{\theta}(k) + K(k+1)y(k+1) \\ &= \hat{\theta}(k) + K(k+1)[y(k+1) - x^T(k+1)\hat{\theta}(k)]\end{aligned}\quad (7.11)$$

Thus, we now have all the information necessary to implement the recursive least squares method. To facilitate easy referencing in the text, the RLS algorithm consisting of Eqs. (7.11), (7.8) and (7.7) is summarized here:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + K(k+1)[y(k+1) - x^T(k+1)\hat{\theta}(k)] \quad (7.12)$$

$$K(k+1) = \frac{P(k)x(k+1)}{1 + x^T(k+1)P(k)x(k+1)} \quad (7.13)$$

$$P(k+1) = P(k) - K(k+1)x^T(k+1)P(k) \quad (7.14)$$

where we have reformulated Eq. (7.8) into Eq. (7.13), because the last term in the square bracket on the right-hand side of Eq. (7.8) is a scalar, and amounts to simple division and not a matrix inversion. Thus, we note that the matrix

inversion required in the offline algorithm has been replaced through scalar division at each step in the recursive version. Thus, the RLS algorithm consists of simple matrix multiplications and additions, whereas the one-shot offline LS procedure involves matrix inversion. To start the algorithm, initial values need to be specified for parameters θ and matrix P . In the absence of any a priori knowledge, we can start with the zero initial values for the parameters; this is similar to the one-shot LS method. The initial P can be assumed to be diagonal matrix with large values, indicating large variances.

It may be recalled from Sec. II.A in Chapter 6 that the estimate of the expected variance in the least squares estimates of n_q parameters from N data points is given by

$$\hat{\sigma}^2 = \frac{1}{N - n_q} \sum_{k=1}^N [y(k) - X(k)\hat{\theta}]^2 \quad (7.15)$$

In the present case, as the estimates evolve over the time with number of data points processed k increasing continuously, the estimate of σ tends to be highly inaccurate initially, because the nonconverged estimates will be used to estimate the covariance matrix. This does not affect the estimates themselves, which are given by Eq. (7.12), but affects only the computation of σ in Eq. (7.15) for k data points. This is one of the limitations of the recursive least squares approach. To obtain an estimate of the accuracy of the estimates, we usually apply Eq. (7.15) once the estimates are converged or when the recursive procedure is terminated.

B. Recursive Weighted Least Squares

The RLS algorithm just described is the simplest of the various recursive procedures. However, in practice the convergence of RLS is generally poor. In order to allow RLS to adapt rapidly with time, the past information must be quickly discarded. This is often accomplished using a forgetting factor, λ , which introduces an exponentially decaying weights on the past measurements.^{1,2} Weighting functions for different values of λ are computed using the function “/FVSysID/chapter07/weightFun_Lamda.m” and are plotted in Fig. 7.1. It is observed that smaller values of λ tend to neglect more and more data points in the farther past. The cost function minimized in this case is given by

$$J(\theta) = \frac{1}{2} \sum_{i=1}^k \lambda^{k-i} \varepsilon^2(i) \quad (7.16)$$

Compared with Eq. (6.37), the exponential weighting is necessary to arrive at recursive version. Without going into the exact derivation, which is very similar to that demonstrated in Sec. II.A for RLS, we simply state here the final algorithmic steps.¹

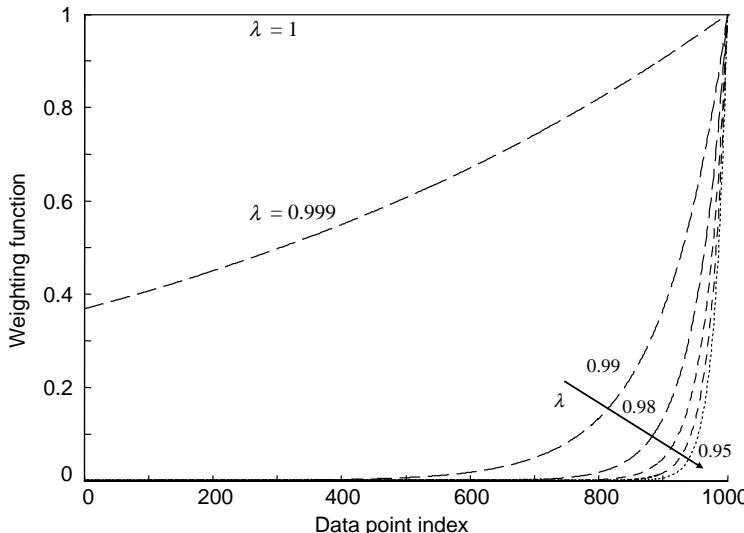


Fig. 7.1 Weighting functions for different forgetting factors.

The RLS with a forgetting factor λ consists of the following computational steps:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + K(k+1)[y(k+1) - x^T(k+1)\hat{\theta}(k)] \quad (7.17)$$

$$K(k+1) = \frac{P(k)x(k+1)}{\lambda + x^T(k+1)P(k)x(k+1)} \quad (7.18)$$

$$P(k+1) = \frac{1}{\lambda} \{P(k) - K(k+1)x^T(k+1)P(k)\} \quad (7.19)$$

We can easily deduce that the standard RLS is a special case of weighted least squares with forgetting factor $\lambda = 1$. Further discussion on the effect of λ and guidelines to choose the same is deferred to Sec. III.C.

C. Locally Weighted Regression Method

Yet another weighted least squares approach is to minimize Eq. (6.37), yielding estimates given by Eq. (6.40) by considering step-by-step sets of data, that is, N , of limited size. Because online estimation problems may impose constraints of storing large size data matrices, the procedure can be implemented for a fixed number of data points, and repeated each time after collecting these data points. It leads to a sequential batch approach. It is also possible to define a weighting matrix in an exponential form to place more importance on data points closer to the current point than on the points in the past, which is the same principle of forgetting factor. Such a diagonal weighting matrix is

given by⁹

$$W_{ii} = e^{-(d_i/2\kappa^2)} \quad (7.20)$$

where d_i is the norm of the difference between the current point and i th one, and κ is the time varying “Gaussian window width,” which can be varied to change to effect of data points in the past. The smaller the κ , the smaller is the set of data points weighted in the cost function, which implies that more emphasis will be given to current data points, and vice versa. For the weighting matrix defined in Eq. (7.20), through proper substitution of W according to Eq. (7.20) in Eq. (6.40), it can be easily verified that the so-called locally weighted regression (LWR) estimates are given by^{9,17,18}

$$\hat{\theta} = (X^T W^2 X)^{-1} X^T W^2 Y \quad (7.21)$$

The simplest approach to implementing the least squares estimation in a batch-sequential mode, with suitably defined weighting matrix, is to simply collect data for a pre-defined length and obtain the estimates by solving, for example, Eq. (7.21). Each time the data matrix is rebuild completely anew, without checking for the information content of the previous data matrix. This is a very inefficient approach, because we may blindly discard useful information, for example that corresponding to dynamic maneuver, which is essential for estimation. A slightly improved strategy, compared with complete rebuild of the data matrix, would be to delete sequentially the oldest data from the data matrix WX . In such a case, we might once again delete rows whose information content is large. Therefore, a much better strategy would be to check the rows of the data matrix for their relative importance, and selectively remove those which contribute less, to be replaced through the new data point. The deleted row could be located anywhere in the data being processed. This is possible because, as already pointed out in Sec. VII of Chapter 6, the least squares estimation does not depend on the temporal relation between the data points.

One criterion to judge the quality of the incoming data will be to evaluate the rows of data matrix WX and delete that which contributes least (and even less than the new data) to the trace of the inverse of the data information matrix, namely $(X^T W^2 X)^{-1}$, which is the first term on the left-hand side of Eq. (7.21). Let us denote WX as follows:⁹

$$WX = \begin{bmatrix} Z \\ H \end{bmatrix} \quad (7.22)$$

where H is the row to be deleted. Then, we have

$$(X^T W^2 X)^{-1} = \left([Z^T \ H^T] \begin{bmatrix} Z \\ H \end{bmatrix} \right)^{-1} = (Z^T Z + H^T H)^{-1} \quad (7.23)$$

Now, denoting $P = Z^T Z + H^T H$ and $R = Z^T Z$ implies $P = R + H^T H$, or in other words $R = P - H^T H$. Now applying the matrix inversion lemma of Eq. (7.5) we get

$$\begin{aligned} R^{-1} &= [P - H^T H]^{-1} \\ &= P^{-1} - P^{-1} H^T (H P^{-1} H^T - I)^{-1} H P^{-1} \end{aligned} \quad (7.24)$$

Taking a trace (i.e., sum of the diagonal elements which is also sum of the eigenvalues) of Eq. (7.24) leads to

$$\text{tr}(R^{-1}) = \text{tr}(P^{-1}) - \text{tr}[P^{-1} H^T (H P^{-1} H^T - I)^{-1} H P^{-1}] \quad (7.25)$$

Since $H P^{-1} H^T$ is a scalar, we can write the second term on the right hand side of Eq. (7.25) as

$$\text{tr}[P^{-1} H^T (H P^{-1} H^T - I)^{-1} H P^{-1}] = \frac{\text{tr}[P^{-1} H^T H P^{-1}]}{H P^{-1} H^T - I} \quad (7.26)$$

Now, using the matrix algebra rule, namely if A and B are matrices such that AB is a square matrix then $\text{tr}(AB) = \text{tr}(BA)$, we can rewrite Eq. (7.26) as

$$\begin{aligned} \text{tr}[P^{-1} H^T (H P^{-1} H^T - I)^{-1} H P^{-1}] &= \frac{H P^{-1} P^{-1} H^T}{H P^{-1} H^T - I} \\ &= \frac{H(P^2)^{-1} H^T}{H P^{-1} H^T - I} \end{aligned} \quad (7.27)$$

Note that trace is not required on the right-hand side of Eq. (7.27), because as already mentioned $H P^{-1} H^T$ is a scalar. Now, denoting the singular value decomposition of a real, symmetric, positive definite matrix $P = USV^T$, it follows that $P^2 = P^T P = VS^2 V^T$ and $P^{-2} = VS^{-2} V^T$, which reduces Eq. (7.27) to

$$\text{tr}[P^{-1} H^T (H P^{-1} H^T - I)^{-1} H P^{-1}] = \frac{H V S^{-2} V^T H^T}{H V S^{-1} V^T H^T - I} \quad (7.28)$$

The problem now reduces to finding row H among all the rows of WX such that the trace in Eq. (7.28) is maximized. Denoting $C = HV$, and making use of the fact that S is diagonal, we can rewrite the right-hand side of Eq. (7.28) as follows:

$$\text{tr}[P^{-1} H^T (H P^{-1} H^T - I)^{-1} H P^{-1}] = \frac{\sum_{j=1}^{n_q} (c_{ij}^2 / s_{jj}^2)}{\sum_{j=1}^{n_q} (c_{ij}^2 / s_{jj}^2) - 1} \equiv F(i) \quad (7.29)$$

Thus, it remains to find the index of the row having maximum value, that is, $\max F(i)$, which is straightforward and needs no further explanation. Although the procedure has been elaborated considering H to be a row vector, it can be extended to a more general case of multiple rows.

The above elaborated procedure is computationally slightly involved, because we need to compute the SVD twice, once from matrix P for the retention and deletion scheme based on Eq. (7.29) and once to compute the estimates using Eq. (7.21). The number of data points used in Eq. (7.21) for sequential batch update and the subsequent test for refreshing the same will influence the computational time; 50 samples at every 10 time steps have been used in practice.⁹ The above procedure is appealing because it leads to inclusion of the new data only when it leads to improved information compared with that which is already available in data set of fixed length size; in turn it results in reduction of the variances of the estimated parameters. The approach also automatically retains the best portions of data which have been processed in the past and, loosely speaking, helps to filter out the noise. In other words the converged estimates, obtained from a dynamic maneuver performed for system identification purposes, do not tend to diverge in spite of the long phases of the maneuver in which there is little dynamic motion. Nevertheless, if recursive estimation runs over a considerable length of time, it may become necessary to reinitialize the data matrix or introduce a forgetting factor to de-emphasize older data. Otherwise, the data matrix may contain information which is too far in the past, and this may affect the ability to track the changes in the system behavior.

D. Fourier Transform Regression

In the previous chapters and preceding sections of this chapter, we have been dealing with methods, both offline and recursive, in the time domain. In this section we slightly deviate from our set goal to address a particular method in the frequency domain, not only because it is based on the equation error approach we studied in Chapter 6, but also since it appears to be quite promising and advantageous compared with other recursive algorithms.^{8,16,19,20} The basis for this Fourier transform regression (FTR) method is the discrete Fourier transform of the time domain data into the frequency domain. This is carried out using sampled time domain data and a simple Euler approximation of the finite Fourier integral. In the following theoretical development, we draw on the work of Ref. 8.

The Fourier transform of an arbitrary signal $x(t)$ is given by

$$\tilde{x}(\omega) = \int_0^T x(t)e^{-j\omega t} dt \quad (7.30)$$

where \sim denotes the discrete Fourier transform of the corresponding variable. The simple Euler approximation to the above Fourier transforms yields

$$\tilde{x}(\omega) \approx \Delta t \sum_{k=0}^{N-1} x_k e^{-j\omega k \Delta t} \quad (7.31)$$

where k indicates the discrete time index, Δt the sampling interval, and N the total number of data points. Denoting the summation on the right-hand side of

Eq. (7.31) as the discrete Fourier transform $\tilde{X}(\omega)$, that is,

$$\tilde{X}(\omega) = \sum_{k=0}^{N-1} x_k e^{-j\omega k \Delta t} \quad (7.32)$$

we can rewrite Eq. (7.31) as follows:

$$\tilde{x}(\omega) \approx \tilde{X}(\omega) \Delta t \quad (7.33)$$

Equation (7.32) is a first-order Euler approximation of the finite Fourier transform of Eq. (7.30), which is valid for a small sampling time Δt . If this is not true, then some corrections will be necessary to reduce the errors.²¹ From Chapter 3, Sec. VIII we recall our discussion on the Euler integration method and its limitations.

Now, applying the Fourier transform to the linear model of Eqs. (3.24) and (3.25), and neglecting the process noise $w(t)$, we obtain an equivalent system represented in the frequency domain as

$$j\omega \tilde{x}(\omega) = A\tilde{x}(\omega) + B\tilde{u}(\omega) \quad (7.34)$$

$$\tilde{y}(\omega) = \tilde{x}(\omega) \quad (7.35)$$

where we have considered only the states as the observation variables. In contrast to Eqs. (3.27) and (3.28), we have omitted here the initial conditions and measurement biases for reasons which we will discuss shortly.

Now, assuming that the measurements of the state, output, and input variables are available, we can make use of the equation error formulation discussed in Chapter 6 to estimate parameters appearing in the matrices A and B appearing in Eq. (7.34). Accordingly, similar to Eq. (6.33) in the time domain, we can write the least squares cost function in the frequency domain for the k th state equation as:²²

$$J_k = \frac{1}{2} \sum_{n=1}^m |j\omega_n \tilde{x}_k(n) - A_k \tilde{x}(n) - B_k \tilde{u}(n)|^2 \quad (7.36)$$

where m is the number of frequencies [instead of N the data points in Eq. (6.33)], A_k and B_k the k th rows of matrices A and B respectively, and $\tilde{x}_k(n)$ the k th element of vector \tilde{x} for frequency ω_n . Symbols $\tilde{x}(n)$ and $\tilde{u}(n)$ denote the Fourier transform of the state and control vectors for frequency ω_n . Similar cost expressions can be written down for other output equations. The number of frequencies m over which the cost function is evaluated depends upon the range of interest, which is usually selected as evenly spaced between ω_{\min} and ω_{\max} . For rigid-body aerodynamic model, it is usually sufficient to consider frequencies from 0.01 to 1.5 Hz. Furthermore, since we are mainly interested in the stability and control derivatives, we can neglect the zero frequency that corresponds to the trim and measurement biases. Thus excluding the zero frequency in the cost function

eliminates the need to estimate initial conditions and measurement biases. This is one of the advantages of the frequency domain approach.

In matrix notation of Eq. (6.3), the standard regression problem with complex data can be formulated as

$$\tilde{Y} = \tilde{X}\Theta + \tilde{\varepsilon} \quad (7.37)$$

where $\tilde{\varepsilon}$ represents the complex equation error in the frequency domain, Θ the vector of unknown model parameters appearing in matrices A_k and B_k , and the matrix of independent variables and vector of dependent variable are given by:

$$\tilde{X} = \begin{bmatrix} \tilde{x}^T(1) & \tilde{u}^T(1) \\ \tilde{x}^T(2) & \tilde{u}^T(2) \\ \vdots & \vdots \\ \tilde{x}^T(m) & \tilde{u}^T(m) \end{bmatrix} \quad \text{and} \quad \tilde{Y} = \begin{bmatrix} j\omega_1 \tilde{x}_k(1) \\ j\omega_2 \tilde{x}_k(2) \\ \vdots \\ j\omega_m \tilde{x}_k(m) \end{bmatrix} \quad (7.38)$$

Similar to Eq. (6.8) in the domain approach, in the present case we can write the least square cost function defined in Eq. (7.36) in terms of \tilde{X} and \tilde{Y} as

$$J = \frac{1}{2}(\tilde{Y} - \tilde{X}\Theta)^* (\tilde{Y} - \tilde{X}\Theta) \quad (7.39)$$

where the superscript “*” denotes the complex conjugate transpose. Following a similar procedure as in Sec. II of Chapter 6, minimization of the cost function of Eq. (7.39) yields²²

$$\hat{\Theta} = [\operatorname{Re}(\tilde{X}^* \tilde{X})]^{-1} \operatorname{Re}(\tilde{X}^* \tilde{Y}) \quad (7.40)$$

The statistical properties of the least squares estimates given by Eq. (7.40) are similar to those discussed in Sec. II.A of Chapter 6. The standard deviations of the estimates are given by the square root of the diagonal elements of the parameter error covariance matrix $P = E\{(\hat{\Theta} - \Theta)(\hat{\Theta} - \Theta)^T\}$, which is given by

$$P = \operatorname{cov}(\hat{\Theta}) = \sigma^2 [\operatorname{Re}(\tilde{X}^* \tilde{X})]^{-1} \quad (7.41)$$

where the equation error covariance σ^2 can be estimated from the residuals,

$$\sigma^2 = \frac{1}{(m - n_q)} [(\tilde{Y} - \tilde{X}\hat{\Theta})^* (\tilde{Y} - \tilde{X}\hat{\Theta})] \quad (7.42)$$

and n_q is the number of unknown parameters Θ .

Equation (7.40) provides a one-shot solution to the least squares problem for data transformed in the frequency domain. Since we are interested in a recursive version of the algorithm, we need to arrive at a procedure to update

point-by-point the discrete Fourier transform defined in Eq. (7.32). We know that at any discrete time point k , the discrete Fourier Transform $\tilde{X}_k(\omega)$ is given by

$$\tilde{X}_k(\omega) = \sum_{i=0}^{k-1} x_i e^{-j\omega i \Delta t} \quad (7.43)$$

We specifically note that $\tilde{X}(\omega)$ has been appended on the left-hand side with subscript k to denote the variables corresponding to those up to the k th point. Likewise, we also note that the upper index of summation N in Eq. (7.32) has been replaced through k for the same reason, and as a consequence the running index is changed to i . Now, if we assume that one more sample point becomes available, then similar to Eq. (7.43), the discrete Fourier transform $\tilde{X}_{k+1}(\omega)$ at the $(k+1)$ data points will be given by

$$\tilde{X}_{k+1}(\omega) = \sum_{i=0}^k x_i e^{-j\omega i \Delta t} \quad (7.44)$$

which can be rewritten by taking out the last point from the summation and then through substitution of Eq. (7.43) as

$$\begin{aligned} \tilde{X}_{k+1}(\omega) &= \sum_{i=0}^{k-1} x_i e^{-j\omega i \Delta t} + x_{k+1} e^{-j\omega(k+1) \Delta t} \\ &= \tilde{X}_k(\omega) + x_{k+1} e^{-j\omega(k+1) \Delta t} \end{aligned} \quad (7.45)$$

Let us take a look at the computational aspects of implementing Eq. (7.45), which are of particular interest to us. For this purpose, we further expand on the exponential term appearing in the right-hand side of Eq. (7.45) as

$$e^{-j\omega(k+1)\Delta t} = e^{-j\omega k \Delta t} e^{-j\omega \Delta t} \quad (7.46)$$

For a given sampling time Δt and frequency ω , the second term on the right hand side of Eq. (7.46) is a constant, which can be computed once outside the recursive loop and stored. Thus, it is fairly straightforward to note that the update of discrete Fourier transform at any time point needs just one addition in Eq. (7.45) and two multiplications: one in Eq. (7.46) and one in Eq. (7.45). Thus, the computational requirements are marginal for computing the Fourier transforms, but this just one part of the overall algorithm.

Equation (7.45) provides a simple recursive formula for updating the discrete Fourier transform at any time point. Once we have obtained the updated discrete Fourier transform $\tilde{X}_{k+1}(\omega)$, the rest of the procedure is the same as that presented in Eqs. (7.38)–(7.40), which are implemented at each time point. The above FTR approach although recursive, is slightly different than that of RLS presented in Sec. II.A. Both the methods do not require storage of time histories in the past. However, computationally the time-domain approach of RLS given by

Eqs. (7.12)–(7.14) involves a simple division by a scalar and matrix multiplications, whereas the frequency-domain approach of FTR needs the inverse of a matrix or equivalently solving linear algebraic equation by a suitable procedure such as Cholesky factorization or singular value decomposition. This is the part of the algorithm that requires the most computational effort. To speed up the computations, the FTR update can be done at some predefined interval by skipping points, instead of updating the estimates at each discrete time point. The number of points that can be skipped without adversely affecting the results depends upon the sampling frequency and the frequencies of interests included in the Fourier transform. At each update point, the information up to the previous sample, including that of the skipped points, is automatically accounted for by the fixed term appearing in Eq. (7.46) that is, as already mentioned earlier, computed ahead of the recursive loop.

Besides simplicity and fairly simple computations, FTR method is associated with a few other advantages; 1) no tuning parameters are required to be specified, and 2) by omitting the zero frequency, the trim conditions and measurement biases are eliminated. Like any other least squares approach, no starting values of parameters are necessary to start the algorithm. Furthermore, the information from the preceding maneuvers or flights can be used in the current recursive estimation without any further overheads by continuing the recursive calculation of the Fourier transform, thereby improving accuracy of the estimates. The approach is, however, based on the assumption that the measurements of the state variables are available and are error free; we had discussed both of these aspects in connection with the least squares technique in Chapter 6. The method is limited to linear models only. It is also plagued by some of the common problems encountered in the recursive least squares procedure, namely de-emphasizing of data in the distant past through a forgetting factor is necessary to enable tracking of rapid changes in the dynamics, and also that some dynamic excitation is necessary, resulting in motion having amplitudes larger than the noise level. This is particularly critical in fault detection applications. With regard to the last aspect, the FTR performs better compared with other methods presented in this section, because by restricting the range of frequencies to be evaluated within a limited band of ω_{\min} to ω_{\max} , the algorithm automatically filters out the noise outside the range of interest.

The last aspect of FTR algorithm that needs some consideration pertains to the treatment of trim values, which basically correspond to the zero frequency. We argued earlier that, by omitting the zero frequency in the evaluation, we can reduce the number of parameters to be estimated, because the initial conditions and bias terms can be neglected. The system equations (7.34) and (7.35) then truly involve perturbed quantities. We recall here our discussion in Sec. V.B of Chapter 3 in terms of the lumped bias parameters in which the nonzero initial conditions were accounted for indirectly through lumped biases. In the present case, this is not possible, and as a consequence to conform to perturbed equations, it is necessary to remove the trim values from the measurements of the motion variables as well as controls before using them in the algorithm. Failure to do so leads to erratic behavior of the FTR algorithm, resulting from a disproportionately large component at zero frequency. There are different procedures that are possible to remove the trim values; for example we can simply subtract the value

of the first data point, or to reduce noise effects we can average the first few data points, from the measurements before starting the recursion, or use a simple high-pass filter. The other repercussion of neglecting the zero frequency is that the FTR procedure only allows estimation of the dynamic derivatives and that we cannot estimate the zero aerodynamic terms. Both of these limitation are, in general, of minor consequence, particularly for the end goal of real-time estimation.

III. Filtering Methods

Filtering techniques belong to a class of stochastic methods and are primarily used for state estimation in the presence of noise. They use the measured outputs y and inputs u , together with the postulated system model and assumed process and measurement noise statistics, to compute the states and their error covariances. Fundamental to all such methods is a two-step procedure consisting of prediction, also called time update, followed by correction, also called measurement update. The procedure developed by Kalman provides an optimal linear state estimator which operates point-by-point and hence is ideally suited to recursive implementation.²³ Derivation of the Kalman filter is provided in Appendix F. Here, we consider two extensions of the Kalman filter which allow state estimation in nonlinear systems. There are many different implementations possible for the Kalman filter and of the extensions leading to numerically more efficient realizations. We restrict ourselves to standard techniques which provide basic understanding of these methods.

Estimation of parameters applying the filtering approach is an indirect procedure, which needs transformation of the parameter estimation problem into a state estimation problem. We do this by artificially defining the unknown parameters as additional state variables. Considering the system parameter vector Θ as the output of an auxiliary dynamic system

$$\dot{\Theta} = 0 \quad (7.47)$$

and by defining an augmented state vector

$$x_a = [x^T \ \Theta^T]^T \quad (7.48)$$

we can represent the extended system as

$$\begin{aligned} \dot{x}_a(t) &= \begin{bmatrix} f[x_a(t), u(t)] \\ 0 \end{bmatrix} + \begin{bmatrix} F & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w(t) \\ 0 \end{bmatrix} \\ &= f_a[x_a(t), u(t)] + F_a w_a(t) \end{aligned} \quad (7.49)$$

$$y(t) = g_a[x_a(t), u(t)] \quad (7.50)$$

$$z(k) = y(k) + Gv(k) \quad (7.51)$$

where the subscript “a” denotes the augmented variables, f_a and g_a are system functions of the state and observation equations of the augmented systems, and F_a is the corresponding process noise matrix. As in the previous cases covered in Chapter 5, Sec. II, the process noise w and measurement noise v are assumed to be zero mean, uncorrelated and mutually independent; F and G are the corresponding noise distribution matrices. We also note here that the extended model will be nonlinear in the extended states x_a , involving products of the states x and parameters Θ , even though the basic system model might have been linear in x . Thus, irrespective of the type of model, parameter estimation using the filtering approach necessarily leads to a nonlinear filtering problem.

A. Extended Kalman Filter

The EKF gives an approximate solution to the nonlinear state estimation problem, also called as nonlinear filtering problem. It also provides a solution to the combined state and parameter estimation problem,²⁴ and has been used in the past to estimate aerodynamic derivatives.^{25–30} It can also be used for the specific case of unstable aircraft identification,^{31–33} which we will address in Chapter 9. The EKF involves linearization of the postulated nonlinear model about some suitable point and the use of the Kalman filter. Appendix F provides the basic procedural details of the EKF. In general, the nonlinearities in the model affect the EKF performance. Therefore, to retain the system nonlinearities in the state estimation as far as possible, we use a mixed version incorporating state prediction with a nonlinear model and covariance propagation based on a first-order approximation of the system dynamics. As already pointed out in Chapter 5, such a mixed nonlinear filter is found to work satisfactorily in practical cases, including those of flight path reconstruction and aerodynamic model identification.

We can apply the EKF computational procedure elaborated in Appendix F to the extended system defined in Eqs. (7.49)–(7.51) to obtain the estimates of the augmented states x_a , which contain the unknown system parameters. The two steps of the EKF are given by:

Prediction (time update) step

$$\tilde{x}_a(k+1) = \hat{x}_a(k) + \int_{t_k}^{t_{k+1}} f[\hat{x}_a(t), \bar{u}(k)] dt \quad (7.52)$$

$$\tilde{P}_a(k+1) \approx \Phi_a(k+1) \hat{P}_a(k) \Phi_a^T(k+1) + \Delta t F_a F_a^T \quad (7.53)$$

with initial conditions

$$\hat{x}_a(1) = x_{a0}, \quad \hat{P}_a(1) = P_{a0} \quad (7.54)$$

and

$$\Phi_a(k+1) = e^{A_a(k)\Delta t} \approx I + A_a(k)\Delta t + A_a^2(k) \frac{\Delta t^2}{2!} + \dots \quad (7.55)$$

$$A_a(k) = \left. \frac{\partial f_a[x_a(t), u(t)]}{\partial x_a} \right|_{x_a=\hat{x}_a(k)} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial \Theta} \\ 0 & 0 \end{bmatrix} \Big|_{x_a=\hat{x}_a(k)} \quad (7.56)$$

Correction (measurement update) step

$$\tilde{y}(k) = g_a[\tilde{x}_a(k), u(k)] \quad (7.57)$$

$$K_a(k) = \tilde{P}_a(k) C_a^T [C_a \tilde{P}_a(k) C_a^T + R]^{-1} \quad (7.58)$$

$$\hat{x}_a(k) = \tilde{x}_a(k) + K_a(k)[z(k) - \tilde{y}(k)] \quad (7.59)$$

$$\hat{P}_a(k) = [I - K_a(k) C_a] \tilde{P}_a(k) [I - K_a(k) C_a]^T + K_a(k) R K_a^T(k) \quad (7.60)$$

where

$$C_a(k) = \left. \frac{\partial g_a[x_a(t), u(t)]}{\partial x_a} \right|_{x_a=\tilde{x}_a(k)} = \left[\frac{\partial g}{\partial x} \left| \frac{\partial g}{\partial \Theta} \right. \right]_{x_a=\tilde{x}_a(k)} \quad (7.61)$$

where \bar{u} denotes the average or interpolated values of the inputs between the time points k and $k+1$, \sim the predicted variables, \wedge the updated variables, K_a the Kalman gain matrix, and $R = GG^T$ the measurement noise covariance matrix.

Depending upon the number of unknown parameters, the matrices A_a and C_a can become much larger than those for the original system. From Eq. (7.56) we observe that many elements of the augmented matrix A_a are zero; this fact can be used while programming the computation of state transition matrix Φ_a defined in Eq. (7.55). Since these computations are performed at every data point, considerable savings in the computational time can be achieved, particularly when the number of unknown parameters far exceeds the number of system states. For propagation of the states across the two discrete time points according to Eq. (7.52) we use one of the Runge–Kutta methods discussed in Chapter 3, Sec. VIII.A.

The first-order approximation of the covariance propagation in Eq. (7.53) is valid for small sampling time, which is usually the case we encounter while dealing with flight data analysis. The initial conditions x_{a0} need to be specified. It is also necessary to specify the initial covariance matrix $P_a(0)$ for the augmented state vector. The initial values for the covariances corresponding to the auxiliary state variables, that is, corresponding to the unknown parameters, should reflect the confidence in the starting values of the parameters Θ . In the absence of any a priori knowledge, it is common to assume fairly high values for the initial covariance matrix $P_a(0)$, indicating lesser confidence. Such a choice also aids better tracking of values which may deviate greatly from the initial guess values. The covariance matrix P_a provides the information about the accuracy of the augmented states, which includes the unknown parameters. Standard

deviations of the parameter estimates are readily obtained as square roots of the corresponding diagonal elements of P_a .

A priori specification of the process and measurement noise covariance matrices, $F_a F_a^T (=Q)$ and $G G^T (=R)$ is necessary to use the EKF as a parameter estimator. The problem of estimating the unknown noise statistics for a known system dynamics is often referred to in the literature as “adaptive filtering.” Several approaches have been adopted in practice, one of them being developed by Myers and Tapley,³⁴ which is more intuitive and more commonly used. We do not address any of these techniques here. For good EKF performance proper values or tuning of these matrices may be necessary.

The measurement noise covariance matrix, R , can usually be obtained in a relatively straightforward manner directly from the measurements using the Fourier decomposition approach³⁵ or from the characteristics of the instrumentation used. The process noise covariance matrix, Q , is, however, more difficult to determine. In practice, Q is usually treated as a diagonal matrix. Its first n_x elements for the states should reflect the process noise (turbulence) level, whereas the following n_q values corresponding to the parameters depend upon the nature of the parameters. For the case considered in Eqs. (7.47)–(7.51), the parameters are treated as constant over the complete period of observation, which is similar to the approach of offline analysis. In such a case the corresponding elements of Q are set to zero; see Eq. (7.49) with a matrix containing F and zeros. This procedure is followed for the test cases addressed in Sec. V. However, if the system parameters change significantly during the period of observation, for example as a result of change of configuration, to due malfunction or failure, or for any other reason, the covariances for the parameters should then be large enough to allow tracking of variations in system parameters. Many attempts have been made recently to estimate the noise covariances adaptively but at the cost of increased computational burden, thus making it undesirable for real-time application.

B. Unscented Kalman Filter

As elaborated in Appendix F, the EKF is a nonoptimal solution based on a first-order approximation of the nonlinear system dynamics. The EKF performance is sensitive to nonlinearities in the system model and to the process and measurement noise covariances which are required to be specified a priori. Strong system nonlinearities, or in other words the higher-order terms neglected in the propagation of states and error covariances, and wrong values of noise statistics may result in biased estimates and in the worst case lead to divergence.³⁶ The mixed filter implementation followed in the previous section using nonlinear equations for propagation of the system states and a linearized model for propagation of error covariances mitigates the problem somewhat, however the fundamental problem of not accounting for nonlinear transformation the random variables undergo remains, which basically affects the accuracy. Various techniques, such as iterated Kalman filter, or second- and higher-order filters, are possible and applied in specific cases to reduce the estimation errors, but they are much more involved. More recently, yet another approach, called the sigma point filters (SPF), has evolved, which retains the standard Kalman filter

form, involves no local iterations, and has better performance.^{37–39} We discuss here just one of them, the UKF.

The UKF is based on propagating a finite set of points, called sigma points, through the nonlinear dynamics, and by approximating the distribution (mean and covariance) through a weighted sum and outer (cross) product of the propagated points. In contrast to the first-order approximation used in the EKF for covariance propagation, in the UKF nonlinear dynamics are used without approximations. Figure 7.2 (adapted from Ref. 39) depicts schematically the two approaches for a two-dimensional case, showing on the left the true mean and covariance propagation using Monte Carlo sampling, in the center the linear propagation in the EKF, and on the right the nonlinear propagation as well as mean and covariance as weighted sums in the UKF. The flaw in the EKF of calculating mean and covariance of random variables undergoing a nonlinear transformation is thus eliminated in the UKF. This difference leads to a better performance of the UKF, which is shown to be equivalent to the second-order EKF, but without calculating the Jacobian or Hessian, that is, first- and second-order approximations of the system dynamics. This characteristic, namely propagation, does not require computations of first- or higher-order approximations of system functions, which together with the resulting improved performance makes it better suited for real-time applications and for nondifferentiable functions. Needless to say, Kalman filter, EKF, and UKF are all equivalent for linear models.

To implement the UKF, it is necessary to define $(2n_a + 1)$ sigma points, where n_a is the total number of states to be estimated, which include the basic system state, the unknown system parameters, as well as the process and measurement noise disturbances. We note the fact that each point consists of a vector. One of the sigma vectors is the expected value of the augmented state vector, and the remaining $2n_a$ points are obtained from the columns of the matrix square

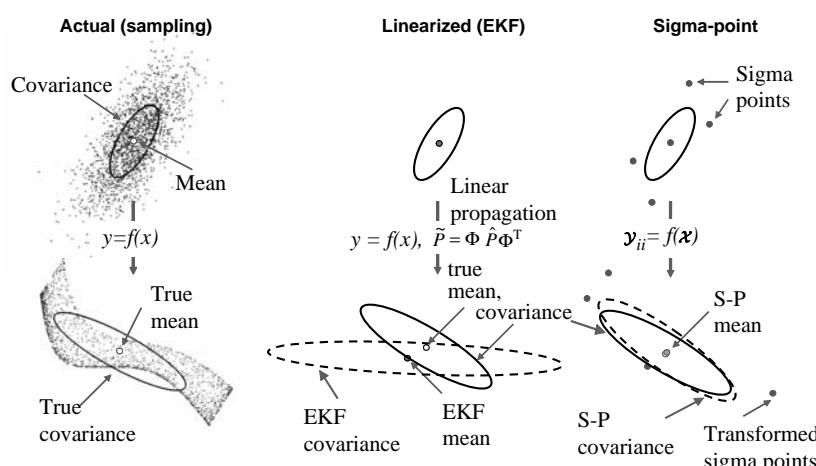


Fig. 7.2 Schematic of linear and nonlinear propagation (Source: Ref. 39).

root $\pm(\gamma P)_i$ for $i = 1, 2, \dots, n_a$, where P is the covariance matrix of the augmented state vector x_a , and $\gamma = \sqrt{(n_a + \lambda)}$ and $\lambda = \alpha^2(n_a + \kappa) - n_a$ are scaling parameters. The constant α determines the spread of the sigma points around the estimated x_a ; it is set equal to a small positive value of less than 1, typically in the range from 0.001 to 1, and $\kappa = 0$ or $3 - n_a$ for state or parameter estimation.^{37,58} The weights required in the computation of mean and covariance are defined as

$$\begin{aligned} W_0^m &= \lambda/(n_a + \lambda) \\ W_0^c &= \lambda/(n_a + \lambda) + (1 - \alpha^2 + \beta) \\ W_i^m &= W_i^c = 1/\{2(n_a + \lambda)\}, \quad i = 1, 2, \dots, 2n_a \end{aligned} \quad (7.62)$$

where the subscript “0” corresponds to the estimated states and $i = 1, 2, \dots, 2n_a$ the other sigma points; the superscripts “m” and “c” indicate weights for the computation of mean and covariance respectively. The constant β is used to incorporate prior knowledge of the distribution of x in the computation of weights for covariances W_0^c ; the optimum value is $\beta = 2$ for Gaussian distribution. The scale parameters are, in general, problem-dependent; some guidelines to choose them are provided in Ref. 40.

To illustrate the UKF procedure, let us first consider a discrete time representation widely used in the applications of the UKF. Accordingly, let the discrete time nonlinear state space model be given by

$$x_{k+1} = f_d(x_k, \Theta_k, u_k, w_k) \quad (7.63)$$

$$y_k = g_d(x_k, \Theta_k, u_k, v_k) \quad (7.64)$$

where x_k is the $(n_x \times 1)$ state vector, Θ_k the $(n_q \times 1)$ vector of unknown parameters, u_k the $(n_u \times 1)$ vector of exogenous inputs, y_k the $(n_y \times 1)$ model output vector, and f_d and g_d the corresponding state and output functions. In a general case, the process noise w_k and the measurement noise v_k are assumed to enter the model nonlinearly as implied by Eqs. (7.63) and (7.64). They are $(n_w \times 1)$ and $(n_v \times 1)$ size vectors, respectively. The simpler case of additive noise will be treated at the end of this section. The measurement vector is denoted by z_k . The augmented state vector of the size $(n_a \times 1)$ is given by

$$x_k^a = [x_k^T \quad \Theta_k^T \quad w_k^T \quad v_k^T]^T \quad (7.65)$$

where the superscript “a” denotes the augmented state vector, $n_a = n_{xp} + n_w + n_v$ and $n_{xp} = n_x + n_q$. Usually we consider cases where $n_w = n_{xp}$ and $n_v = n_y$. The noise processes w_k and v_k are assumed to be zero mean, and with covariance matrices Q and R , respectively. The starting values of the augmented state

vector and its covariance are given by

$$\begin{aligned}\hat{x}_0^a &= E\{x_0^a\} = E\{\hat{x}_0^T \quad \Theta^T \quad w_0^T \quad v_0^T\} \\ &= [\hat{x}_0^T \quad \Theta_0^T \quad 0 \quad 0]\end{aligned}\quad (7.66)$$

$$P_0^a = E\{(x_0^a - \hat{x}_0^a)(x_0^a - \hat{x}_0^a)^T\} = \begin{bmatrix} P_{x0} & 0 & 0 & 0 \\ 0 & P_{\Theta 0} & 0 & 0 \\ 0 & 0 & Q & 0 \\ 0 & 0 & 0 & R \end{bmatrix} \quad (7.67)$$

Following the guidelines already discussed, we choose the constants α , β , and the scaling parameters κ , γ , and λ , and compute the set of weights W_i^m and W_i^c for $i = 0, 1, \dots, 2n_a$.

Without going into mathematical derivation, the standard UKF algorithm can be summarized as follows:³⁹

- 1) Set discrete time point $k = 1$ and build up the augmented state vector \hat{x}_k^a and the corresponding covariance matrix \hat{P}_k^a for the initial values according to Eqs. (7.66) and (7.67), respectively.
- 2) Calculate the $(2n_a + 1)$ sigma points:

$$\hat{\mathcal{X}}_k^a = \left[\hat{x}_k^a \quad \hat{x}_k^a - \gamma\sqrt{\hat{P}_k^a} \quad \hat{x}_k^a + \gamma\sqrt{\hat{P}_k^a} \right] \quad (7.68)$$

- 3) Compute the predicted (time updated) states and covariances:

$$\tilde{\mathcal{X}}_{k+1}^x = f_d[\hat{\mathcal{X}}_k^x, u_k, \hat{\mathcal{X}}_k^w] \quad (7.69)$$

$$\tilde{x}_{k+1} = \sum_{i=0}^{2n_a} W_i^m \tilde{\mathcal{X}}_{i,k+1}^x \quad (7.70)$$

$$\tilde{P}_{k+1} = \sum_{i=0}^{2n_a} W_i^c [\tilde{\mathcal{X}}_{i,k+1}^x - \tilde{x}_{k+1}] [\tilde{\mathcal{X}}_{i,k+1}^x - \tilde{x}_{k+1}]^T \quad (7.71)$$

- 4) Perform the measurement update:

$$\mathcal{Y}_{k+1} = g_d[\tilde{\mathcal{X}}_{k+1}^x, u_{k+1}, \tilde{\mathcal{X}}_{k+1}^w] \quad (7.72)$$

$$\tilde{y}_{k+1} = \sum_{i=0}^{2n_a} W_i^m \mathcal{Y}_{i,k+1} \quad (7.73)$$

$$P_{\tilde{y}\tilde{y}_{k+1}} = \sum_{i=0}^{2n_a} W_i^c [\mathcal{Y}_{i,k+1} - \tilde{y}_{k+1}] [\mathcal{Y}_{i,k+1} - \tilde{y}_{k+1}]^T \quad (7.74)$$

$$P_{\tilde{x}\tilde{y}_{k+1}} = \sum_{i=0}^{2n_a} W_i^c [\tilde{\mathcal{X}}_{i,k+1}^x - \tilde{x}_{k+1}] [\mathcal{Y}_{i,k+1} - \tilde{y}_{k+1}]^T \quad (7.75)$$

$$K_{k+1} = P_{\tilde{x}\tilde{y}_{k+1}} P_{\tilde{y}\tilde{y}_{k+1}}^{-1} \quad (7.76)$$

$$\hat{x}_{k+1} = \tilde{x}_{k+1} + K_{k+1}(z_{k+1} - \tilde{y}_{k+1}) \quad (7.77)$$

$$\hat{P}_{k+1} = \tilde{P}_{k+1} - K_{k+1} P_{\tilde{y}\tilde{y}_{k+1}} K_{k+1}^T \quad (7.78)$$

5) Increment k and jump back to step 2 to continue.

In Eqs. (7.68)–(7.78), $\mathcal{X}^a = [(\mathcal{X}^x)^T (\mathcal{X}^w)^T (\mathcal{X}^v)^T]$ denotes the sigma points, \mathcal{X}^x corresponds to the system states and unknown parameters, “~” (tilde) the predicted values, and “^” (hat) the corrected values.

The foregoing UKF development is valid for a general case of noise disturbances entering the system nonlinearly and for discrete-time models; see Eqs. (7.63) and (7.64). Now, getting back to our specific interest, let us focus on the mixed continuous–discrete time state space models with additive noise disturbances:

$$\dot{x}(t) = f[x(t), u(t), \Theta] + w(t) \quad (7.79)$$

$$y(t) = g[x(t), u(t), \Theta] \quad (7.80)$$

$$z(t_k) = y(t_k) + v(t_k) \quad (7.81)$$

For such a model the UKF steps are very similar to those of Eqs. (7.68)–(7.78), but for the two modifications: 1) propagation of sigma points through numerical integration, and 2) simplification resulting from additive noise disturbances. As in the case of the EKF, to propagate the sigma points based on the continuous time nonlinear equation, we use one of the numerical integration methods instead of Eq. (7.69):

$$\tilde{\mathcal{X}}_{k+1}^a = \hat{\mathcal{X}}_k^a(k) + \int_{t_k}^{t_{k+1}} f[\mathcal{X}^a(t), \bar{u}(k)] dt \quad (7.82)$$

For additive zero-mean noise disturbances, it turns out that augmentation of the state vector through noise vectors is not necessary. In such a case, the augmented state vector is given by $x_k^a = [x_k^T \Theta_k^T]^T$, exactly as considered in the case of EKF, the initial covariance matrix P_0^a in Eq. (7.67) corresponds to that for the states and system parameters only, and the total number of states to be estimated is $n_a = n_{xp}$. The respective noise covariances Q and R are simply

added on the right-hand sides of Eqs. (7.71) and (7.74), yielding

$$\tilde{P}_{k+1} = \sum_{i=0}^{2n_a} W_i^c [\tilde{\mathcal{X}}_{i,k+1}^x - \tilde{x}_{k+1}] [\tilde{\mathcal{X}}_{i,k+1}^x - \tilde{x}_{k+1}]^T + Q \quad (7.83)$$

$$P_{\tilde{y}\tilde{y}_{k+1}} = \sum_{i=0}^{2n_a} W_i^c [\mathcal{Y}_{i,k+1} - \tilde{y}_{k+1}] [\mathcal{Y}_{i,k+1} - \tilde{y}_{k+1}]^T + R \quad (7.84)$$

The reduced size of the augmented state vector (and of sigma points) leads to a reduction in the computational burden.

In the case of UKF, some care is necessary when specifying the initial covariance matrix P_0^a , because the sigma points are generated from the specified initial augmented states x_0^a and P_0^a ; see Eq. (7.68). Each sigma point comprises system states and parameters. Completely inappropriate values may result in large deviations in the computed responses compared with the measured outputs for the first very few data points. Usually, this is not critical, because the updated covariances are used to generate sigma points at each data point subsequently. In general, it does not affect the parameter convergence over a large number of data points. This phenomenon is specific to UKF and does not arise in EKF, where large values are usually specified for the initial covariances.

To recapitulate, Eq. (7.69) or (7.82) propagates the sigma points using the nonlinear dynamics, the mean is computed as a weighted sum given by Eq. (7.70), and the error covariance by Eq. (7.71) or (7.83). The outputs for the sigma points using the nonlinear model are provided by Eq. (7.72) and the system output is the weighted sum given by Eq. (7.73). This is followed by computing the covariance and cross-correlation in Eq. (7.74) or (7.84) and (7.75), respectively, which is required to compute the Kalman gain [Eq. (7.76)]. Once the gain is calculated, the predicted states and their covariances can be updated according to Eqs. (7.77) and (7.78). By comparing the UKF steps given above with those of the EKF in the previous section, we notice that the major changes pertain to computing the mean and covariances of states and outputs, whereas the measurement update is basically similar. It is also apparent that the first- or second-order approximations of the system dynamics are not required.

Thus, we have two versions of the unscented Kalman filter: 1) a general case of noise disturbance entering the system nonlinearly, represented by the augmented vector $x_k^a = [x_k^T \Theta_k^T w_k^T v_k^T]^T$ to which we apply Eqs. (7.66)–(7.68), (7.82), and (7.70)–(7.78), and 2) a special case of additive noise, represented by the state vector $x_k^a = [x_k^T \Theta_k^T]^T$, in which the simplified equations (7.83) and (7.84) are used to compute covariances. For an arbitrary choice of the scaling parameter κ the two versions based on augmentation and nonaugmentation of states through noise vectors can result in some difference; however for the specific choice of $\kappa = 3 - n_a$ they are theoretically equivalent.⁴¹ The assumption of additive process and measurement noise corresponds to the cases we considered in the other chapters.

Even though the sigma point approach is more accurate and helps to mitigate the nonlinearity effects, as in the case of the EKF, proper specification of the

noise covariances is necessary. The augmentation of the state vector through noise vectors in Eq. (7.65) and of the corresponding covariances in Eq. (7.67) is necessary in a general case when noise affects the system nonlinearly, but it does not imply tuning of noise covariances Q and R . This is evident from the additive noise case, which clearly shows that the covariances are just added and kept fixed; see Eqs. (7.83) and (7.84). For such a case of additive noise where the covariances appear explicitly in the covariance prediction equations, tuning techniques similar to those for the EKF or other Kalman filters can be applied. Adaptive tuning is also possible in the other general case of UKF in which the process noise covariance is implicit in the prediction equation, Eq. (7.71).⁴² However, as already mentioned in Sec. III.A, we do not address any of these techniques here.

C. State Estimator without Knowledge of Noise Covariances

In general, inaccurate noise statistics and cross-correlated noise may lead to degraded performance of the Kalman filter algorithms. To circumvent these difficulties, an alternative approach is sometimes attempted based on an extension of the RLS with forgetting factor incorporating dynamic system matrix. Such an algorithm, called EFRLS (extended forgetting factor recursive least squares) does not require knowledge of the noise covariances.^{15,43} Other algorithms are also available that do not require prior knowledge of measurement errors,⁴³ but we restrict ourselves here to the EFRLS only. In its basic form, the EFRLS is developed for the linear dynamic system defined as

$$x(k+1) = \Phi(k)x(k) + w(k) \quad (7.85)$$

$$y(k) = C(k)x(k) \quad (7.86)$$

$$z(k) = y(k) + v(k) \quad (7.87)$$

where k is the discrete time index, Φ the system matrix, w and v the zero-mean random variable. Without going into exact derivation of the algorithm which is found in Ref. 15, the recursive estimate of the states of the above discrete system are given by:

$$x(k+1) = \Phi(k)x(k) + \Phi(k)L(k+1)[z(k+1) - C(k+1)\Phi(k)x(k)] \quad (7.88)$$

$$\begin{aligned} L(k+1) &= P(k)\Phi^T(k)C^T(k+1) \\ &\times [\lambda I + C(k+1)\Phi(k)P(k)\Phi^T(k)C^T(k+1)]^{-1} \end{aligned} \quad (7.89)$$

$$P(k+1) = \frac{1}{\lambda}\Phi(k)[I - L(k+1)C(k+1)\Phi(k)]P(k)\Phi^T(k) \quad (7.90)$$

where λ is the forgetting factor (generally between 0.98 and 0.995), which is an extension of the forgetting factor used in the standard least squares method. As in the recursive weighted least squares procedure, in the present case too, the performance of EFRLS is determined by the choice of forgetting factor.

The investigations show that the EFRLS is less sensitive to the choice of forgetting factor.

The EFRLS computational steps are somewhat similar to the EKF discussed in Sec. III.A. However, there is a subtle difference between the two procedures. EKF incorporates the noise processes explicitly, and has two tuning parameters corresponding to process and measurement noise matrices, compared with the single adaptation parameter λ of EFRLS. In the case of EKF, the performance is mainly influenced by the ratio between the process noise and measurement noise covariances. By lowering or increasing the measurement noise covariance we can put increased or reduced faith in the measurements, respectively; in the first case it would lead to larger variations in the state estimates from sample to sample, whereas a larger value would lead to smoother estimates, which also would mean that rapid fluctuations are not tracked. The reverse is true of the process noise influence.

By de-emphasizing the older data, the forgetting factor leads to a faster adaptation of time-varying parameters. The smaller the forgetting factor is, the better the adaptation to current data. The faster convergence, however, is associated with increased sensitivity to the noise. In fact, if λ is much less than 1, the estimates which might have converged to the true value tend to diverge or oscillate due to small disturbances or noise acting as inputs. Possible different approaches to overcome the high sensitivity to noise are through a time-variant forgetting factor or Kalman filter.⁴⁴

The choice of λ is based on the following considerations.⁴⁵ If the process noise variance is expected to be large, λ should be small, since the past data is not giving more information on the current augmented state states. If the process noise variance is smaller than the measurement noise variance, λ should be of a larger value. This implies that more data should be used to average out the effect of the noise on measurements. Hence, λ should be very close to 1 but less than 1. We can also look at the forgetting factor in terms of the memory index. The memory index (*MI*) of RLS filter can be defined as $MI = 1/(1 - \lambda)$. Thus, if $\lambda = 1$, then *MI* is infinity, implying that the filter has infinite memory. This means that the entire data is given equal weighting, which is the case with the standard recursive least squares method. It is also possible to arrive at a reasonable λ based on the condition number (ratio of the magnitudes of the largest to smallest eigenvalue) of the data matrix. A large condition number indicates an ill-conditioned matrix, and accordingly less weight should be given to the past data. The reader may recall that we used in Sec. X.C of Chapter 4 a similar concept of condition number to detect the ill-conditioning of the information matrix and possible neglect of the search directions to avoid intermediate convergence of the Gauss–Newton or other optimization methods.

Although the EFRLS algorithm is similar to Kalman filter, there is a subtle difference in the two cases. Let us take a look at: 1) applicability or extensions to more general nonlinear models, and 2) the matrix P appearing in the algorithms and computation of the standard deviations.

As already discussed in Sec. III.A, the Kalman filter algorithm can be readily extended to any general nonlinear state space model. Extension of the EFRLS algorithm to a continuous-time state-space model of the form $\dot{x}(t) = Ax(t) + w(k)$ is fairly straightforward. The first term on the right-hand side of Eq. (7.88), namely $\Phi(k)x(k)$, is simply replaced through states obtained

by numerical integration, and Φ appearing in Eqs. (7.88)–(7.90) through the state transition matrix $\Phi = e^{A\Delta t}$. However, extension of the EFRLS to a more general case with deterministic exogenous inputs u represented by

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + w(t) \quad (7.91)$$

$$y(k) = C(k)x(k) + D(k)u(k) + v(k) \quad (7.92)$$

is more involved. For linear systems, the principle of superposition allows us to decompose the above system into two, the first stochastic and the other deterministic. Denoting $x(k) = x_s(k) + x_d(k)$ and $y(k) = y_s(k) + y_d(k)$, it can be easily verified that the system given by Eqs. (7.85)–(7.87) can be decomposed into: 1) $x_s(k+1) = \Phi(k)x_s(k) + w(k)$; $y_s(k) = C(k)x_s(k) + v(k)$; and 2) $x_d(k+1) = \Phi(k)x_d(k) + \Psi Bu(k)$; $y_d(k) = C(k)x_d(k) + D(k)u(k)$, where Ψ is the integral of Φ . We know that $w(t)$ and $w(k)$ are not the same, but since we are dealing here with an estimator without knowledge of noise covariances, the difference is immaterial. To the first system in terms of the state and observation variables (x_s , y_s) we can apply the EFRLS algorithm given by Eqs. (7.88)–(7.90), and the second system in terms of (x_d , y_d) being deterministic amounts to simple simulation. The above approach, although feasible for linear systems, is not very flexible. Furthermore, it cannot be readily extended to non-linear model postulates. Thus, the EFRLS algorithm in its present form appears to have a limited scope for applicability.

In the case of Kalman or extended Kalman filter, the matrix P represents the covariance of the state prediction error, and hence the standard deviation of the estimates can be readily computed as the square root of the diagonal elements as the propagation progresses in time. On the other hand, the matrix P appearing in Eq. (7.90) is a data matrix and not the covariance matrix of the state prediction error. Therefore, the standard deviations cannot be obtained directly for EFRLS as for the case of the Kalman filter. As in the case of RLS, the standard deviations can be computed based on Eq. (7.15) using the complete set of data after completing the EFRLS estimation.

IV. Algorithmic Implementation and Software

As in the case of Chapters 4–6, we provide here simple implementations of the software for recursive parameter estimation based on the RLS, FTR, EKF, UKF (in two versions), and EFRLS methods discussed in the foregoing sections; these programs enable closer understanding of the algorithmic steps that we have so far studied in this chapter. Furthermore, it allows us to generate the results that we will discuss here. The source codes (Matlab® m-files) for recursive parameter estimation are provided in the directory /FVSysID/chapter07/.

In the case of the EKF and both versions of the UKF, we have implemented here the algorithms in their so-called standard versions. Numerically more efficient implementations based on UD-factorization are possible. Particularly for UKF, the square-root implementation has a further advantage, because the square-root of the covariance matrix is a part of the algorithm, which is also needed in the computation of the sigma points.^{46,47} Thus, a considerable reduction in the computational overhead is possible. Computational speed can

also be enhanced by vectorizing the matrix operations at some of the steps involved in the weighted sum for mean and covariances by treating the weights as a matrix having the same weights in all the columns. All such extensions are left to the reader. We deal here with standard versions and treating weights as a column vector, because it allows the algorithmic steps given in the text to be traced back and also because it leads to a more readable code.

The program “attas_regLs_RLS” is a simple implementation of the RLS algorithm described in Sec. II.B for a test case which we analyzed in Chapter 6, Sec. IX.B. For recursive estimation based on state space models, the main program “mainRPE” is the starting point. It provides an option to invoke different recursive algorithms, to define the test case to be analyzed, and to call the function for model definition:

test_case	integer flag for the test case
method	flag to choose the RPE method
EKF:	extended Kalman filter
UKF:	unscented Kalman filter (simplified case of additive process and measurement noise, i.e., $x_k^a = [x_k^T \Theta_k^T]^T$)
UKFaug:	augmented unscented Kalman filter (general case of state augmentation through process and measurement noise, i.e., $x_k^a = [x_k^T \Theta_k^T w_k^T v_k^T]^T$)
EFRLS:	state estimator without knowledge of noise covariances
FTR:	Fourier transform regression
ALL:	sequential application of EKF, UKF, UKFaug, EFRLS, and FTR

Depending upon the selected method, a few constants may be necessary. For example, in the case of the EFRLS it is necessary to specify the forgetting factor “lambda”; for FTR it is necessary to specify the update rate for FFT and the range of frequencies to be accounted for, and in the case of UKF or UKFaug it is necessary to specify the scaling parameters α , β , κ . It is possible to apply any one of the five algorithms to the specified problem by specifying the flag “method” appropriately. Otherwise, setting it to ALL will apply each of them in sequence; this option will be used for the examples presented in the next section, because it allows comparative results to be generated more readily. The choice of the method will depend upon the type of the system model.

Different system models to be analyzed are denoted through the integer flag “test_case,” which is to be uniquely associated with the state space model and with a user-provided interface in the form of a function defining the model parameters, flight data and other relevant details. All but the FTR method need functions to code the defined state space model. The model definition comprises the following information:

state_eq	function name (to code the right hand side of the state equations)
obser_eq	function name (to code the right hand side of the observation equations)

Nx	number of state variables (n_x)
Ny	number of output variables (n_y)
Nu	number of input variables (n_u)
NparSys	number of system parameters
Nparam	total number of system and bias parameters

The above list shows on the left-hand side the variable names used in the program mainRPE and in other functions called therein, followed by the description and the notation used to denote these variables in the text. In the case of the FTR, which is based on a least squares approach, there are no state equations. Accordingly, the above two functions are not necessary.

The functions to code the state and observation equations used in the UKFaug are extended versions of the above two functions. They are denoted “state_eq_xyNoise” and “obser_eq_xyNoise,” respectively. In these functions, the noise vectors obtained from the sigma points appear explicitly in the state and observation equations. The major part remains the same as in the equations “state_eq” and “obser_eq”; only the noise needs to be incorporated appropriately.

It is also necessary to specify the starting values for the parameters, integer flags to indicate whether the particular parameter is free or fixed, initial conditions, the perturbations for numerical approximations, and covariances of measurement noise, augmented states and state propagation error:

param	starting values for unknown parameters (Θ_0)
parFlag	flags to define free and fixed parameters = 1: free parameters (to be estimated) = 0: fixed parameters (not to be estimated)
NparID	total number of unknown parameters, that is, nonzero elements of parFlag (n_q)
x0	initial conditions on state variables x
delxa	perturbation in augmented states for numerical approximations of system matrices
rr	diagonal elements of measurement noise covariance matrix
qq	diagonal elements of process noise covariance matrix
p0	initial state propagation error covariance matrix

The number of augmented states Nxp is given by Nx + NparID, and the augmented states by $x_{ah} = [x_0^T \ \Theta_0^T]^T$. The qq and p0 are to be specified for the Nxp augmented states.

The flight data to be analyzed is also loaded in the model definition function, and requires specification or assignment of the following information:

Ndata	number of data points
dt	sampling time
Z(Ndata, Ny)	flight data for measured outputs (N, n_y)
Uinp(Ndata, Nu)	flight data for measured control inputs (N, n_u)

While loading the flight data in the arrays $Z(Ndata,Ny)$ and $Uinp(Ndata,Nu)$, the units of the variables must match with those used in the postulated model and programmed in the state and observation functions “state_eq” and “obser_eq,” respectively. The model specification in terms of $(Nx, Ny, Nu, NparSys, Nxp)$ must also match that coded in the above two user functions for the postulated model.

At the end of the state cum parameter estimation, depending upon the method and the model, plots are generated of the time histories, convergence of system states or of parameters as appropriate. Additionally, comparison is made of the parameter estimates provided by the different estimation methods.

V. Examples

We consider here four examples to demonstrate some of the recursive algorithms detailed in the previous sections. The first example pertains to a model in an equation error form (i.e., without state equations), whereas the other three pertain to state-space models. The second and the third examples are based on linear models in terms of dimensional derivatives. The fourth example considers a nonlinear model pertaining to lift, drag, and pitching moment. Based on the last three examples, the various RPE algorithms are evaluated considering the following issues: 1) computational effort required, 2) parameter convergence to reference values and the time taken, 3) robustness in the presence of noise, and 4) computed variance of the estimation error. The estimates obtained applying these algorithms are compared against those obtained from other offline nonrecursive methods discussed in previous chapters.

The data analyzed in the first three examples was gathered from flight tests carried out with the research aircraft ATTAS.⁴⁸ Several repeat runs were performed and analyzed; we restrict ourselves to just one typical dataset in each case. In the fourth case we analyze data that was gathered with test aircraft HFB-320. The flight data analyzed in these cases is supplied to the reader as sample flight data for verification and to try out other options, and possibly to verify software for modification or development. These data files are found under the directory `/FVSysID/flt_data/`.

A. Estimation of Aerodynamic Parameters of a Linear Model

To demonstrate the RLS method, we turn our attention to the example considered in Chapter 6, Sec. IX.B, in which the LS method was applied to flight data from three concatenated maneuvers performed at 16,000 ft and at 200 kts with the test aircraft ATTAS. The first maneuver is a multistep elevator input exciting the short period motion, the second is an aileron input resulting in a bank-to-bank motion, and the third is a rudder doublet input exciting the Dutch roll motion. The three maneuvers are 25, 30, and 30 s long, respectively. The LS estimates for the aerodynamic model postulated in Eq. (6.76), obtained by running the program “`/FVSysID/chapter06/attas_regLS`,” were provided in Table 6.2.

Now, we run the same case by calling the program “`/FVSysID/chapter07/attas_regLS_RLS`,” which includes the same data pre-processing step as

described in Chapter 6, Sec. IX.B to compute the aerodynamic force and moment coefficients and also the procedures to apply the LS method. These two steps remain unchanged. For each aerodynamic coefficient, the RLS algorithm of Eqs. (7.17)–(7.19) is appended in the new program “attas_regLS_RLS.” The models for the aerodynamic force and moment coefficients consist of

$$\begin{aligned} C_D &= C_{D0} + C_{D\alpha}\alpha \\ C_L &= C_{L0} + C_{L\alpha}\alpha \\ C_m &= C_{m0} + C_{m\alpha}\alpha + C_{mq}q^* + C_{m\delta_e}\delta_e \\ C_Y &= C_{Y0} + C_{Y\beta}\beta \\ C_\ell &= C_{\ell0} + C_{\ell\beta}\beta + C_{\ell p}p^* + C_{\ell r}r^* + C_{\ell \delta_a}\delta_a \\ C_n &= C_{n0} + C_{n\beta}\beta + C_{np}p^* + C_{nr}r^* + C_{n\delta_r}\delta_r \end{aligned} \quad (7.93)$$

As in the case of the least squares method, the aerodynamic derivatives appearing in Eq. (7.93) are estimated by considering the drag, lift, pitching moment, side force, rolling moment, and yawing moment coefficients one at a time. The forgetting factor is chosen to be $\lambda = 1$. As a typical case, the estimates of a few selected parameters are plotted in Fig. 7.3, which shows the RLS estimates as time histories in continuous lines. For comparison purposes,

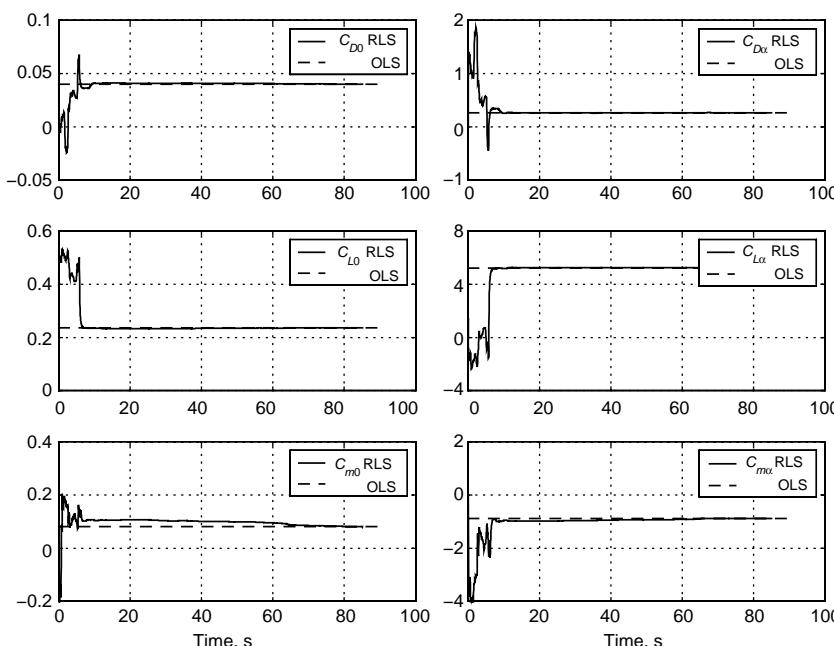


Fig. 7.3 Estimates of lift, drag, and pitching moment parameters applying RLS method.

the one-shot OLS estimates are shown as dotted lines as constant values for the complete duration of the data analyzed.

As expected, the RLS estimates at the end of the maneuver match with OLS estimates. In most cases it can be observed that the convergence is good. For example, as seen in the figure, the estimates of drag and lift parameters converge during the first 25 s, in which the short period motion is excited. Although not shown in the figure, by running the program “attas_regLS_RLS,” it can be verified that the parameters pertaining to the rolling motion show variations during the first 25 s, but they are adjusted once the appropriate information becomes available during the following aileron input maneuver. The parameters of the yawing motion are also estimated during this maneuver, because of the coupling between the rolling and yawing motion. Some further adjustments occur once the information from the rudder input is evaluated. The convergence of the parameters of the pitching moment coefficient, particularly the pitch damping derivative C_{mq} , was rather slow. The above discussion, the effect of the forgetting factor, and plots of the other aerodynamic parameters appearing other force and moment coefficients can be verified by running the test case as just described.

For the reasons already elaborated in Sec. II.A, the standard deviations are not plotted in Fig. 7.3 as the estimates evolve in time. Instead, they are computed after the estimation based on the complete length of data using Eq. (6.21), as done for the least squares method. The standard deviations in the two cases of LS and RLS were practically the same.

B. Longitudinal Motion

A second-order linear model pertaining to the aircraft short-period motion excited through an elevator input is given by

State equations

$$\begin{aligned}\dot{\alpha} &= Z_0 + Z_\alpha \alpha + (1 + Z_q)q + Z_{\delta_e} \delta_e \\ \dot{q} &= M_0 + M_\alpha \alpha + M_q q + M_{\delta_e} \delta_e\end{aligned}\quad (7.94)$$

Observation equations

$$\begin{aligned}\alpha_m &= \alpha \\ q_m &= q\end{aligned}\quad (7.95)$$

where the subscript m in Eq. (7.95) refers to the measured values. The angle of attack, α , and pitch rate, q , are the two state variables; the measured α and q are the two output variables, and elevator deflection δ_e is the control input. $Z()$ and $M()$ denote the dimensional parameters pertaining to vertical force and pitching motion respectively. The unknown parameter vector Θ , consisting of the dimensional derivatives and aerodynamic zero terms, is given by

$$\Theta = [Z_0 \ Z_\alpha \ Z_q \ Z_{\delta_e} \ M_0 \ M_\alpha \ M_q \ M_{\delta_e}]^T \quad (7.96)$$

Thus, the postulated model is defined as follows:

	No. of variables		Function name
States	2	α, q	“xdot_TC11_lon_sp”
Outputs	2	α, q	“obs_TC11_lon_sp”
Inputs	1	δ_e	

The right-hand sides of the state equation (7.94) are programmed in the function “xdot_TC11_lon_sp,” and those of the observation equation (7.95) in “obs_TC11_lon_sp.” The above postulated model is designated as the eleventh test case. Thus, the model definition for this example, provided by the function “mDefCase11.m” called from the main program “mainRPE,” is as follows:

```

test_case = 11; % index for the test case
state_eq = 'xdot_TC11_lon_sp'; % function for state equations
obser_eq = 'obs_TC11_lon_sp'; % function for observation equations
Ny = 2; % number of observation variables
Nx = 2; % number of states
Nu = 1; % number of input (control) variables
NparSys = 8; % number of system parameters
Nxp = Nx + NParSys; % total number of states
(dt = 0.04); % (system states + parameters)
% sampling time

```

Deviating from the above model definition, the state and observation equations used in the UKFaug are coded in “xdot_TC11_lon_sp_xyNoise” and “obs_TC11_lon_sp_xyNoise,” respectively (see elaboration in Sec. IV).

To increase the length of the data segment being processed recursively, two sets of short-period data are cascaded. The data is loaded from the following two files:

```

load ..\f1t_data\fAttasElv1;
load ..\f1t_data\fAttasElv2;

```

and stored in the arrays Z(Ndata,Ny) and Uinp(Ndata,Nu). The storage of data is not essential for recursive algorithms, because states and parameters are updated point-by-point. However, for comparison purposes the various RPE algorithms are run in an offline environment, that is, after the data has been recorded. The stored data allows us to make plots of the time histories as well.

First, we generate the reference values applying the output error method, that is, “/FVSysID/chapter04/ml_oem,” for which we use the same flag `test_case = 11` and the same functions for the state and observation equations, but the following additional information concerning the individual time segments is necessary:

<code>Nzi = 2;</code>	% number of time segments
<code>izhf = [Nts1; Nts1 + Nts2];</code>	% cumulative index

Table 7.1 Parameter estimates from flight data

Parameter	Reference values (OEM)	Estimates				
		EFRLS	FTR	EKF	UKF	UKFaug
Z_0	-0.009 (7.25) ^a	-0.0075	—	-0.0075 (15.3)	-0.0077 (14.6)	-0.0090 (2.78)
Z_α	-0.483 (1.36)	-0.500	-0.477 (3.74)	-0.496 (2.28)	-0.496 (2.24)	-0.484 (0.50)
Z_q	0.104 (5.28)	0.109	0.105 ^b (1.29)	0.101 (9.66)	0.103 (9.31)	0.108 (1.88)
$Z_{\delta e}$	0.676 (1.76)	0.669	0.665 (5.47)	0.647 (3.58)	0.653 (3.48)	0.680 (0.67)
M_0	0.475 (0.3)	0.465	—	0.468 (0.28)	0.468 (0.26)	0.474 (0.10)
M_α	-4.927 (0.33)	-4.846	-4.929 (1.52)	-4.873 (0.26)	-4.873 (0.25)	-4.913 (0.11)
M_q	-2.006 (0.57)	-1.972	-2.031 (2.96)	-1.969 (0.57)	-1.969 (0.54)	-2.014 (0.20)
$M_{\delta e}$	-7.208 (0.35)	-7.129	-7.254 (2.11)	-7.085 (0.38)	-7.081 (0.36)	-7.222 (0.12)

^aValues in parentheses denote relative standard deviation in percent.

^bObtained from FTR estimate $(1 + Z_q)$.

where Nts1 and Nts2 are the number of data points in the two individual time segments and izhf is the cumulative index at which the maneuvers end when they are concatenated. The estimates obtained applying “ml_oem” are summarized in Table 7.1 along with those from the other methods. The “ml_oem” estimates are plotted in Fig. 7.4 as a constant value over the complete duration of data in the figures, which show convergence of the estimates from other methods.

Now, we set the flag “method= ALL” in “/FVSysID/chapter07/mainRPE” to apply the five recursive methods, namely EKF, UKF, UKFaug, EFRLS, and FTR. This yields the results summarized in Table 7.1, which gives average values over the last few data points, typically 10 for the EKF, UKF, UKFaug, and EFRLS, and five for the FTR. Representative values have been specified for the measurement noise covariance matrix (see function “mDefCase11.m”) and the process noise covariance matrix is also specified as diagonal with values of 1×10^{-7} for states and zero for the extended states, that is, parameters, according to Eq. (7.49). In each case the time history match between the flight-measured data and model output was found to be satisfactory; these plots as well as those of the parameter estimates from individual RPE methods are also not included in the text here, but can be easily generated and verified running this case. The estimates of the six derivatives are plotted graphically in Fig. 7.4.

Let us now compare the numerical values and accuracies of the estimates from the various methods. The estimates of all the parameters obtained applying various methods appear plausible. They are within reasonable ranges and are comparable to the reference values provided by the output error method.

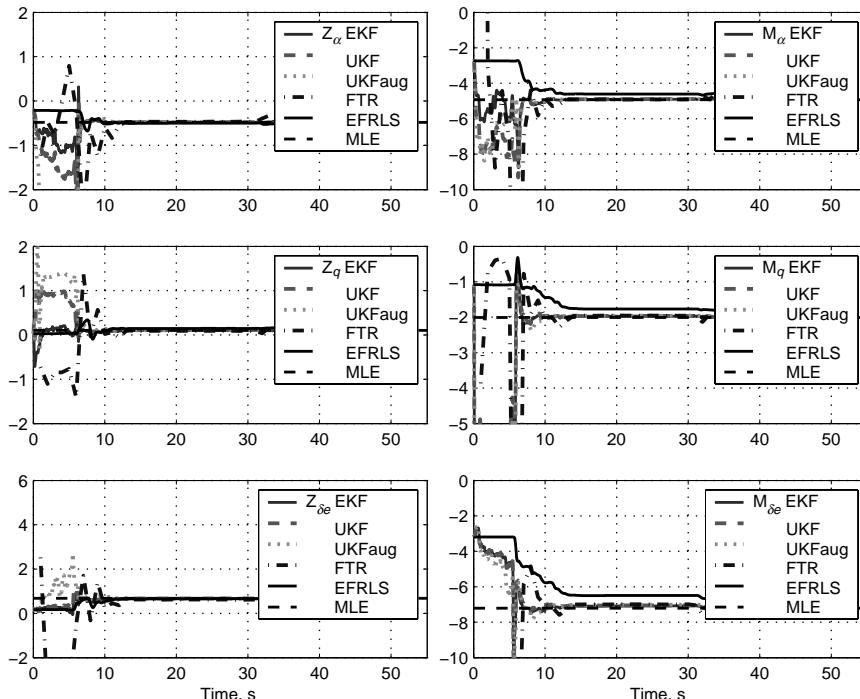


Fig. 7.4 Recursive estimates of longitudinal derivatives.

The standard deviations for some of the estimates by EKF, UKF and FTR are somewhat higher than those for the offline method or those from UKFaug. In an overall sense, the UKFaug estimates compare best with the OEM results.

As already discussed in Sec. II.D in the case of FTR, the aerodynamic bias parameters Z_0 and M_0 are not estimated. Moreover, trim values were also subtracted from the measured data (this is done in the function “ftr_mod” implementing the FTR). The large fluctuations in the estimates prior to initiation of the flight maneuver (at about $t = 6$ s) are due to lack of information in the data content up to then. However, as can be observed from the parameter time history plots, they tend to converge to the reference values as more and more information becomes available in the data.

The robustness of the EKF, UKF, UKFaug, and EFRLS algorithms has been tested by initiating the recursion with arbitrary starting values, whereby the algorithms converged without any numerical difficulties. The parameters are well estimated once the input is applied, here from 6 to 12 s which is followed by the next excitation from 31 to 37 s. During the period between the two maneuvers, which corresponds to flight without any specific control input, the converged estimates show no divergence. The additional excitation leads to hardly any changes in the EKF, UKF, UKFaug, and EFRLS estimates, whereas the FTR estimates are marginally adjusted during the maneuver itself.

C. Lateral-directional Motion

A second-order linear model pertaining to the lateral directional motion excited through aileron and rudder inputs is given by the state equations:

$$\begin{aligned}\dot{p} &= L_p p + L_r r + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r + L_\beta \beta + L_0 \\ \dot{r} &= N_p p + N_r r + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r + N_\beta \beta + N_0\end{aligned}\quad (7.97)$$

and by the observation equations:

$$\begin{aligned}p_m &= p \\ r_m &= r\end{aligned}\quad (7.98)$$

where p is the roll rate, r the yaw rate, β the sideslip angle, δ_a the aileron deflection and δ_r the rudder deflection. In this simplified model, we treat the angle of sideslip as an input variable, and hence it does not appear as a state equation. $L_{(.)}$ and $N_{(.)}$ are the dimensional parameters pertaining to the rolling and yawing motion which are to be estimated. The unknown parameter vector Θ consisting of the dimensional parameters is given by:

$$\Theta = [L_p \ L_r \ L_{\delta_a} \ L_{\delta_r} \ L_\beta \ L_0 \ N_p \ N_r \ N_{\delta_a} \ N_{\delta_r} \ N_\beta \ N_0]^T \quad (7.99)$$

Thus, the postulated model is defined as follows:

	No. of variables	Function name
States	2	p, r “xdot_TC03_lat_pr”
Outputs	2	p, r “obs_TC03_lat_pr”
Inputs	3	$\delta_a, \delta_r, \beta$

The right-hand sides of the state equation (7.97) are programmed in the function “xdot_TC03_lat_pr,” and those of the observation equation (7.98), in “obs_TC03_lat_pr.” The above postulated model is designated as the third test case. Thus, the model definition provided by the function “mDefCase03.m,” which is called from the main program “mainRPE,” is as follows:

```
test_case = 3; % flag for the test case
state_eq = 'xdot_TC03_lat_pr'; % function for state equations
obser_eq = 'obs_TC03_lat_pr'; % function for observation equations
Ny = 2; % number of observation variables
Nx = 2; % number of system states
Nu = 3; % number of input (control) variables
NparSys = 12; % number of system parameters
Nxp = Nx + NParSys; % total number of states
(dt = 0.04); % (system states + parameters)
% sampling time
```

Deviating from the above model definition, the state and observation equations used in the UKFaug are coded in “xdot_TC03_lat_pr_xyNoise” and “obs_TC03_lat_pr_xyNoise,” respectively (see elaboration in Sec. IV).

Parameters pertaining to the lateral-directional motion are estimated from a bank-to-bank maneuver and a Dutch roll motion. As in the case of the previous example, we analyze two sets of cascaded flight data, each with an aileron input followed by a rudder input. The data is loaded from the following two files:

```
load ..\f1t_data\fAttasAilRud1;
load ..\f1t_data\fAttasAilRud2;
```

and stored in the arrays $Z(N_{\text{data}}, N_y)$ and $U_{\text{inp}}(N_{\text{data}}, N_u)$. Since the general procedure of data analysis is similar to that followed for the previous example, we only give here the bare minimum details, because the general remarks are valid in this case too. The reference values are generated applying the output error method, that is, “/FVSysID/chapter04/ml_oem,” by setting flag $\text{test_case} = 3$. The information about the two time segments (N_{zi} and N_{zf}) is defined as in the case of the previous example.

Table 7.2 summarizes the estimates obtained applying the output error method “/FVSysID/chapter04/ml_oem,” along with those from the recursive methods.

Table 7.2 Estimates of parameters pertaining to lateral-directional motion

Parameter	Reference values (OEM)	Estimates				
		EFRLS	FTR	EKF	UKF	UKFaug
L_p	-2.067 (0.21) ^a	-2.094	-2.046 (1.70)	-2.077 (0.16)	-2.072 (0.15)	-2.078 (0.12)
L_r	1.029 (0.84)	1.051	1.041 (6.96)	1.021 (0.64)	1.019 (0.62)	1.042 (0.48)
$L_{\delta\alpha}$	-6.306 (0.17)	-6.391	-6.367 (1.23)	-6.349 (0.13)	-6.334 (0.12)	-6.324 (0.10)
$L_{\delta r}$	1.145 (1.84)	1.238	1.277 (16.59)	1.195 (1.43)	1.195 (1.38)	1.065 (1.17)
L_β	-3.844 (0.38)	-3.909	-3.786 (3.28)	-3.802 (0.29)	-3.800 (0.28)	-3.835 (0.22)
L_0	0.010 (3.45)	0.0085	—	0.0103 (2.76)	0.0104 (2.65)	0.0093 (2.16)
N_p	-0.184 (1.34)	-0.197	-0.162 (6.97)	-0.171 (1.16)	-0.170 (1.14)	-0.176 (0.26)
N_r	-0.486 (0.96)	-0.458	-0.433 (5.44)	-0.425 (0.97)	-0.425 (0.95)	-0.460 (0.20)
$N_{\delta a}$	-0.391 (1.64)	-0.408	-0.352 (7.21)	-0.380 (1.25)	-0.378 (1.23)	-0.383 (0.31)
$N_{\delta r}$	-1.761 (0.63)	-1.773	-1.747 (3.94)	-1.716 (0.61)	-1.719 (0.60)	-1.764 (0.13)
N_β	3.031 (0.30)	3.029	3.074 (1.31)	3.029 (0.23)	3.031 (0.22)	3.025 (0.06)
N_0	0.0346 (0.45)	0.035	—	0.034 (0.55)	0.034 (0.54)	0.034 (0.09)

^aValues in parentheses denote relative standard deviation in percent.

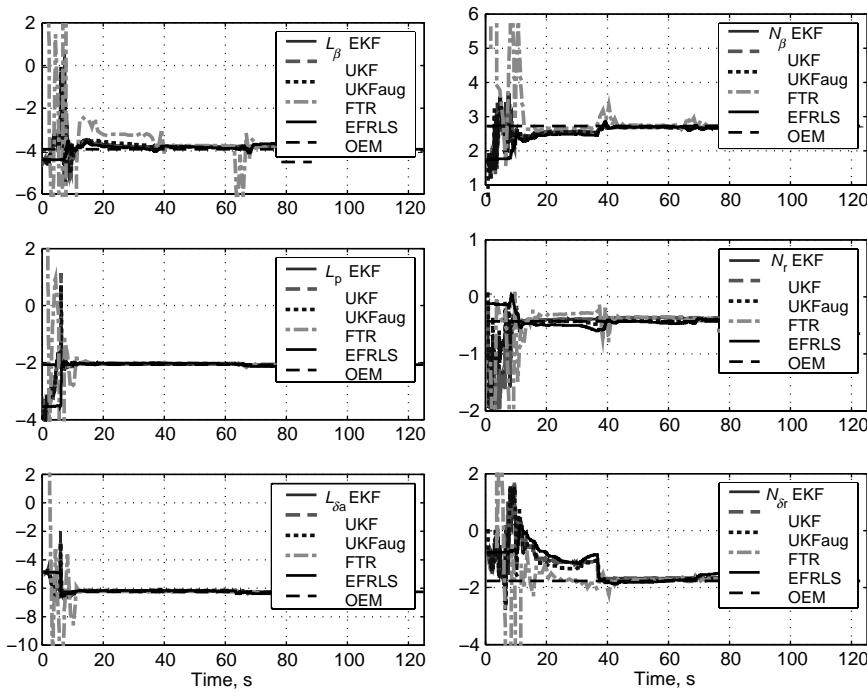


Fig. 7.5 Recursive estimates of lateral-directional derivatives.

The relative standard deviations of the estimates are provided in brackets. The OEM estimates are the reference values for evaluating the RPE algorithms. They are plotted as a constant value over the complete duration of data in Fig. 7.5.

Having generated the reference values, we now apply the RPE algorithms “/FVSysID/chapter07/mainRPE” by setting the flags “method = ALL” and “test_case = 3.” The results are provided in Table 7.2, giving average values over the last few data points. In the case of FTR, the aerodynamic bias parameters L_0 and N_0 are not estimated and trim values were subtracted from the measured data (see function “ftr_mod” implementing the FTR). As in the case of the previous example, representative values were specified for the measurement noise covariance matrix (see function “mDefCase03.m”) and the process noise covariance matrix is also specified as diagonal with values of 1×10^{-7} for states and zero for the extended states, that is, parameters, according to Eq. (7.49).

The same characteristic behavior of the five RPE algorithms discussed in Sec. V.C repeated in the present case. Figure 7.5 gives the plots of some selected derivatives of the lateral-directional motion. From Table 7.2 and Fig. 7.5 we note that the estimates from the various RPE algorithms compare reasonably well with the reference values from OEM. The performance of all the RPE algorithms was satisfactory and encountered no numerical difficulties.

D. Estimation of Nondimensional Longitudinal Derivatives

The fourth example pertains to determination of the lift, drag, and pitching moment coefficients of the research aircraft HFB-320 considered in Chapter 5, Sec. XI.B. As already pointed out there, the system equations (5.86) in terms of variables in the stability axes (V, α) contain not only the common trigonometric and multiplicative nonlinearities, but in addition, the variable dynamic pressure $\bar{q}(=1/2\rho V^2)$, which multiplies all of the aerodynamic derivatives, introducing an additional nonlinearity. The observations in Eq. (5.88) are coupled through C_X and C_Z . The maneuver analyzed is the longitudinal motion excited through a multistep elevator input, resulting in short period motion and a pulse input leading to phugoid motion. The state vector is $x = [V \alpha \theta q]^T$ and the unknown parameters appearing in Eq. (5.87) are $\Theta = [C_{D0} C_{DV} C_{Da} C_{L0} C_{LV} C_{La} C_{m0} C_{mv} C_{ma} C_{mq} C_{mde}]^T$.

Since it is now attempted to estimate the augmented state vector $x^a = [x^T \Theta^T]^T$ subject to the model postulated in Eqs. (5.86)–(5.89), we require recursive methods that cater for nonlinear state space models. Accordingly, the choice of methods reduces to EKF, UKF, and UKFaug, because the RLS procedure is based on a different model structure, FTR is applicable to linear models only, and the EFRLS algorithm presented in Sec. III.C does not directly cater for nonlinear state and observation equations. For the EKF, UKF, and UKFaug methods the postulated model is defined as follows:

	No. of variables	Function name
States	4	V, α, θ, q “xdot_TC04_hfb_lon”
Outputs	7	$V, \alpha, \theta, q, \dot{q}, a_x, a_z$ “obs_TC04_hfb_lon”
Inputs	2	δ_e, F_e

Thus, the model definition for this example provided by the function “/FVSysID/chapter07/mDefCase04.m,” called from the main program “mainRPE,” is as follows:

```

test_case = 4; % integer flag for the test case
state_eq = 'xdot_TC04_hfb_lon'; % function for state equations
obser_eq = 'obs_TC04_hfb_lon'; % function for observation equations
Nx = 4; % number of states
Ny = 7; % number of observation variables
Nu = 2; % number of input (control) variables
NparSys = 11; % number of system parameters
Nparam = NparSys + Nx; % total number of parameters
dt = 0.1; % sampling time

```

Deviating from the above model definition, the state and observation equations used in the UKFaug are coded in “xdot_TC04_hfb_lon_xyNoise” and “obs_TC04_hfb_lon_xyNoise,” respectively (see elaboration in Sec. IV).

In Chapter 5, Sec. XI.B it has been demonstrated that the data being analyzed contain atmospheric turbulence and that the filter error method “ml_fem”

Table 7.3 Estimates of parameters pertaining to lateral-directional motion

Parameter	Filter error method (FEM)	RPE methods		
		EKF	UKF	UKFaug
C_{D0}	0.123 (2.45) ^a	0.124 (2.50)	0.123 (2.64)	0.124 (2.55)
C_{DV}	-0.0645 (3.95)	-0.0652 (4.01)	-0.0642 (4.27)	-0.0653 (4.09)
$C_{D\alpha}$	0.320 (2.26)	0.319 (2.33)	0.319 (2.40)	0.316 (2.37)
C_{L0}	-0.0929 (21.1)	-0.085 (23.5)	-0.087 (22.9)	-0.099 (20.1)
C_{LV}	0.149 (11.1)	0.144 (11.7)	0.147 (11.4)	0.157 (10.6)
$C_{L\alpha}$	4.328 (1.08)	4.303 (1.14)	4.289 (1.14)	4.303 (1.13)
C_{m0}	0.112 (3.27)	0.112 (4.28)	0.112 (4.29)	0.115 (3.40)
C_{mv}	0.0039 (82.1)	0.0046 (90.5)	0.0045 (92.2)	0.0022 (154)
$C_{m\alpha}$	-0.968 (1.12)	-0.971 (1.54)	-0.970 (1.54)	-0.983 (1.24)
C_{mq}	-34.710 (2.27)	-34.937 (2.85)	-35.363 (2.82)	-35.098 (2.27)
$C_{m\delta e}$	-1.529 (1.27)	-1.533 (1.65)	-1.539 (1.65)	-1.552 (1.31)

^aValues in parentheses denote relative standard deviation in percent.

provided better results than the output error method. The estimates obtained running the program “/FVSysID/chapter05/ml_fem” for test_case = 4 are provided in Table 7.3. To enable comparison of the recursive estimates with those from the offline filter error method, we specify the process and measurement noise covariances obtained by running the filter error method “ml_fem.” This was carried out in “/FVSysID/chapter07/mDefCase04.m,” whereby it may be recalled that the filter error method is based on the mixed continuous-discrete system, and accordingly the factor of $1/\Delta t$ has been taken care of when specifying the process noise covariance.

We observe that the recursive methods EKF, UKF, and UKFaug converge well without encountering any numerical difficulties. Figure 7.6 shows convergence plots of a few selected parameters. The error bounds are not shown in this figure, but can be verified easily by running this test case. The numerical values of all the derivatives appear plausible and compare well with those from the filter error method. Looking at the numerical estimates, at their convergence plots and at the time histories of the model outputs, it appears that the performance of these methods is comparable.

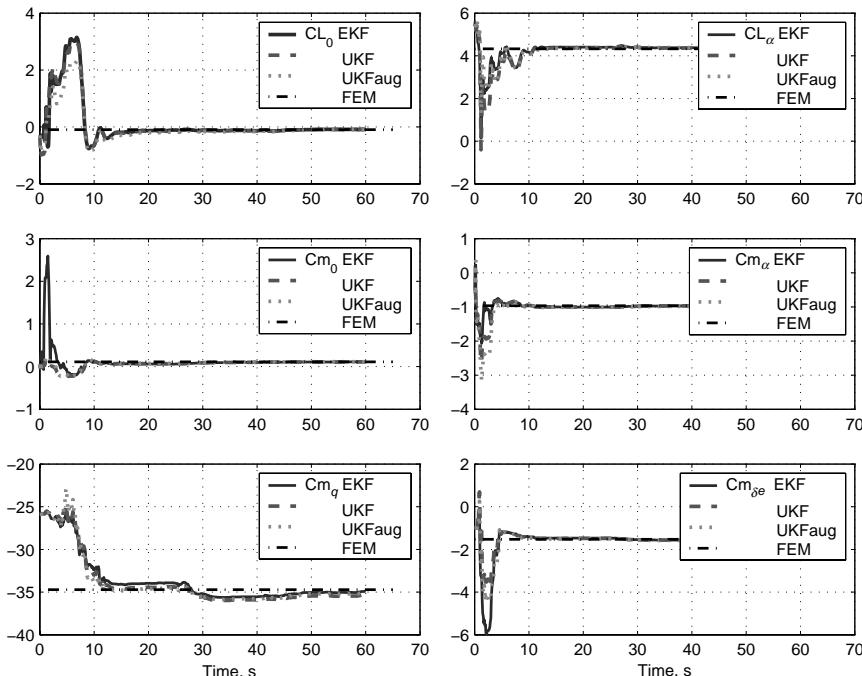


Fig. 7.6 Recursive estimates from nonlinear state space model.

By running the test case, it can be verified that the time history plots from EKF show a good match between the measured and model estimated responses. The match for the UKF and UKFaug is also good, except for the large deviations in some of the variables for the first data point only. For the reasons discussed in Sec. III.B, these deviations are attributed the large initial covariance matrix P_0^a . We note that the state variables α , θ , and q in the programmed model are in terms of radians and radians per second, whereas the corresponding initial covariances are specified as in 10. The subsequent parameter convergence is, however, not affected. More realistic initial covariance values in the case of UKF eliminates these discrepancies in the match and yield very similar final estimates. The same large initial covariance values were used in all the three cases to bring out this specific aspect and also to allow comparison of results starting from the same conditions.

In this section we have not presented the results provided by the EFRLS method for the reasons already elaborated, namely the method does not directly cater for nonlinear state and observation equations. By setting method = EFRLS, by changing the if-statement in the main program “mainRPE.m” appropriately, and running the same, it can be verified that the convergence is far from satisfactory.

VI. Comparative Evaluation of Recursive Algorithms

Based on the four examples presented here, we can now try to arrive at a consolidated performance evaluation and some general conclusions. In general, the various recursive algorithms work satisfactorily in practice on real flight data, although care had to be exercised in each case.

The performance of the RLS or that of the EFRLS is affected by the forgetting factor used to de-emphasize the older data. The forgetting factor leads to a faster adaptation of variations in the system, but a nonoptimal choice can lead to a poor performance or divergence. The EFRLS performed reasonably well when applied to the second and the third examples (see Tables 7.1 and 7.2). In these cases the observation model includes only the states, and the state space model is linear. In the fourth example, the performance of EFRLS was very poor. This is attributed to the fact that the state space model is nonlinear and specifically to the observation equations, which include not only states, but also other nonlinear terms and contributions from exogenous inputs. The EFRLS as implemented here does not cater for this general case. As already elaborated, the RLS and EFRLS methods do not directly yield the accuracies of the estimates as the states are propagated; the standard deviations are usually obtained once the recursive estimation is completed.

In the FTR method there are no tuning parameters, but trim values have to be removed before computing the discrete Fourier transforms, and zero aerodynamic terms cannot be estimated. Removal of trim values from the measured data is a trivial numerical computation and inability to estimate zero aerodynamic terms is not a serious limitation. The FTR method is found to be consistent and outperforms the other methods. With lesser computational requirements and with no necessity to tune parameters, FTR appears to be more appealing for online applications than the EFRLS, EKF, UKF, and UKFaug. It is, however, applicable to linear model postulates only.

The EKF, UKF, and UKFaug, by their basic formulation, automatically cater to linear or nonlinear state space models. Furthermore, additional observation variables in any form, for example the angular accelerations $\dot{\phi}$ and \dot{r} or other output variables such as flow angles, can be easily included in the estimation, which will lead to better convergence and accuracy of the estimates.

It is noticed that the UKF performance was comparatively insensitive to the choice of the scaling factors β and κ . They help to improve the higher-order contributions, which are negligible for the models considered here. The two UKF algorithms are more involved than the EKF procedure. Let us now compare the computational time required by the three filtering algorithms. For the second example discussed in Sec. V.B, the EKF, UKF and UKFaug required on average 4, 8.8, and 11.8 s, respectively; for the third example presented in Sec. V.C the required time was roughly 14.2, 32.4, and 39 s, whereas for the more complex fourth model 3.4, 7.6, and 14.3 s were needed. The computational time depends upon the size of the model and on the complexity of the state equations. The above comparative time requirements indicate that the computational load of the more advanced UKF in its simplified version is at least twice that of the conventional EKF, whereas that of the UKFaug for a more general case of noise affecting the system nonlinearly three to four times.

The three nonlinear state estimators also worked when the initial values were fairly far from the final estimates. Only in a very few cases, when the initial errors were very large and the nonlinearities were very strong, did the UKF tend to converge, whereas the EKF showed convergence problems. Interestingly, the performance of the EKF and that of the UKF in terms of the convergence and parameter accuracies was very similar for the test cases presented here as well as for several others. From Tables 7.1 and 7.2 it is observed that the standard deviations of the estimates were consistently lower for the UKF than for the EKF case, but only marginally. For these two cases, they conform to the concept of improving accuracy through the sigma point propagation, but the same cannot be said for the last example presented in Table 7.3, as there is no clear trend that can be observed. In such a realistic practical case with moderate nonlinearities and non-negligible process and measurement noise, it is not obvious whether any one of these methods performs better than the others. However, we recall that the process and measurement noise covariances were optimally prespecified for the filtering methods, whereas the filter error method estimates noise covariances along with the system parameters. In the absence of accurate knowledge about these noise characteristics, the performance of EKF, UKF, and UKFaug may be affected. In general, it was observed that the performance of these three state estimators was not overly sensitive to noise covariances, and reasonable values are usually sufficient. As already pointed out earlier in this chapter, tuning of noise covariances is not simple.

Thus, the UKF, although theoretically quite interesting because it eliminates the flaws in EKF resulting from the linearization used in the propagation, did not show any significant advantages in terms of the convergence or achieved accuracy over the EKF. Based on our investigations, it appears that the EKF will most likely serve the purpose of estimating aerodynamic parameters from flight data. Such a conclusion has also been reached in another application as well.⁴⁹

VII. Concluding Remarks

In this chapter we have discussed several basic issues related to the recursive parameter estimation, which processes the data point-by-point, and are suitable for online applications. Both pros and cons of the basic methodology have been presented in general, as well as critical numerical aspects are elaborated for a few specific algorithms. The techniques addressed here belong to the two broad categories, namely least squares and nonlinear filtering approach. A comparative study has been carried out on flight data analyzed in an offline environment to estimate dimensional derivatives pertaining to the short period and lateral-directional motion, as well as the nondimensional derivatives pertaining to lift, drag and pitching moment coefficients. The results from the standard offline, iterative methods form the basis for evaluating the performance of these recursive methods. The various RPE algorithms detailed in this chapter, namely 1) extended Kalman filter, 2) unscented Kalman filter, 3) extended forgetting factor recursive least squares, and 4) Fourier transform regression, were evaluated for convergence, accuracy, and numerical effort. A consolidated

performance evaluation together with general conclusions has been presented which helps selection of a suitable choice of method in a given case.

Balancing between the advantages, disadvantages and limitations of recursive parameter estimation, in general, we tend to favor the offline estimation methods discussed in Chapters 4–6 which are much more extensive, accurate, and robust. They are able to cover a more complex and broad spectrum of problems. The near-real-time processing performed onboard using offline algorithms on stretches of data may be an alternative. Some of these ideas were elaborated in this chapter. In some specific applications, however, recursive algorithms will be necessary, for example when certain action is to be taken immediately in real-time, affecting the aircraft behavior in flight, for example fault detection followed by adaptive control and control reconfiguration. This choice between the off-line nonrecursive and on-line recursive approach in a specific case underlines the fundamental philosophy of system identification, namely we should first define the goal and the most appropriate methods should be chosen accordingly.

References

- ¹Ljung, L. and Söderström, T., *Theory and Practice of Recursive Identification*, The MIT Press, Cambridge, MA, 1983.
- ²Haykin, S., *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, NJ, 1991.
- ³Young, P., *Recursive Estimation and Time-Series Analysis*, Springer, Berlin, 1984.
- ⁴Plackett, R. L., "Some Theorems in Least Squares," *Biometrika*, Vol. 37, No. 1/2, 1950, pp. 149–157.
- ⁵Sidar, M., "A Recursive On-Line Estimation Method with Application to Aircraft Dynamics Parameter Identification," *Israel Journal of Technology*, Vol. 14, 1976, pp. 56–65.
- ⁶"Mini-Issue on NASA's Advanced Control Law Program for the F-8 DFBW Aircraft," *IEEE Transactions on Automatic Control*, Vol. AC-22, No. 5, 1977, pp. 752–804.
- ⁷Ward, D. G., Monaco, J. F., and Bodson, M., "Development and Flight Testing of a Parameter Identification Algorithm for Reconfigurable Control," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, 1998, pp. 948–956.
- ⁸Morelli, E. A., "Real-Time Parameter Estimation in the Frequency Domain," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 5, 2000, pp. 812–818.
- ⁹Song, Y., Campa, G., Napolitano, M., Seanor, B., and Perhinschi, M. G., "On-Line Parameter Estimation Techniques Comparison Within a Fault Tolerant Flight Control System," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 3, 2002, pp. 528–537.
- ¹⁰Bauer, J. E., Crawford, D. B., Gera, J., and Andrisani, D., "Real-Time Comparison of X-29A Flight Data and simulation Data," *Journal of Aircraft*, Vol. 26, No. 2, 1989, pp. 117–123.
- ¹¹Quanwei, J. and Qiongkang, C., "Dynamic Model for Real-Time Estimation of Aerodynamic Characteristics," *Journal of Aircraft*, Vol. 26, No. 4, 1989, pp. 315–321.
- ¹²Keller, B. S., "On-line Physical Parameter Identification and Adaptive Control of Launch Vehicle," Ph. D. Thesis, Stanford University, Stanford, MA, March 1993.
- ¹³Laban, M., "On-Line Aircraft Aerodynamic Model Identification," Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, May 1994.

- ¹⁴Chu, Q. P., Mulder, J. A., and van Woerkom, P. T. L. M., "Modified Recursive Maximum Likelihood Adaptive Filter for Nonlinear Aircraft Flight-Path Reconstruction," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 6, 1996, pp. 1285–1295.
- ¹⁵Zhu, Y., "Efficient Recursive State Estimator for Dynamic Systems without Knowledge of Noise Covariances," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 35, No. 1, 1999, pp. 102–114.
- ¹⁶Basappa, K., and Jategaonkar, R. V., "Evaluation of Recursive Methods for Aircraft Parameter Estimation," AIAA Paper 2004-5063, Aug. 2004.
- ¹⁷Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W., *Applied Linear Regression Models*, 3rd ed., Richard D. Irwin Inc., Burr Ridge, IL, 1996.
- ¹⁸Atkeson, C. G., Moore, A. W., and Schaal, S., "Locally Weighted Learning for Control," *Artificial Intelligence Review*, Vol. 11, 1997, pp. 75–113.
- ¹⁹Seanor, B., Song, Y., Napolitano, M. R., and Campa, G., "Comparison of On-Line and Off-Line Parameter Estimation Techniques Using the NASA F/A-18 HARV Flight Data," AIAA Paper 2001-4261, Aug. 2001.
- ²⁰Moes, T. R., Smith, M. S., and Morelli, E. A., "Flight Investigation of Prescribed Independent Surface Excitations for Real-Time Parameter Identification," AIAA Paper 2003-5702, Aug. 2003.
- ²¹Morelli, E. A., "High Accuracy Evaluation of the Finite Fourier Transform Using Sampled Data," NASA TM 110340, June 1997.
- ²²Klein, V., "Aircraft Parameter Estimation in the Frequency Domain," AIAA Paper 78-1344, Aug. 1978.
- ²³Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME—Journal of Basic Engineering*, Vol. 82, Series D, March 1960, pp. 35–45.
- ²⁴Kashyap, R. L., "A New Method of Recursive Estimation in Discrete Linear Systems," *IEEE Transactions on Automatic Control*, Vol. AC-15, No. 1, 1970, pp. 18–24.
- ²⁵Chen, R. T. N., Eulrich, B. J., and Lebacqz, J. V., "Development of Advanced Techniques for the Identification of V/STOL Aircraft Stability and Control Parameters," Cornell Aeronautical Lab., Inc., Report No. BM-2820-F-1, 1971.
- ²⁶Eulrich, B. J. and Rynaski, E. G., "Identification of Nonlinear Aerodynamic Stability and Control Parameters at High Angles of Attack," AGARD CP-172, May 1975, Paper 2.
- ²⁷Klein, V. and Schiess, J. R., "Compatibility Check of Measured Aircraft Responses Using Kinematic Equations and Extended Kalman Filter," NASA TN D-8514, Aug. 1977.
- ²⁸Speyer, J. L. and Crues, E. Z., "On-Line Aircraft State and Stability Derivative Estimation Using the Modified-Gain Extended Kalman Filter," *Journal of Guidance and Control*, Vol. 10, No. 3, 1987, pp. 262–268.
- ²⁹Jategaonkar, R. V. and Plaetschke, E., "Estimation of Aircraft Parameters Using Filter Error Methods and Extended Kalman Filter," DFVLR-FB 88-15, March 1988.
- ³⁰Jategaonkar, R. V. and Plaetschke, E., "Algorithms for Aircraft Parameter Estimation Accounting for Process and Measurement Noise," *Journal of Aircraft*, Vol. 26, No. 4, 1989, pp. 360–372.
- ³¹Kokolios, A., "Use of a Kalman Filter for the Determination of Aircraft Aerodynamic Characteristics from Flight Test Data," AIAA Paper 94-0010, Jan. 1994.
- ³²Plaetschke, E., Jategaonkar, R. V., Rohlf, D., and Weiss, S., "Methoden zur Schätzung der Parameter eines Flugzeugs im Post-Stall-Bereich," *Proceedings of the Symposium on 'Sicherheit im Luftverkehr'*, Technical University of Braunschweig, Germany, Sept. 13–15, 1994, pp. 297–306 (in German).

- ³³Garcia-Velo, J. and Walker, B. K., "Aerodynamic Parameter Estimation for High Performance Aircraft Using Extended Kalman Filtering," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 6, 1997, pp. 1257–1259.
- ³⁴Myers, K. A. and Tapley, B. T., "Adaptive Sequential Estimation with Unknown Noise Statistics," *IEEE Transactions on Automatic Control*, Vol. 21, No. 4, 1976, pp. 520–523.
- ³⁵Morelli, E. A., "Estimating Noise Characteristics from Flight Test Data Using Optimal Fourier Smoothing," *Journal of Aircraft*, Vol. 32, No. 4, 1995, pp. 689–695.
- ³⁶Ljung, L., "Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems," *IEEE Transactions on Automatic Control*, Vol. AC-24, No. 1, 1979, pp. 36–50.
- ³⁷Julier, S., Uhlmann, J., and Durrant-Whyte, H. F., "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, 2000, pp. 477–482.
- ³⁸Wan, E. A. and van der Merwe, R., "The Unscented Kalman Filter for Nonlinear Estimation," *Proceedings of the IEEE Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control AS-SPCC*, Lake Louise, Alberta, Canada, 1–4 Oct. 2000, pp. 153–158.
- ³⁹van der Merwe, R., Wan, E. A., and Julier, S., "Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion—Applications to Integrated Navigation," AIAA Paper 2004-5120, Aug. 2004.
- ⁴⁰Julier, S. and Uhlmann, J. K., "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, Vol. 92, March 2004, pp. 401–422.
- ⁴¹Wu, Y., Hu, D., Wu, M., and Hu, X., "Unscented Kalman Filtering for Additive Noise Case: Augmented Versus Nonaugmented," *IEEE Signal Processing Letters*, Vol. 12, No. 5, 2005, pp. 357–360.
- ⁴²Lee, D.-J. and Alfriend, K. T., "Adaptive Sigma Point Filtering for State and Parameter Estimation," AIAA Paper 2004-5101, Aug. 2004.
- ⁴³Bousson, K. and Paglione, P., "On-Line Aircraft Stability Derivative Estimation," *Proceedings of the 2001 Aerospace Congress*, Society for Automotive Engineers, Paper SAE 2001-01-2982, Sept. 2001.
- ⁴⁴Pouliez, A. D. and Stavrakakis, G. S., *Real Time Fault Monitoring of Industrial Processes*, Kluwer Academic, Dordrecht, 1994.
- ⁴⁵Escobet, T. and Travé-Massuyès, L., "Parameter Estimation Methods for Fault Detection and Isolation," Bridge Workshop, Via Lattea, Italy, 5–9 March 2001, pp. 1–11; also published as LAAS Report No. 01471, Laboratory for Analysis and Architecture of Systems (LAAS), Toulouse, France.
- ⁴⁶van der Merwe, R., Wan, E. A., and Julier, S., "The Square-Root Unscented Kalman Filter for State and Parameter Estimation," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001, pp. 3461–3464.
- ⁴⁷Brunke, S. and Campbell, M. E., "Square Root Sigma Point Filtering for Real-Time, Nonlinear Estimation," *Journal of Guidance and Control and Dynamic*, Vol. 27, No. 2, 2004, pp. 314–317.
- ⁴⁸Jategaonkar, R. V., "Identification of the Aerodynamic Model of the DLR Research Aircraft ATTAS from Flight Test Data," DLR FB 90-40, Sept. 1990.
- ⁴⁹Wendel, J., Maier, A., Metzger, J., and Trommer, G. F., "Comparison of Extended and Sigma-Point Kalman Filters for Tightly Coupled GPS/INS Integration," AIAA Paper 2005-6055, Aug. 2005.

Chapter 8

Artificial Neural Networks

I. Introduction

IN THE preceding chapters we focused on parameter estimation methods suitable for phenomenological models based on the physical insight. Developing such models, which we usually prefer because it leads to better understanding of the underlying physics, can be highly demanding. This is particularly the case when the phenomenon being investigated is highly nonlinear, for example, in our case, aerodynamic effects at separated flow conditions. Since we wish to estimate and validate such models from measured input–output data, the artificial neural networks (ANN) provide an alternative approach to model building.^{1–7} The ANNs are neuroscience-inspired computational tools, extensively used for pattern recognition. They provide a general framework for nonlinear functional mapping of the input–output subspace, and as such are part of the system identification methodology which deals implicitly with the measured input–output data. The ANNs, or simply termed NNs, are alternatively called computational neural networks to clearly differentiate them from the biological neural networks. They are also referred to as “universal approximators” or “connectionist models.”

Historically, the concept of neural networks can be traced to McCulloch and Pitts, a neuroanatomist and a mathematician, who showed in 1943 that principally any arithmetic or logical function can be computed using a network consisting of simple neurons.⁸ The neurons were binary switches, which were triggered in the absence of any inhibitory input when the sum of excitatory inputs exceeded a threshold. This led to realization of the Boolean functions such as “AND,” “OR,” or “XOR,” the basic building blocks of a network. The first learning rule was subsequently formulated in 1949 by Hebb.⁹ Pursuing these findings, the first perceptron, a single neuron associated with an effectiveness (weight) which could be adjusted, was developed in 1958 by Rosenblatt, a psychologist, who is widely regarded as the founder of the neural networks.¹⁰ He also developed learning procedure based on an error-correction in the Hebbian sense and applied perceptrons for pattern recognition. The scope of applications of such perceptrons was limited mainly because the learning was restricted to linearly separable patterns. Introduction of hidden layers increased the efficiency; however it posed the problem of deriving a suitable learning rule that required a procedure to compute the desired output, which was known only at the output layer and not for the hidden layer. The major advancement was made in

1974 by Werbos, who developed the technique, called back-propagation, which allowed adjustment of weights in a hidden layer.^{11,12} After more than a decade the procedure was re-invented in 1986 by Rumelhart et al.¹³ It turns out that back-propagation is the same as the Kelly–Bryson gradient algorithm used in the discrete-time optimal control problems.¹⁴ Back-propagation is an iterative procedure that allows determination of weights of an arbitrarily complex neural network.

The unique feature of ANNs is its highly interconnected structure spread over multiple layers, each with a large number of simple processing elements. A typical ANN consists of an input layer, one or more hidden layers, and an output layer. The number of nodes (i.e., neurons or processing elements) in the input and output layers is automatically fixed through the numbers of input–output variables, that is, the data subspace to be matched, whereas the number of nodes in the hidden layers varies from case to case. The processing elements in the hidden layer are invariably continuous, nonlinear functions; these sigmoidal (S-shape) functions basically provide the global approximation capability to ANNs. The elements in the output layers may be linear or nonlinear. The ANNs are trained by exposing them to measured input–output data (also called patterns). Training, also called learning, is to adjust the free parameters, namely the weights associated with the various nodal connections so that the error between the measured output and network output to the same inputs will be minimized, and thus amounts a classical optimization problem. The knowledge acquired through learning is stored in the weights thus determined. The ability of ANNs to generalize the system behavior from limited data makes them an ideal tool for characterizing complex nonlinear processes.^{15–17}

Several types of neural networks result from different ways of interconnecting the neurons and the way the nodes function in hidden layers. The three networks most commonly used in system identification are: 1) feedforward neural network (FFNN), 2) recurrent neural network (RNN), and 3) radial basis function (RBF) neural network.

As the name implies, FFNNs process the information in only one direction through the various layers. They consist of input, output, and usually one or two hidden layers. They are the simplest of the NNs. On the other hand, RNNs are characterized by a bi-directional flow, that is, having one or more neurons that feeds data back into the network and thereby modifies the inputs to the neurons. Amongst this class, the RNN developed by Hopfield¹⁸ is more commonly used in practice.^{19–23} Owing to the feedback structure, they are amenable to state space model representations. A RBF is a multivariable functional interpolation which depends upon the distance of input signal vector from the center, which is determined with reference to the input distribution.^{24,25} The basic structure of RBFs, which contain only one hidden layer and a linear output layer, is similar to that of a single-layer perceptron, but with a difference that the hidden layer inputs are not weighted, instead the hidden nodes perform the radial basis functions. In general, RBFs are powerful, efficient, and less prone to local minima, whereas the performance tends to be dominated by the spread of the input–output subspace.

Yet another type of network that finds application in modeling is the so-called local model network (LMN) developed by Johansen and Foss.²⁶ The basic idea

behind the approach is to partition the multidimensional input space into several subspaces, to which we fit linear models. They are then pieced together in a form of a simple network. A weighted sum of these local models gives the overall network output.^{26–29} Use of a Gaussian weighting function ensures continuity over the partitions. Two issues involved in the use of LMN are: 1) the structure comprising a number of input space partitions, their sizes and location, and 2) estimation of parameters in each of these models. An adaptive partitioning procedure allows efficient breakdown of the input space into the least number of parts, having as large a range of each variable as possible without sacrificing the accuracy of linear models. The estimation of parameters can be carried out by applying any standard algorithm such as equation error or output error method. The approach has some advantages over the classical FFNN, namely that physical meaning can be assigned to estimated parameters and that a priori information can be made use of. Applicability of the approach to model hinge moment coefficients from flight data²⁸ and also to model nonlinearities in the lift coefficients due to stall hysteresis has been demonstrated,²⁹ but otherwise LMNs have not yet found widespread use in aerodynamic modeling from flight data.

In the case of flight vehicles, NNs are used for a variety of different applications, such as 1) structural damage detection and identification,³⁰ 2) fault diagnostics (detection and isolation) leading to compensation of control surface failures,^{31–33} 3) modeling of aerodynamic characteristics from flight data,^{34–39} 4) generalized reference models for six degrees-of freedom motion simulation using global aerodynamic model including unsteady aerodynamics and dynamic stall,^{40–42} and for detection of unanticipated effects such as icing,⁴³ and 5) autopilot controllers and advanced control laws for applications such as carefree maneuvering.^{44,45} Other applications include, for example, calibration of flush air data sensors,⁴⁶ multisensor data fusion, mission planning, and generation of mission-plan in real time,⁴⁷ and modeling of unstable aircraft or aero-servoelastic effects. Although FFNNs have been used for the above and other purposes, engineers well versed with the so-called conventional approach always had in the past some hesitations in applying these techniques. The reservations on the overall suitability and efficacy of the NN approach stem partly from the background and outlook of the analysts, partly from the empirical character and black-box model structure without explicit formulation of dynamic system equations and of aerodynamic phenomenon, and partly from the uncertainties associated with the extrapolation capabilities of models trained on a limited set of data. The last issue, which is still an open issue, is critical for databases used in certification and for flight safety aspects to guarantee sufficient reliability. Furthermore, incremental updates to existing NN or tuning of submodels is difficult. Nevertheless, in several applications fairly good results can be obtained easily through simple NNs, which may serve the intended purpose of mapping the overall cause–effect relationship.

It is beyond the scope of this book to provide a detailed account of various types of NN and their capabilities. We restrict ourselves only to the FFNNs, with a goal of acquiring a basic understanding of these techniques for the specific purpose of aerodynamic modeling. The treatment provided here should serve as a starting point for this rapidly evolving approach, which is complementary to the

conventional techniques. In this chapter we start with basics of neural processing, and concentrate on forward propagation and network training using back-propagation. Two modifications to improve the convergence are presented. This is followed by summarizing a study that suggests possible choices for the network tuning parameters such as input–output data scaling, number of nodes, learning rate, and slope of sigmoidal activation function. A brief description of a simple software implementation is provided. Finally, we address three examples of increasing complexity to demonstrate FFNN modeling capabilities. The first example considers a fairly simple case of modeling lateral–directional motion; the second example deals with modeling lift, drag, and pitching moment coefficients from data with atmospheric turbulence. The same two examples considered in the previous chapters provide a comparative background. The third example pertains to a significantly more complex nonlinear phenomenon of aircraft undergoing quasi-steady stall.

II. Basics of Neural Network Processing

The basic building block of neural network comprises a computational element called neuron, shown in Fig. 8.1. Each neuron receives inputs from multiple sources, denoted u , an $(n_u \times 1)$ vector. Usually an additive bias representing nonmeasurable input is appended to the external inputs, which is denoted b and has some constant value. These inputs are multiplied by effectiveness factors called weights (W_i , $i = 1, \dots, n_u$), and added up to yield the weighted sum which passes through an activation function f to provide an output. Thus, a neuron represents a multi-input single-output subsystem. Obviously, the input–output characteristic of a neuron depends on the weights W and the activation function f . Different types of activation function have been used in NN applications, such as linear, linear with a limiter, switching type, hyperbolic tangent, and logistic function.² For the reasons which we will discuss subsequently, the most common choice is the one shown in Fig. 8.1.

A feedforward neural network comprises a large number of such neurons which are interconnected and arranged in multiple layers. Each node in a layer is connected to each node in the next layer. Figure 8.2 shows a schematic of

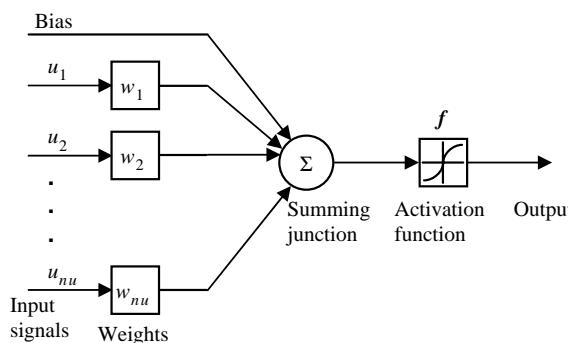
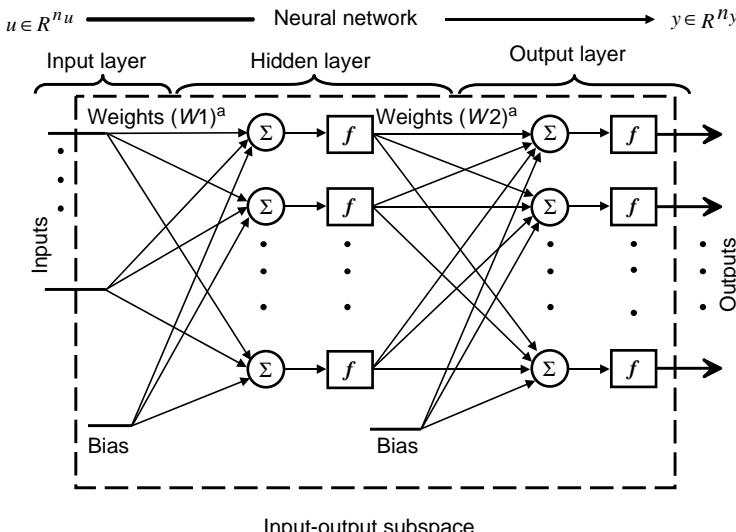


Fig. 8.1 Schematic of a single neuron.



**Fig. 8.2 Schematic of a feedforward neural network with one hidden layer.
aUnknown parameters.**

FFNN with one hidden layer. The input space to be mapped consists of n_u inputs and n_y outputs. Accordingly, the number of nodes in the input and output layers is fixed automatically. The internal behavior of the network is characterized by the inaccessible (hidden) layer. To be able to cater to any general nonlinear problem, it is mandatory that the activation function of the hidden layer must be nonlinear, whereas that in the output layer may be linear or nonlinear. If all neurons in an NN, including those in the hidden layer, had linear activation function, the scope of modeling is restricted to linearly separable subspaces. In the case of NN with more than one hidden layer, the basic procedure remains the same as that depicted in Fig. 8.2, wherein the output of the first hidden layer feeds the second hidden layer, the output of which then passes through the next hidden layer or through the output layer.

The neural network performance, that is, its ability to accurately duplicate the data used in training with adequate prediction capability, depends on the size of the networks, on the number of hidden layers and on the number of neurons in each hidden layer. The larger the size is, the larger the computational effort. NN networks having more than two hidden layers are extremely rare in practice. In general, it has been shown that an FFNN with a single hidden layer and any continuous sigmoidal activation function can arbitrarily well approximate any given input–output subspace.^{48,49} The so-called sigmoidal function $f(u)$ is smooth around zero, and has values between [0, 1] for inputs in the range $[-\infty, \infty]$. It is also easily differentiable, a property that is very useful in the training algorithm which we will address in the next section. The practical investigations reported in the past have also shown that a single hidden layer FFNN

with adequate number of neurons is quite sufficient for aerodynamic modeling from flight data.³⁷ Accordingly, we consider in this book FFNNs with only one hidden layer.

The choice of number of neurons in the hidden layer is little trickier, because the optimum number of neurons required in a specific case depends on several factors, such as number of input and outputs, the amount of noise in the data to be matched, and the complexity of the input–output subspace. Surprisingly, it also depends on the training algorithm. In the NN literature several rules of thumb have been suggested for n_h , the number of neurons in the hidden layer, such as: 1) somewhere between n_u and n_y , 2) two-thirds of $(n_u + n_y)$, or 3) less than $2n_u$. The efficacy of these rules is doubtful, because they do not account for the aforementioned issues of noise and complexity.⁷ Too few neurons may lead to under-fitting, whereas too many may result in over-fitting the complex subspaces, in both cases affecting the generalization, that is, overall performance and predictive capability. The best approach to determine the optimum number of n_h , which we have mostly followed in the past, appears to be the one based on numerical investigations of trying out many networks with different number of hidden neurons, and choosing the one yielding minimum estimated error.

Application of FFNN to a given data consists of two parts, namely training and prediction:

- 1) During the first part, called training, the network is exposed to given data set. The free parameters of the network, that is, the weights, are determined by applying some suitable algorithm. The weights are continuously adjusted until some convergence criterion is met. This learning phase, which essentially leads to characterizing the input–output characteristics, corresponds to the modeling part of the system identification.
- 2) During the second part, called prediction, the same set of data or possibly other data not used in training is passed through the above network, keeping the weights fixed, and checking the prediction capability. In other words, this step corresponds to the model validation.

III. Training Algorithms

Having elaborated on the basics of the NN, we now turn our attention to procedures of determining the weights. Back-propagation (BP) is the most commonly used method for this purpose, the essential idea behind this approach being to view the error as a function of network weights or parameters and to perform gradient descent in the parameter space to search for a minimum error between the desired and estimated values. Strictly speaking, back-propagation in itself is not an optimization procedure; it provides a means to compute the gradients, but we start with the output layer and apply the chain rule to obtain gradients at the previous (hidden) layers. In other words, we propagate the error backwards, hence the name of the algorithm. The actual minimization is based on the steepest descent method. The training algorithm comprises two steps: 1) forward pass, which basically corresponds to simulation, that is, keeping the weights fixed the input signal is propagated through the network; this is necessary

to compute the error between measured outputs and network computed responses; 2) backward pass, which propagates the error backwards through various layers, computes the gradients, and updates the weights.

There are two common types of BP learning algorithms: 1) batch or sweep-through, and 2) sequential or recursive. The batch BP updates the network weights after presentation of the complete training data set. Hence a training iteration incorporates one sweep-through of all the training patterns. In the case of the recursive BP method, also called pattern learning, the network weights are updated sequentially as the training data set is presented point by point. We address here only the recursive BP, which is more convenient and efficient compared with the batch BP.^{2,13} A single recursive pass through the given data set is not sufficient to minimize the error; hence the process of recursively processing the data is repeated a number of times, leading to a recursive-iterative process.

A. Forward Propagation

As already pointed out, forward pass refers to the computational procedure of applying the measured input data to the network and computing the outputs. To begin with, it is necessary to specify starting values for the weights. They are usually initialized randomly. Let us denote the weights of a FFNN with a single hidden layer, shown in Fig. 8.2, as follows:^{37,50}

- W_1 weight matrix between input and hidden layer ($n_h \times n_u$)
- W_{1b} bias weight vector between input and hidden layer ($n_h \times 1$)
- W_2 weight matrix between hidden and output layer ($n_y \times n_h$)
- W_{2b} bias weight vector between hidden and output layer ($n_y \times 1$)

where n_u is number of nodes in the input layer, n_h is the number of nodes in the hidden layer and n_y is the number of nodes in the output layer.

Denoting the given input vector as u_0 , propagation from input layer to a hidden layer yields:

$$y_1 = W_1 u_0 + W_{1b} \quad (8.1)$$

$$u_1 = f(y_1) \quad (8.2)$$

where y_1 is the vector of intermediate variables (output of the summing junctions), u_1 is the vector of node outputs at the hidden layer and f is the vector of nonlinear sigmoidal node activation functions defining the node characteristics. The hyperbolic tangent function is chosen here:

$$f_i(y_1) = \tanh\left(\frac{\gamma_1}{2} y_1(i)\right) = \frac{1 - e^{-\gamma_1 y_1(i)}}{1 + e^{-\gamma_1 y_1(i)}}, \quad i = 1, 2, \dots, n_h \quad (8.3)$$

where γ_1 is the slope (gain) factor of the hidden layer activation function. All the nodes at any particular layer have, in general, the same slope factor.

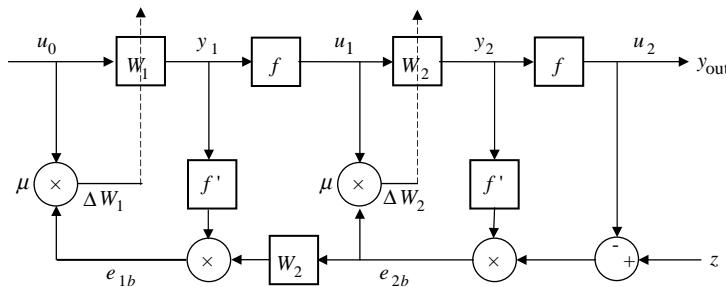


Fig. 8.3 Schematic of forward pass and back-propagation computations.

The node outputs u_1 at the hidden layer form the inputs to the next layer. Propagation of u_1 from the hidden layer to the output layer yields

$$y_2 = W_2 u_1 + b \quad (8.4)$$

$$u_2 = f(y_2) \quad (8.5)$$

where y_2 is the vector of intermediate variables (output of the summing junctions), u_2 is the vector of node outputs at the output layer and f is the vector of sigmoidal node activation functions. Once again, the hyperbolic tangent function is chosen for the node activation function:

$$f_i(y_2) = \tanh\left(\frac{\gamma_2}{2} y_2(i)\right) = \frac{1 - e^{-\gamma_2 y_2(i)}}{1 + e^{-\gamma_2 y_2(i)}}, \quad i = 1, 2, \dots, n_y \quad (8.6)$$

where γ_2 is the slope (gain) factor of the output layer activation function.

Thus, propagation of the inputs u_0 through the input, hidden, and output layers using known (fixed) weights yields the network estimated outputs u_2 . The above computational steps of the forward pass and those of the back propagation discussed next are shown schematically in Fig. 8.3, where the bias is not shown explicitly. The goal of optimization boils down to finding optimal values for the weights such that the outputs u_2 match with the measured system responses z .

B. Standard Back-propagation Algorithm

The back-propagation learning algorithm is based on optimizing a suitably defined cost function. At each point, the local output error cost function, which is the sum of the squared errors, is given by

$$E(k) = \frac{1}{2} [z(k) - u_2(k)]^T [z(k) - u_2(k)] = \frac{1}{2} e^T(k) e(k) \quad (8.7)$$

where k is the discrete data index, z the measured response vector, u_2 the network estimated response vector, and $e = (z - u_2)$ the error.

Minimization of Eq. (8.7) applying the steepest descent method yields

$$W_2(k+1) = W_2(k) + \mu \left(-\frac{\partial E(k)}{\partial W_2} \right) \quad (8.8)$$

where μ is the learning rate parameter and $\partial E(k)/\partial W_2$ is the local gradient of the error cost function with respect to W_2 . The learning rate is equivalent to a step size, and determines the speed of convergence. A judicious choice of μ , greater than zero, is necessary to ensure reasonable convergence rate.

Substitution of Eqs. (8.4) and (8.5) in Eq. (8.7), and the partial differentiation with respect to the elements of matrix W_2 yields the gradient of the cost function:

$$\frac{\partial E(k)}{\partial W_2} = -f'[y_2(k)][z(k) - u_2(k)]u_1^T(k) \quad (8.9)$$

where $f'[y_2(k)]$ is the derivative of the output node activation function, that is, of Eq. (8.6). Now, substituting Eq. (8.9) in Eq. (8.8) and by defining e_{2b} as

$$e_{2b}(k) = f'[y_2(k)][z(k) - u_2(k)] \quad (8.10)$$

the weight-update rule for the output layer is obtained as

$$W_2(k+1) = W_2(k) + \mu e_{2b}(k)u_1^T(k) \quad (8.11)$$

Similarly, substituting Eqs. (8.1) and (8.2) in Eq. (8.7), and partial differentiation with respect to W_1 yields

$$\frac{\partial E(k)}{\partial W_1} = -f'[y_1(k)]W_2^T e_{2b}(k)u_0^T(k) \quad (8.12)$$

where $f'[y_1(k)]$ is the derivative of the hidden layer activation function, that is, of Eq. (8.3). Now, once again following a similar procedure and by defining e_{1b} as

$$e_{1b}(k) = f'[y_1(k)]W_2^T e_{2b}(k) \quad (8.13)$$

the weight-update rule for the hidden layer is obtained as

$$W_1(k+1) = W_1(k) + \mu e_{1b}(k)u_0^T(k) \quad (8.14)$$

For the hyperbolic tangent function, chosen as the hidden and output layer node activation functions, respectively, in Eqs. (8.3) and (8.6), the derivatives $f'[y_1(k)]$ and $f'[y_2(k)]$ are given by

$$f'(y_i) = \frac{\gamma_l}{2} \left[1 - \tanh^2 \left(\frac{\gamma_l y_i}{2} \right) \right] = \frac{2\gamma_l e^{-\gamma_l y_i}}{(1 + e^{-\gamma_l y_i})^2} \quad (8.15)$$

where l ($=1$ or 2) is the index for the input-to-hidden and hidden-to-output layers.

To summarize, the complete algorithm comprises 1) computation of the hidden and output node activation functions and outputs, Eqs. (8.1), (8.2) and (8.4), (8.5), 2) computation of the derivatives of the node activation functions, Eq. (8.15) for hidden and output layers, 3) computation of the error functions, Eqs. (8.10) and (8.13), and 4) computation of the new weights from Eqs. (8.11) and (8.14). These steps are recursively repeated for $k = 1, \dots, N$, where N is number of data points. At the end of the recursive loop, the mean square error (MSE) is computed:

$$\sigma^2 = \frac{1}{Nn_y} \sum_{k=1}^N \sum_{j=1}^{n_y} [z_j(k) - u_j(k)]^2 \quad (8.16)$$

Several such training iterations are carried out until the MSE satisfies some convergence criterion, for example the relative change from iteration to iteration less than a specified value or until the maximum number of iterations is reached. Usually a large number of iterations, ranging from a few hundreds to thousands is required to achieve good training and predictive performance.

The convergence rate of the steepest descent optimization is slow, affected by poor initial weights, and prone to finding a local minimum. It turns out that the standard BP algorithm discussed in this section may not yield training error to a level near the globally optimum value.^{7,51} Increasing the number of neurons in the hidden layer usually helps to find a good local optimum. In such a case it is necessary to ascertain the predictive capability of the oversized network. Other approaches are to modify the optimization method. There are several possible techniques, including the more advanced Levenberg–Marquardt algorithm. We restrict ourselves here to the two modifications which are commonly used in the NN applications.

C. Back-propagation Algorithm with Momentum Term

The performance of the standard BP algorithm described in the foregoing section is determined by the step size (learning rate) μ . A very small value implies very small steps, leading to extremely slow convergence. On the other hand, larger values result in faster learning, but can lead to rapid changes the direction of descent from discrete point to point resulting in parasitic oscillations, particularly when the cost function has sharp peaks and valleys. To overcome these difficulties, the update Eqs. (8.11) and (8.14) are modified as follows:

$$W_2(k+1) = W_2(k) + \mu e_{2b}(k)u_1^T(k) + \Omega[W_2(k) - W_2(k-1)] \quad (8.17)$$

$$W_1(k+1) = W_1(k) + \mu e_{1b}(k)u_0^T(k) + \Omega[W_1(k) - W_1(k-1)] \quad (8.18)$$

where Ω is called the momentum factor. As implied by the difference $[W_l(k) - W_l(k-1)]$, the additional term in the update equations makes the

optimization move along the temporal average direction used in the past iteration. Over-relaxation of the gradient update, that is, of Eqs. (8.11) and (8.14), through the last terms on the right-hand side of Eqs. (8.17) and (8.18), respectively called the momentum term, helps to damp out parasitic oscillations in the weight update. As Ω is chosen greater than zero and less than 1, it approximately amounts to increasing the learning rate from μ to $\mu/(1 - \Omega)$ without magnifying the parasitic oscillations.² For this reason, the last term in the above equations is also called as acceleration term.

D. Modified Back-propagation Algorithm

Yet another modified BP algorithm results from minimization of the mean squared error with respect to the summing junction outputs (i.e., inputs to the nonlinearities). This is in contrast to the standard BP procedure that minimizes the mean squared error with respect to the weights. We draw here heavily on the development presented in Ref. 52 to summarize the principle behind the algorithm and give only the computational steps necessary for implementation. Briefly, the error signals generated through back-propagation are used to estimate the inputs to the nonlinearities. These estimated inputs to the nonlinearities and the inputs to the respective nodes are used to produce updated weights through a set of linear equations, which are solved using a Kalman filter at each layer. The overall training procedure is similar to that of the standard BP algorithm discussed in Sec. III.B, except for the weight update formulas, which require computation of the Kalman gains at each layer. Hence, we provide here only those steps that differ from the ones given in Sec. III.B.

The update equations for the output layer and hidden layer are given by

$$W_2(k+1) = W_2(k) + [d(k) - y_2(k)]K_2^T(k) \quad (8.19)$$

$$W_1(k+1) = W_1(k) + \mu e_{1b}(k)K_1^T(k) \quad (8.20)$$

where K_2 and K_1 are the Kalman gain vectors of the size $(n_h + 1, 1)$ and $(n_u + 1, 1)$ associated with the hidden layer and output layer respectively, and d is desired summation output given by

$$d(k) = \frac{1}{\gamma} \ln \left[\frac{1 + z(k)}{1 - z(k)} \right] \quad (8.21)$$

The Kalman gains for the two layers are given by

$$K_1(k) = \frac{D_1(k)u_0(k)}{\lambda_1 + u_0^T(k)D_1(k)u_0(k)} \quad (8.22)$$

$$K_2(k) = \frac{D_2(k)u_1(k)}{\lambda_2 + u_1^T(k)D_2(k)u_1(k)} \quad (8.23)$$

and the matrices D , representing the inverse of the correlation matrices of the training data, are given by

$$D_1(k+1) = \frac{D_1(k) - K_1(k)u_0^T(k)D_1(k)}{\lambda_1} \quad (8.24)$$

$$D_2(k+1) = \frac{D_2(k) - K_2(k)u_1^T(k)D_2(k)}{\lambda_2} \quad (8.25)$$

where λ_1 and λ_2 denote the forgetting factors. These factors allow the new information to dominate. If any one of the measured reposes $z(k)$ happens to be exactly unity, then the term $[1 - z(k)]$ in the denominator of Eq. (8.21) leads to numerical difficulties. This can be avoided by detecting such cases and simply adding a very small fraction. The other option is to use a linear activation function in the output layer.⁵⁰

The experimental investigations pertaining to neural network aerodynamic modeling from flight data confirm the improved convergence rate compared with standard BP or the momentum term. Although the computational overhead at each node is higher than that for the other two algorithms, the modified algorithm requires a smaller total overhead. It is also less sensitive to the choice of the network parameters such as slope of the activation function or initial weights, and yields training error to a level near the globally optimum value.

Interestingly, in the context of a different estimation technique we have already come across this property of improved convergence though incorporation of Kalman gain. We recall from Chapter 5, Sec. XI.A that, compared with the standard output error method, a more advanced filter error method incorporating Kalman or extended Kalman filter provided significantly improved convergence. In general, such approaches are less sensitive to stochastic disturbances and lead to a more nearly linear optimization problem, which has fewer local minima and a faster convergence rate.

IV. Optimal Tuning Parameters

The standard BP or BP with a momentum term is characterized by a slow convergence and high sensitivity to parameters like learning rate and initial weights. There are several interrelated parameters that influence the FFNN performance: 1) input and output data scaling, 2) initial network weights, 3) number of hidden nodes, 4) learning rate, 5) momentum parameter, and 6) slope (gain) factors of the sigmoidal function. Although some heuristics rules of thumb are found in the literature, the choice of these parameters is usually based on the experimental investigations. Based on a study carried out on a typical flight data pertaining to lateral directional motion, an FFNN with six inputs (sideslip angle β , roll rate p , yaw rate r , aileron deflection δ_a , spoiler deflection δ_{SP} , and rudder deflection δ_r) and three outputs (side force coefficient C_Y , rolling moment coefficient C_ℓ , and yawing moment coefficient C_n), the following general guidelines are suggested:³⁷

Number of hidden layers	1	(1)
Number of hidden nodes	6	(5–8)
Data scaling range	−0.5–0.5	(−0.5–0.5)
Nonlinear function slopes	0.85/0.6	(0.6–1.0)
Learning rate parameter	0.125	(0.1–0.3)
Momentum parameter	0.5	(0.3–0.5)
Initial random weights	0.3	(0.0–1.0)

Typical values are provided in the second column. In general, it is adequate to choose these parameters within a certain band, listed in parentheses.

For the reasons elaborated in Sec. II, and supported by the above investigations based on networks with one and two hidden layers, FFNN with a single hidden layer is considered adequate for aerodynamic modeling from flight data. In most of the cases fewer than eight neurons in the hidden layer were sufficient for good predictive capability. In some cases, where the input–output relationship is highly nonlinear, more nodes may be necessary.

In general, it is known that minimization of an error function is affected by numerical values of the input–output data, particularly if the values differ much in magnitude. In these cases data scaling is recommended while applying the optimization methods. This is found to be particularly the case when training FFNN. Here, scaling refers to arranging the values between the chosen lower and upper limits such that all the variables have similar orders of magnitude. It is our experience that data scaling significantly improves the convergence of FFNN training to a reasonable number of iterations. The scaling range of −0.5–0.5 is found to be optimal for both input and output variables.

Since the weights appearing in an FFNN have no physical significance, we cannot make use of any a priori information that may be available from classical approaches. The initial weights are normally set to small random numbers to avoid saturation in the neurons. The algorithm does not work properly if the initial weights are either zero or poorly chosen nonzero values. Random weights in the range 0–1 led to the best FFNN convergence, while the random weights of magnitude more than 1 led to poor convergence in some cases. Larger weights on the hidden-to-output layer than those for the input-to-hidden usually help to improve the overall training process. However, no significant differences were observed when the same factor, as suggested in the above list, was used.

Larger learning rate μ leads to a faster convergence, but the resulting NN may not have sufficient prediction capabilities. In addition, inappropriate selection of other influencing parameters in combination with a large value of learning rate may lead to oscillations. By analogy to the human brain, it can be said that the network is unable to retain the complete information if it is fed in at a very fast rate. From this aspect, we prefer smaller values for μ , which calls for a large number of training iterations. Learning rates in the range 0.1–0.3 are optimal.

The effect of the momentum term, which is introduced to improve the convergence rate and damp out the larger oscillations in the network weights, is not obvious. If the other parameters like learning rate and initial weights are

chosen properly, the influence of this parameter on the network convergence is negligible. If at all, then Ω in the range from 0.3 to 0.5 is suggested. As a caveat, in some cases, a momentum parameter above 0.8 value leads to complete divergence.

With regard to the slopes of the nonlinear activation functions of the hidden layer nodes (γ_1) and of the output layer nodes (γ_2), it was observed that very small slope factors close to zero have an adverse effect. It was also found that the performance was not satisfactory for the slope factor values of more than 1. In general, the slope factor γ_1 of the hidden layer will have more influence on network convergence than the slope factor γ_2 of output layer nonlinear function. This observation conforms to the basic FFNN property, namely the hidden layer provides primarily the characterization of the internal behavior of the network and global functional approximation. The type of activation function, linear or nonlinear, for the output layer plays only a secondary role. In our study we have retained nonlinear activation function, mainly because it leads to a more general representation. It has also been observed that different slope factors for hidden and output layer functions help to improve the convergence. Accordingly, we suggest using slopes of 0.85 and 0.6 for the hidden and output layers, respectively.

Finally, we once again recall from Sec. III.D that the modified BP algorithm using Kalman gains tends to be more robust to the choice of initial weights and learning rates. The momentum term is not involved, but instead we need to specify the forgetting factors λ_1 and λ_2 . They should be very close to 1. In typical cases, which we will address subsequently in this chapter, a forgetting factor of 0.999 is used. Smaller values provided quite erratic results. The choice of number of neurons and the effect of data scaling remains the same as discussed above. The learning rate μ may differ in the present case from that suggested earlier for the other algorithms.

V. Extraction of Stability and Control Derivatives from Trained FFNN

The inability to provide physical interpretations to the weights of a trained FFNN is one of the limitations of neural processing. In a limited sense, one can attempt to overcome this limitation by extracting the linearized derivatives from a trained network. This becomes possible by using the basic concepts of numerical perturbations and the definitions of the stability and control derivatives appearing in the aerodynamic force and moment coefficients. These derivatives can be thought of as the change in the aerodynamic force and moment coefficients due to a small variation in either the motion or control variable while the rest of the variables are held constant.

The starting point to apply this approach, called the delta method,^{39,53} is the availability of a trained FFNN with appropriate inputs and outputs. As considered in the previous section, such a network for the lateral-directional motion may consist of six inputs ($\beta, p, r, \delta_a, \delta_{RSP}, \delta_r$) and three outputs (C_Y, C_ℓ, C_n). Now, let us consider an extraction of dihedral effect, the derivative $C_{\ell\beta}$. For this purpose, we perform FFNN predictions, which consist only of the forward pass keeping the weights fixed, with two sets of perturbed β input variable, keeping the other inputs as used in the training unchanged. Let the perturbation be denoted by

constant value $\Delta\beta$. The same perturbation is used for all the N data points. Denoting the FFNN outputs corresponding to perturbed inputs $\beta + \Delta\beta$ and $\beta - \Delta\beta$ as C_ℓ^+ and C_ℓ^- , respectively, the derivative $C_{\ell\beta}$ can be approximated as

$$C_{\ell\beta} = \frac{C_\ell^+ - C_\ell^-}{2\Delta\beta} \quad (8.26)$$

For the reasons we have already elaborated in Chapter 3, central differencing is preferred over the one-sided forward or backward difference approximation.

Application of Eq. (8.26) to complete data yields N values, or in other words time histories, for the derivative to be extracted. These extracted values are plotted as histograms, which usually show a near-normal distribution, from which the mean representing the aerodynamic derivative can be determined. Equivalently, a simple averaging process can also be used. Following the above procedure any other derivative can be extracted from the trained FFNN.

Although the delta method may be a viable approach, let us take a look at the practical implications. First, if we are interested in extracting linear derivatives from a given set of input–output data, the classical approach of least squares or total least squares covered in Chapter 6 is more suitable and simpler, as well as quite sufficient. The regression approach is more direct and would yield more reliable estimates than those through the roundabout route of using FFNN. Linear regression is, in fact, equivalent to a simplest form of FFNN without any hidden layer, and having linear activation function. The least squares estimates are also obtained without any a priori knowledge about the starting values. Secondly, we are well aware that the validity and range of applicability of linear derivatives is restricted; in some cases the linearized derivative may not even make any sense. The main advantage and strength of FFNN modeling is in its ability to capture highly nonlinear complex phenomena in a global sense, for which we may not be in a position to postulate models with physical understanding. In such cases only will the advantages of FFNN be fully exploited.

VI. FFNN Software

There are many commercial as well as free software packages available, covering different types of NNs and training algorithms.⁷ Instead of using one of those, we develop simple software here based on the two back-propagation training algorithms discussed earlier in this chapter. This simple version provided herewith is quite sufficient to trace the procedures elaborated in Sec. II and to generate the results that we will discuss here. It also helps to provide a somewhat uniform flight data interface. The source codes (Matlab® m-files) for feedforward neural network with back propagation are provided in the directory /FVSysID/chapter08/.

The main program “mainFFNN” is the starting point. It provides an option to invoke a test case to be analyzed, and an interface to model definition function specifying the input–output data space to be modeled. Specification or assignment of the following information is necessary:

test_case	integer flag for the test case
Zout(Ndata,Ny)	flight data for measured outputs (N, n_y)

Xin(Ndata,Nu) flight data for measured control inputs (N, n_u)
 dt sampling time

Thus, Xin and Zout define the subspace to be matched. The following related information is automatically generated from the sizes of the above arrays.

Nu	number of input variables (n_u)
Ny	number of output variables (n_y)
Ndata	number of data points used in the training phase (N)
NdataPred	number of data points to be used for prediction

The above list shows on the left side the variable names used in the program “mainFFNN” and in other functions called therein, followed by the description and the notation used to denote these variables in the text. If prediction is performed on the same data set used in the training, then NdataPred is the same as Ndata. If validation is to be performed on a completely different data set, it will be necessary to run only the forward pass, for which minor modifications will be necessary, which are left to the reader. The sampling time (dt) is not used in the algorithm, but for generating time history plots. Although there is no restriction on the units of the various input and output variables, we usually load the flight data in Zout(Ndata,Ny) and Xin(Ndata,Nu) having consistent units.

The neural network related parameters ($\gamma_1, \gamma_2, \mu, \Omega$) are set to the recommended default values in the main program “mainFFNN.m.”

γ_1	slope of activation function from input to hidden layer
γ_2	slope of activation function from hidden to output layer
μ	learning rate parameter
Ω	momentum parameter
iScale	integer flag to scale the data
SCmin	lower limit for data scaling
SCmax	upper limit for data scaling

These tuning parameters may have to be adapted in a specific case to obtain optimal convergence during training and to result in a good predictive capability.

The network size (number of nodes in the hidden layer), the maximum number of iterations of the recursive–iterative training, and the training algorithm are specified interactively by keying in the following information:

NnHL1	number of nodes in the hidden layer
itMax	maximum number of iteration
trainALG	training algorithm: = 0: back-propagation with momentum term = 1: modified back-propagation using Kalman gains

At the end of the neural network training and prediction cycles, depending upon the test_case, plots are generated of the time histories of measured and estimated responses, showing the predictive capability of the trained network.

VII. Examples

To demonstrate the ability of FFNNs to model any input–output subspace we consider here three examples of different complexities. The first one pertains to a fairly simple and classical example of modeling lateral-directional motion. In the second example, we consider the case of estimating lift, drag, and pitching moment coefficients from flight data with atmospheric turbulence. The last example addresses a highly nonlinear aerodynamic phenomenon of quasi-steady stall.

A. Modeling of Lateral-directional Aerodynamics

As a first test case, we once again turn our attention to the example considered in Chapter 6, Sec. IX.B and Chapter 7, Sec. V.A, in which the LS and RLS methods were applied to flight data from three concatenated maneuvers performed at 16000 ft and at 200 kts nominal speed with the test aircraft ATTAS. Recall that the first maneuver is a multistep elevator input exciting the short period motion, the second is an aileron input resulting in the bank-to-bank motion and the third is a rudder doublet input exciting the Dutch roll motion. The three maneuvers are 25, 30, and 30 s long, respectively. The control inputs applied during these three flight maneuvers are found in Fig. 6.7 and a comparison of the flight derived (i.e., measured) force and moment coefficients with those estimated by the least squares method using the model postulated in Eq. (6.76) is shown in Fig. 6.8. To this data we apply here the neural network technique to model the lateral-directional aerodynamics.

Now, we run this case by calling the program “/FVSysID/chapter08/mainFFNN,” which includes the same data pre-processing step as described in Chapter 6, Sec. IX.B to compute the aerodynamic force and moment coefficients. This step of deriving aerodynamic force and moments coefficients C_D , C_L , C_m , C_Y , C_ℓ , C_n from measured accelerations and angular rates remains unchanged. The details are not repeated here, but they can be easily traced in the data definition function “mDefCase23.m,” which calls the data preprocessing function “umr_reg_attas.” We designate this case as $\text{test_case} = 23$.

Having derived the aerodynamic force and moment coefficients from relevant measured data, we now define the input–output subspace to be modeled using FFNN as follows:

Number of dependent variables:	3
Dependent variables:	C_Y , C_ℓ , C_n
Number of independent variables:	5
Independent variables:	β , p , r , δ_a , δ_r

The corresponding data is stored in the arrays $\text{Zout}(N, n_y)$ and $\text{Xin}(N, n_u)$, respectively. Accordingly, the number of nodes in the input layer are $n_u = 5$ and those in the output layer are $n_y = 3$. The length of the data (Ndata) for training is 2128, and the same length is used for prediction purpose. The recommended default values ($\gamma_1 = 0.85$, $\gamma_2 = 0.6$, $\mu = 0.125$, $\Omega = 0.5$) are used for the neural network parameters. The number of nodes in the hidden layer $NnHL1$ (= 8)

and the maximum number of iterations $itMax$ ($= 2000$) are specified interactively by keying in the values. Both training algorithms were investigated.

The choice of $NnHL1=8$ and $itMax = 2000$ provided typical results for predictive capability shown in Fig. 8.4. The two plots on the bottom show the aileron and rudder deflections (δ_a , δ_r), and the three plots on the top the match for the side force, rolling moment and yawing moment coefficients. Since initial weights are initiated randomly, it will not be possible to reproduce exactly the same results, but several repeated trials yielded similar model quality for the match between the measured data and predicted responses. The residual error after the training phase is not the same as that obtained from the prediction step on the same data used in the training. This is because the weights are altered recursively for each point during training, whereas during the prediction step the estimated weights are kept fixed. It is for this reason that the prediction step needs to be performed to verify the performance over the complete data of N points.

In this particular training run, starting from the cost function value of 3×10^{-7} , the back-propagation algorithm reduced the error significantly to

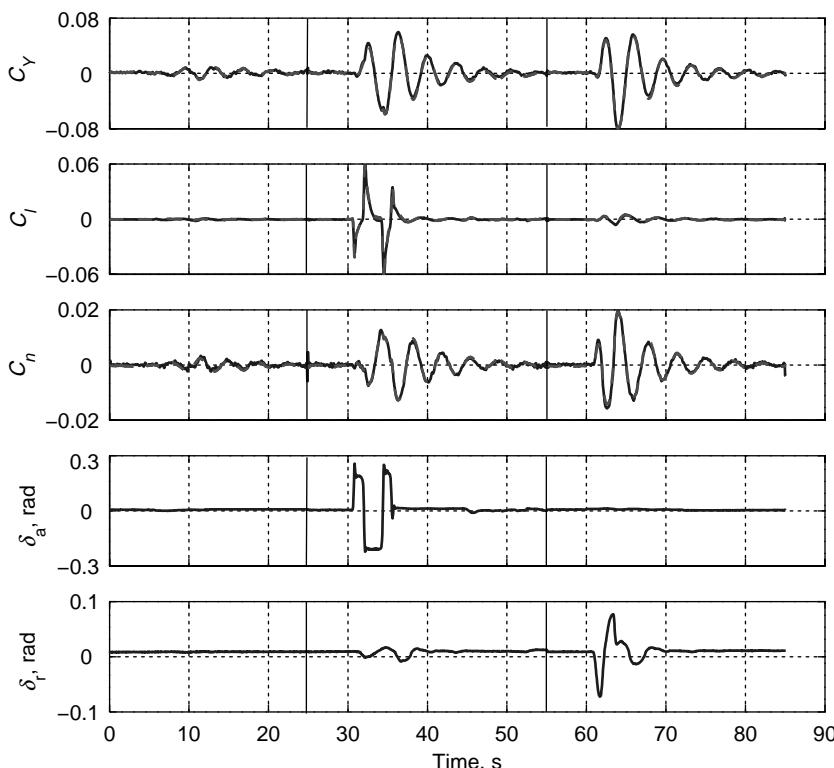


Fig. 8.4 Predictive capability of network trained on lateral-directional data (—, flight measured; - - -, FFNN predicted).

roughly 5×10^{-13} in less than 50 iterations, reaching a cost function value of 4.7×10^{-13} after 500 iterations, and 3.5×10^{-13} after 2000 iterations. The application of the modified back-propagation algorithm using Kalman gain showed very fast decrease in the first few iterations, but the stringent tolerances for terminations were not reached even after a few hundred iterations. This is attributed to the fact that data is corrupted with noise, and after capturing the main system characteristics, the network tries to account for random noise, leading to small oscillations in the cost function. To avoid such a phenomenon, other termination criterions should be introduced. We also make a note of the fact that the initial cost function during the training phase at the starting iteration, even when using the same starting weights, is different for the two methods, because the back-propagation which adjusts the weights recursively over N data points is done differently. The cost function minimum from the modified algorithm was observed to be slightly lower than that from the standard back-propagation with momentum term. This supports the observation made earlier that the network size and ability to capture the input–output relation also depends on the training algorithm.

B. Modeling of Longitudinal Data with Process Noise

The second example pertains to modeling of lift, drag, and pitching moment coefficients of the research aircraft HFB-320 considered in Chapter 5, Sec. XI.B and in Chapter 7, Sec. V.D. It is now attempted to model these force and moment coefficients using a feedforward neural network. Since they are not directly measured, as in the previous cases, a two-step approach is necessary.

We designate this case as Test_case = 4. The data preprocessing and definition of input–output subspace is performed in the function “mDefCase04.m,” which is called from the main program /FVSysID/chapter08/mainFFNN. The input and output variables of the data preprocessing step are given by

Input variables:	ZAccel	$a_x, a_y, a_z, \dot{p}, \dot{q}, \dot{r}$
	Uinp	$\delta_e, \delta_a, \delta_r, p, q, r, V, \alpha, \beta, \bar{q}, F_{eL}, F_{eR}$
Output variables:	Z	$C_L, C_D, C_Y, C_\ell, C_m, C_n$

where all other variables been defined in the examples discussed in the previous chapters. Although the measured signal for the dynamic pressure is available, it is computed $\bar{q} = \rho V^2 / 2$ using measured true airspeed signal assuming a constant value for the density of air ρ .

The flight data was gathered during a longitudinal motion excited through a multistep elevator input resulting in short period motion and a pulse input leading to phugoid motion. The recorded data are loaded from the data files:

```
load ... \flt_data\hfb320_1_10.asc;
```

and stored in the arrays Z(Ndata,Ny) and Uinp(Ndata,Nu). Since we are analyzing one maneuver and since the angular accelerations ($\dot{p}, \dot{q}, \dot{r}$) were available from the data-processing step, the following information pertaining to number

of time segments being analyzed is not mandatory; however, for the sake of uniformity with the other examples, we specify it as follows:

$$\begin{aligned} \text{Nzi} &= 1; && \% \text{ number of time segments} \\ \text{izhf} &= [\text{Nts1}]; && \% \text{ cumulative index} \\ \text{dt} &= 0.1; && \% \text{ sampling time} \end{aligned}$$

where Nts1 is the number of data points and izhf is the cumulative index at which the maneuvers end when more time segments are concatenated. In the present case the total number of data points is $N = 601$. The above information is needed in the function “ndiff_Filter08.m” to numerically differentiate measured angular rate for concatenated multiple maneuvers using the procedure described in Chapter 2, Sec V. Not all signals from the input array Uinp are required in the present case, but for the sake of similarity with other similar cases it is made oversized.

Having defined the necessary details, the aerodynamic force and moment coefficients are computed according to Eqs. (6.71)–(6.74) in “mDefCase04.m” by calling the data-preprocessing function “umr_reg_hfb”:

$$[Z, \text{Uinp}] = \text{umr_reg_hfb}(Z\text{Accel}, \text{Uinp}, \text{Nzi}, \text{izhf}, \text{dt}, \text{test_case});$$

The computational steps in “ume_reg_hfb.m” are very similar to those described in Chapter 6, Sec. IX.A and carried out in the function “umr_reg_attas.m” for the examples discussed in Chapter 6, Sec. IX.B.

Using the flight-derived aerodynamic coefficients now available from the foregoing preprocessing step in array Z, and the input in the array Uinp, we now formulate the input–output subspace to be modeled by a FFNN as follows:

Number of dependent variables:	3
Dependent variables:	C_D, C_L, C_m
Number of independent variables:	4
Independent variables:	δ_e, α, q, V

The data to be modeled is stored in the arrays $Zout(N, n_y)$ and $Xin(N, n_u)$, respectively. Accordingly, the number of nodes in the input layer is $n_u = 4$ and that in the output layer is $n_y = 3$.

We use the complete length of the time segment ($N_{\text{data}} = 601$) for training as well as for prediction purposes. We specify interactively the number of neurons $N_{\text{nHL1}} = 6$, and the maximum number of iterations $itMax$. Starting from the default values for the network parameters ($\gamma_1, \gamma_2, \mu, \Omega$), through trial and error more suitable values for these parameters yielding good approximation to input–output subspace as well as having good prediction capabilities, were determined. They turn out to be ($\gamma_1 = 0.5, \gamma_2 = 0.5, \mu = 0.125, \Omega = 0.5$). Thus, smaller gains were found to be better in the present case.

A neural network with $N_{\text{nHL1}} = 6$ and $itMax = 2000$ provided typical results for predictive capability, shown in Fig. 8.5. The three plots on the bottom show the input variables δ_e, q, α (the fourth input variable V is not shown in the figure, but can be seen in Fig. 5.5). The three plots on the top show the match for the

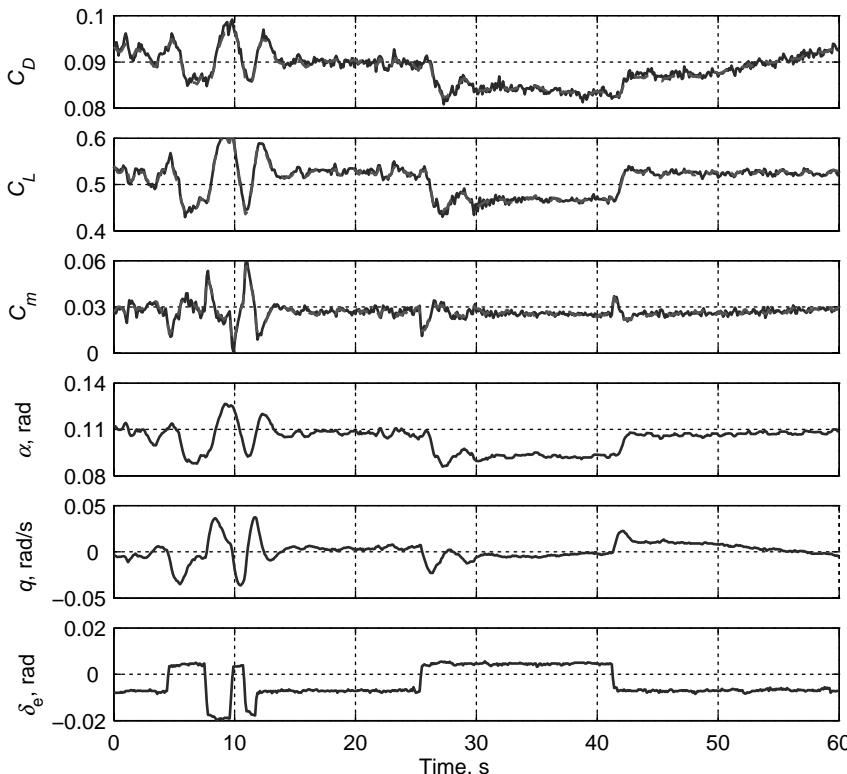


Fig. 8.5 Predictive capability of network trained on longitudinal data with atmospheric turbulence (—, flight measured; - - -, FFNN predicted).

drag, lift, and pitching moment coefficients, respectively. The predictive capability of the trained FFNN for data with atmospheric turbulence is found to be good, and comparable to that obtained using the filter error method (see Fig. 5.6, which shows a match for motion variables). Although a plot is provided with $itMax = 2000$, even fewer iterations provided acceptable modeling capabilities. Both training algorithms, BP with momentum term and modified BP, were tried, and led to similar conclusions, as pointed out for the previous example.

C. Quasi-steady Stall Modeling

The last example pertains to modeling of a complex nonlinear aerodynamic phenomenon. Flight data were gathered with the test aircraft VFW-614 ATTAS at an altitude of 16,000 ft and for clean configuration undergoing quasi-steady stall. In this section we apply the FFNN model without any reference to the physics behind the stall hysteresis. We will reconsider the same example later in Chapter 12 and apply the classical parameter estimation

method to demonstrate that, through advanced modeling concepts, it becomes possible to extract models based on the physics of unsteady aerodynamics.

As in the previous two examples, the first step consists of data preprocessing to compute the aerodynamic force and moment coefficients that we wish to model. These precomputations are performed in the function “umr_reg_attas.m.” The exact details can be easily traced, which are very similar to the procedure described in Chapter 6, Sec. IX.B, and also used in Sec. VII.A. We designate this case as $\text{test_case} = 27$, and accordingly these steps, including definition of input–output subspace are performed in the function “mDefCase27.m” called from the main program /FVSysID/chapter08/mainFFNN. The recorded data from two stall maneuvers (data files \FVSysID\flt_data\fAttas_qst01.asc and fAttas_qst02.asc) are analyzed to obtain the input–output model representation through FFNN. The number of time segments is set to $Nzi = 2$ and the cumulative index $izhf$ set accordingly to $[Nts1; Nts1 + Nts2]$, where $Nts1$ and $Nts2$ are the number of data points in the two individual time segments. We restrict ourselves to longitudinal motion only.

Having derived the aerodynamic force and moment coefficients from the relevant measured data, we now define the input–output subspace to be modeled using FFNN as follows:

Number of dependent variables:	3
Dependent variables:	C_D, C_L, C_m
Number of independent variables:	5
Independent variables:	$\alpha, q, \delta_e, Ma, \dot{\alpha}$

The data to be modeled are stored in the arrays $Zout(N, n_y)$ and $Xin(N, n_u)$, respectively. Accordingly, the number of nodes in the input layer is $n_u = 5$ and that in the output layer is $n_y = 3$.

We use the complete length of the data ($Ndata = 5052$) for the training as well as for prediction purposes. After setting the integer flag $\text{test_case} = 27$, we run the program “mainFFNN.m.” The number of neurons $NnHL1$ and the maximum number of iterations $itMax$ are specified interactively. Through repeated trials, neural network parameters, yielding a good match between measured data and FFNN outputs during training, and having adequate model predictive capability, were determined. In general, it was observed that a somewhat larger number of neurons in the hidden layer was necessary, and the gains of the activation functions were smaller.

It was also observed that finding a suitable FFNN architecture with adequate predictive capability for the lift and drag coefficients, C_L and C_D , was simpler and possible with network tuning parameter close to default values. The modeling of pitching moment coefficient, C_m , having adequate predictive capability, was more difficult to achieve. Since the goal here is limited to demonstrating the FFNN applicability to nonlinear systems and not the exact numerical results, we have not performed rigorous optimization of network tuning parameters and run the training to convergence. For a typical case with ($\gamma_1 = 0.1, \gamma_2 = 0.1, \mu = 0.3, \Omega = 0.5$) and $NnHL1 = 12$, an FFNN trained over 2000 iterations yields the prediction plots shown in Fig. 8.6, giving the time histories.

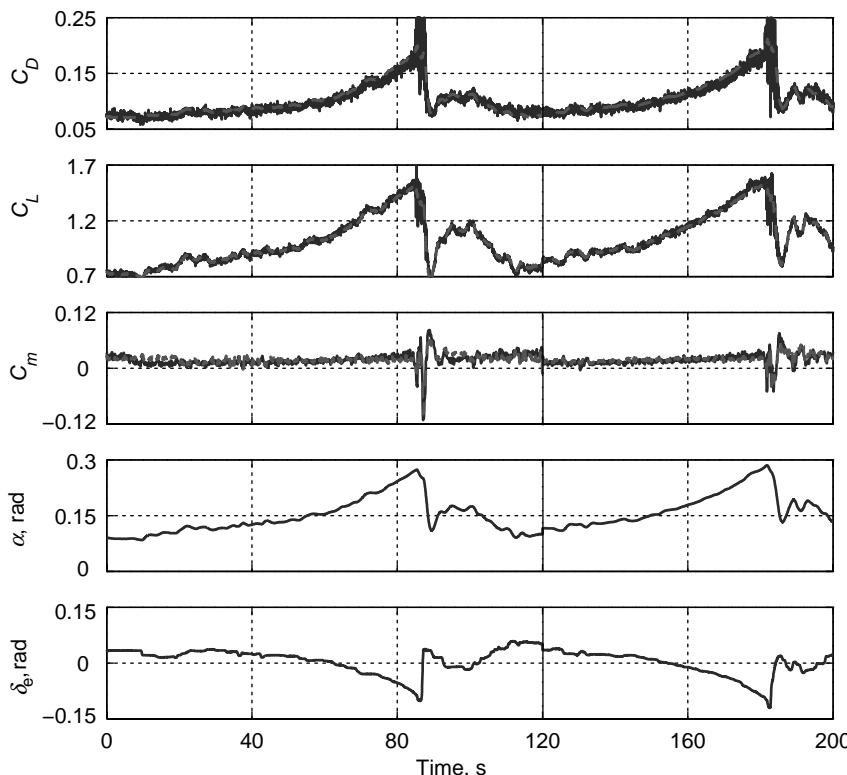


Fig. 8.6 Modeling of quasi-steady stall, time histories (—, flight derived; - - -, model estimated).

The cross plot of the lift coefficient vs angle of attack is shown in Fig. 8.7. The ability of the trained FFNN to reproduce the stall hysteresis is evident from these figures.

Both the training algorithms were investigated. As in the previous two examples, but for the initial faster convergence of the modified back-propagation algorithm, the two algorithms required a large number of iterations. The rate of change of angle of attack is used as one of the input variables. Although $\dot{\alpha}$ is not measured directly, it is generated through numerical differentiation in the data preprocessing step. The other alternative would be to use time signals from the past, $\alpha(k-1)$, $\alpha(k-2)$, and so on, during the k th step.⁴¹

In the above analysis we developed a single NN with 12 nodes in the hidden layer providing functional approximation to a multi-input–multi-output ($n_u = 5$, $n_y = 3$) subspace. As mentioned earlier in this section, tuning of the pitching moment coefficient was considerably more difficult than that of the lift and drag coefficients. To simplify the overall training task, we can use the concepts of modular neural networks (MNN).⁵⁴ Instead of matching

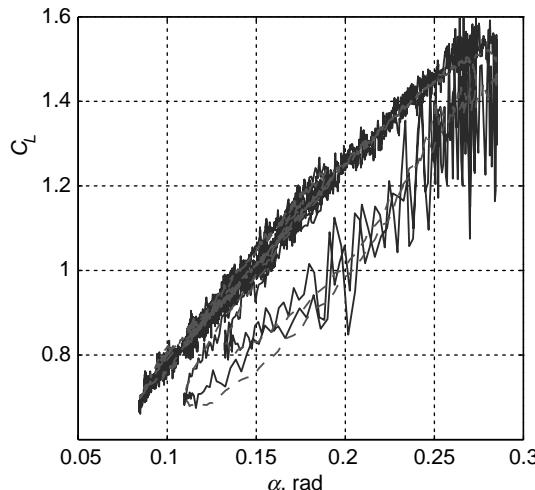


Fig. 8.7 Stall hysteresis (—, flight derived; - - -, estimated).

the multiple outputs through a single network, multiple modules are used, each consisting of a suitable neural network characterizing just one aerodynamic coefficient. Thus, the problem is broken down to submodels, each representing a multi-input single-output subspace. This approach provides more flexibility and it may turn out that smaller neural networks are adequate for each module. Usually, the training task is also simpler. It is also possible to further expand on the MNN concept to incorporate the a priori information based on physical understanding in the network, leading to structured neural networks (SNN). In such networks, instead of modeling the total coefficient as an output of a network, each derivative appearing in the aerodynamic coefficient is modeled as an NN. Of course, the training of such SNNs is more involved than that of the simpler FFNNs we considered here.⁵⁵

We once again highlight the fact that in none of the three examples discussed here have we postulated any models in the classical form as required by the other estimation methods. As demonstrated here, the FFNN approach is solely based on the input–output matching using a black-box approach. It is also evident that the approach to arriving at an appropriate NN architecture, also called topology, is more or less a trial and error procedure. We attempt to avoid under- or overfitting the input–output subspace, which is necessary to ensure good prediction on data not used in the training and also for extrapolation on data outside the range of the training data set. This generalization property depends, in general, on the network topology in terms of number of inputs and number of neurons, in other words on the degrees of freedom, which can be approximated as $n_h(n_u + 2)$. For good generalization, the size of the training set should be much greater than the degrees of freedom.^{5,42} Two different techniques, called regularization and early-stopping, are used to improve this important property. The regularization approach minimizes a cost function comprising error term and

sum of squares of weights, and thereby penalizes large weights which usually lead to poor extrapolation. The early stopping, as the name implies, is an empirical approach and aims to capture the main input–output characteristics. In general, it is not necessary that the larger the number of training iterations is, the better the trained FFNN, because the network starts to fit the noise as well. Recall our observations made in Sec. VII regarding oscillations in the weights as the optimum is approached. For better generalization of FFNN, the early stopping tries to avoid fitting the noise. We do not cover these and other techniques here; they can be found in any standard literature dealing explicitly with neural networks.^{5,7}

VIII. Concluding Remarks

In this chapter we looked at the basic ideas behind the ANN processing and briefly touched on some of the commonly applied networks. We concentrated on just one of them, namely the FFNN which processes the data unidirectionally. The FFNNs are trained using a gradient-based algorithm using back-propagation. We derived the standard version of the same, which optimizes an error cost function with respect to the weights. This was followed by a short description of two extensions. The first was based on adding a momentum term to damp out the parasitic oscillations resulting from rapid changes in the direction of descent. The other modification was based on optimizing an error cost function with respect to inputs to the activation function instead of weights, leading to incorporation of Kalman gain. These extensions were intended to speed up the convergence. A recursive–iterative approach was used to train the FFNNs. The convergence of the modified back-propagation algorithm with Kalman gain is faster during the initial iterations, and leads to a lower minimum during training phase. Usually a large number of iterations is necessary to obtain a trained network with adequate prediction capabilities. Near the optimum the optimization may show a tendency to oscillate, particularly when the data is noisy. Without going into exact details, possible approaches to overcoming difficulties were indicated.

In general, noise in the measurements, complexity of the data subspace, and network parameters play a significant role in NN tuning, affecting the overall convergence and performance. Based on a study pertaining to the lateral-directional motion, some typical values have been suggested for the tuning parameters, which are rough guidelines. Mostly these parameters have to be adjusted in each specific case, as done for the three examples discussed. Several trial and error attempts may be necessary to arrive at an optimal combination of network tuning parameters. A more formal and systematic procedure is necessary to simplify this task.

The flexibility of FFNNs to provide global functional approximation capability is at the cost of higher overheads during the training. For the same size of the input–output data, training of FFNNs requires significantly larger computational time compared with the classical method based on the state space model. Several hundreds to thousands of iterations are not uncommon. On the other hand, once such a NN has been obtained, then the network responses can be computed more easily and with much smaller computational overhead. Hence,

for cases in which we are not particularly interested in physics behind the phenomenon, but only in duplicating the overall input–output characteristics, the black-box model may be an alternative. Owing to the universal approximation feature, the FFNN potentials are enormous, particularly for highly complex non-linear systems. Such networks have found certain acceptance in control applications and also in flight-simulation development. Their suitability for aerodynamic modeling from flight data has been demonstrated, but not yet evolved to an acceptance level enjoyed by the other estimation methods we have covered in the other chapters. The generalization (extrapolation) property, which depends on the NN architecture, remains an important issue. To improve upon this aspect, proper excitation of system is necessary, covering a wide range of static and dynamic data used in training. Generation of such a data rich in information contents usually requires understanding of the physics behind the process.

References

- ¹Zurada, J. M., *Introduction to Artificial Neural Systems*, West, New York, 1992.
- ²Cichocki, A. and Unbehauen, R., *Neural Networks for Optimization and Signal Processing*, John Wiley & Sons, New York, 1993.
- ³Masters, T., *Practical Neural Network Recipes in C++*, Academic Press, San Diego, CA, 1993.
- ⁴Haykin, S., *Neural Networks—A Comprehensive Foundation*, Macmillan, New York, 1994.
- ⁵Hassoun, M. H., *Fundamental of Artificial Neural Networks*, The MIT Press, Cambridge, MA, 1995.
- ⁶Reed, R. D. and Marks II, R. J., *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*, The MIT Press, Cambridge, MA, 1999.
- ⁷Sarle, W. S. (ed.), “Neural Network FAQ, Part 1 to 7,” Periodic posting to the Usenet newsgroup comp.ai.neural-nets; <ftp://ftp.sas.com/pub/neural/FAQ.html>, 1997.
- ⁸McCulloch, W. S. and Pitts, W., “A Logical Calculus of the Ideas Immanent in Nervous Activity,” *Bulletin of Mathematical Biophysics*, Vol. 5, 1943, pp. 115–133. Reprinted in *Embodiments of Mind*, by W. S. McCulloch, The MIT Press, Cambridge, MA, 1965, pp. 19–39.
- ⁹Hebb, D. O., *The Organization of Behavior: A Neuropsychological Theory*, John Wiley & Sons, New York, 1949.
- ¹⁰Rosenblatt, F., “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain,” *Psychological Review*, Vol. 65, No. 6, 1958, pp. 386–408.
- ¹¹Werbos, P. J., “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences,” Ph.D. Dissertation, Applied Mathematics, Harvard University, Cambridge, MA, 1974.
- ¹²Werbos, P. J., *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*, John Wiley & Sons, New York, 1994.
- ¹³Rumelhart, D. E., Hinton, G. E., and Williams, R. J., “Learning Internal Representations by Error Propagation,” in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, edited by, D. E. Rumelhart and J. L. McClelland, The MIT Press, Cambridge, MA, 1986, pp. 318–362.

- ¹⁴Dreyfus, S. E., "Artificial Neural Networks, Back Propagation, and the Kelly-Bryson Gradient Procedure," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 5, 1990, pp. 926–928.
- ¹⁵Narendra, K. S. and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, 1990, pp. 4–27.
- ¹⁶Chu, S. R., Rahamat, S., and Tenorio, M., "Neural Networks for System Identification," *IEEE Control Systems Magazine*, April 1990, pp. 31–35.
- ¹⁷Sjöberg, J., Hjalmarsson, H., and Ljung, L., "Neural Networks in System Identification," *Proceedings of the IFAC Symposium on System Identification and Parameter Estimation*, Vol. 2, 1994, pp. 49–72.
- ¹⁸Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Science*, Vol. 79, 1982, pp. 2554–2558.
- ¹⁹Rao, J. R., "Parameter Estimation of State Space Models by Recurrent Neural Networks," *IEE Proceedings on Control Theory Applications*, Vol. 142, No. 2, 1995, pp. 385–388.
- ²⁰Rao, J. R. and Jategaonkar, R. V., "Aircraft Parameter Estimation Using Recurrent Neural Networks—A Critical Appraisal," AIAA Paper 95-3504, Aug. 1995.
- ²¹Shen, J. and Balakrishnan, S. N., "A Class of Modified Hopfield Networks for Aircraft Identification and Control," AIAA Paper 1996-3428, Aug. 1996.
- ²²Faller, W. E., Smith, W. E., and Huang, T. T., "Applied Dynamic System Modeling: Six Degree-of-Freedom Simulation of Forced Unsteady Maneuvers Using Recursive Neural Networks," AIAA Paper 1997-336, Jan. 1997.
- ²³Hu, Z. and Balakrishnan, S. N., "Parameter Estimation in Nonlinear Systems Using Hopfield Neural Networks," *Journal of Aircraft*, Vol. 42, No. 1, 2005, pp. 41–53.
- ²⁴Powell, M. J. D., "Radial Basis Functions for Multivariable Interpolations: A Review," in *Algorithms for Approximations*, edited by J. C. Mason and M. G. Cox, Oxford University Press, Oxford, 1987, pp. 143–167.
- ²⁵Broomhead, D. S. and Lowe, D., "Multivariable Functional Interpolation and Adaptive Networks," *Complex Systems*, Vol. 2, 1988, pp. 321–355.
- ²⁶Johansen, T. A. and Foss, B. A., "A NARMAX Model Representation for Adaptive Control Based on Local Models," *Modeling, Identification and Control*, Vol. 13, No. 1, 1992, pp. 25–39.
- ²⁷Murray-Smith, R., "A Local Model Network Approach to Nonlinear Modeling," Ph.D. Dissertation, University of Strathclyde, Glasgow, 1994.
- ²⁸Weiss, S. and Thielecke, F., "Aerodynamic Model Identification Using Local Model Networks," AIAA Paper 2000-4098, Aug. 2000.
- ²⁹Giesemann, P., "Identifizierung nichtlinearer statischer und dynamischer System mit Lokalmodell-Netzen," DLR FB 20001-32, Jan. 2002 (in German).
- ³⁰Tsou, P. and Shen, M.-H. H., "Structural Damage Detection and Identification Using Neural Networks," *AIAA Journal*, Vol. 32, No. 1, 1994, pp. 176–183.
- ³¹Rauch, H. E., Kline-Schoder, R. J., Adams, J. C., and Youssef, H. M., "Fault Detection, Isolation, and Reconfiguration for Aircraft Using Neural Networks," AIAA Paper 1993-3870, Aug. 1993.
- ³²Napolitano, M. R., Neppach, C., Casdorph, V., and Naylor, S., "Neural-Network-based Scheme for Sensor Failure Detection, Identification, and Accommodation," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 6, 1995, pp. 1280–1286.

³³De Weerdt, E., Chu, Q., and Mulder, J., "Neural Network Aerodynamic Model Identification for Aerospace Reconfiguration," AIAA Paper 2005-6448, Aug. 2005.

³⁴Hess, R. A., "On the Use of Back Propagation with Feed-Forward Neural Networks for the Aerodynamic Estimation Problem," AIAA Paper 93-3638, Aug. 1993.

³⁵Youssif, H. M. and Juang, J.-C., "Estimation of Aerodynamic Coefficients Using Neural Networks," AIAA Paper, 93-3639, Aug. 1993.

³⁶Linse, D. J. and Stengel, R. F., "Identification of Aerodynamic Coefficients Using Computational Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 6, 1993, pp. 1018–1025.

³⁷Basappa, and Jategaonkar, R. V., "Aspects of Feed Forward Neural Network Modeling and Its Applications to Lateral-Directional Flight Data," DLR IB 111-95/30, Sept. 1995.

³⁸Amin, S. M., Gerhart, V., and Rodin, E. Y., "System Identification via Artificial Neural Networks: Applications to On-line Aircraft Parameter Estimation," AIAA Paper 97-5612, Oct. 1997.

³⁹Ghosh, A. K., Raisinghani, S. C., and Khubchandani, S., "Estimation of Aircraft Lateral-Directional Parameters Using Neural Networks," *Journal of Aircraft*, Vol. 35, No. 6, 1998, pp. 876–881.

⁴⁰Faller, W. E. and Schreck, S. J., "Neural Networks: Applications and Opportunities in Aeronautics," *Progress in Aerospace Sciences*, Vol. 32, 1996, pp. 433–456.

⁴¹Rokhsaz, K. and Steck, J. E., "Use of Neural Networks in Control of High-Alpha Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 5, 1993, pp. 934–939.

⁴²Scharl, J. and Mavris, D., "Building Parametric and Probabilistic Dynamic Vehicle Models Using Neural Networks," AIAA Paper 2001-4373, Aug. 2001.

⁴³Johnson, M. D. and Rokhsaz, K., "Using Artificial Neural Networks and Self-organizing Maps for Detection of Airframe Icing," *Journal of Aircraft*, Vol. 38, No. 2, 2001, pp. 224–230.

⁴⁴Napolitano, M. R. and Kincheloe, M., "On-line Learning Neural-network Controllers for Autopilot Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 6, 1995, pp. 1008–1015.

⁴⁵Yavrucuk, I., Prasad, J. V. R., and Calise, A., "Adaptive Limit Detection and Avoidance for Carefree Maneuvering," AIAA Paper 2001-4003, Aug. 2001.

⁴⁶Rohloff, T. J., Whitmore, S. A., and Catton, I., "Fault Tolerant Neural Network Algorithm for Flush Air Data Sensors," *Journal of Aircraft*, Vol. 36, No. 3, 1999, pp. 541–549.

⁴⁷"Knowledge-Based Guidance and Control Functions," AGARD AR-325, Jan. 1995.

⁴⁸Cybenko, G., "Approximation by Superposition of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems*, Vol. 2, 1989, pp. 303–314.

⁴⁹Hornik, K., Stinchcombe, M., and White, H., "Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks," *Neural Networks*, Vol. 3, 1990, pp. 551–560.

⁵⁰Rao, J. R. and Jategaonkar, R. V., "Artificial Neural Networks for Aerodynamic Modeling," DLR IB 111-94/41, Oct. 1994.

⁵¹Lawrence, S., Giles, C. L., and Tsoi, A. C., "What Size Neural Network Gives Optimal Generalization? Convergence Properties of Backpropagation," Institute for Advanced Computer Studies, University of Maryland, College Park, MD, Technical Report UMIACS-TR-96-22 and CS-TR-3617, 1996.

⁵²Scalero, R. S. and Tepedelenlioglu, N., "A Fast New Algorithm for Training Feed-forward Neural Networks," *IEEE Transactions on Signal Processing*, Vol. 40, No. 1, Jan. 1992, pp. 202–210.

⁵³Raisinghani, S. C., Ghosh, A. K., and Kalra, P. K., "Two New Techniques for Aircraft Parameter Estimation Using Neural Networks," *Aeronautical Journal*, Vol. 192, No. 1011, 1998, pp. 25–29.

⁵⁴Jordan, M. I., "Modular and Hierarchical Learning Systems," in *The Handbook of Brain Theory and Neural Networks*, edited by M. A. Arbib, The MIT Press, Cambridge, MA, 1995.

⁵⁵Keeman, V., "System Identification Using Modular Neural Network with Improved Learning," *Proceedings of the International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing* (NICROSP '96), 1996, p. 40.

This page intentionally left blank

Chapter 9

Unstable Aircraft Identification

I. Introduction

THE DEMANDS of high-performance characteristics such as high maneuverability of modern flight vehicles have led to aerodynamically unstable aircraft configurations. Advancements in modern control theory, microelectronics, and computer techniques make such basically unstable aircraft flyable. As depicted in Fig. 9.1, there are two possible approaches to unstable aircraft identification, namely closed-loop and open-loop identification.¹ Both the approaches pose certain problems; however, the nature of the difficulties encountered is characteristically different in the two cases. The classical closed loop identification approach to model the command to response behavior does not pose any numerical problems, because the overall system is stable and any standard parameter estimation method like for example output error can be applied without encountering any serious difficulty. The main limitation of this approach is that it yields an “equivalent” model, giving overall system representation.

Although highly unstable aircraft can be flown only with the aid of a flight controller, that is, in closed loop, determination of aerodynamic characteristics of the basic unstable aircraft, that is, of the open-loop plant, is of primary interest in several instances: 1) for validation of wind-tunnel and theoretical estimates, and if necessary, updating of these predictions with those from flight data, which is essential for new conceptual designs in particular, 2) for flight envelope expansions, and 3) to support flight evaluation of the control systems and provide more accurate models for design modifications of the stability and control augmentation system. Open-loop parameter estimation in the time domain using the classical output error method poses numerical difficulties, because numerical integration of inherently unstable and highly sensitive systems may lead to numerical divergence in the simulation and optimization.^{2,3} Such difficulties are not encountered in the analysis of marginally unstable phugoid or spiral modes. In the case of highly unstable systems, special techniques and modifications are necessary to prevent the growth of errors introduced by poor initial guess values, round-off or discretization errors, and propagated by inherent instabilities of the system equations. Such approaches are based either on limiting the integration interval or making more efficient use of the observed data.

Assuming that we know the controller exactly (for that matter even inaccurately), identification of open-loop unstable aircraft via closed-loop identification,

although feasible, is rather impractical. To obtain the open-loop parameters of the basic aircraft from such an attempt, it would require incorporation of the models for the controller and actuator dynamics in the estimation procedure. As already pointed out, the overall system being stable, any standard parameter estimation method would work. With the current state of the art, even the increased model size should not be a serious problem. The primary difficulty is to obtain the exact description of the complex control laws containing discrete nonlinearities, and of the actuator performance and controller gains which may be flight-dependent (gain scheduling). Moreover, this approach may result in open-loop parameter estimates with low accuracy. The primary difficulties related to identifiability of the parameters are threefold:

- 1) The controller reacts to the motion and suppresses the oscillatory and transient motion. That is what the controller is designed for; it does its job well. It is, however, detrimental to parameter estimation, because it drastically reduces the information contents required for estimating the parameters.⁴
- 2) Feedback leads to correlated inputs and motion variables.
- 3) Since the measured variables get fed back, the measurement noise in these variables acts as input, thereby introducing a component of stochastic input to the system.

Apart from the modeling of aerodynamics at high angles of attack, various challenges encountered in the unstable or highly augmented aircraft identification can be broadly classified into two categories: 1) choice of a suitable parameter estimation method, and 2) improvement of parameter identifiability.

Besides the least squares techniques, extended Kalman filter, and filter error method, several variants based on the output error approach are possible, such as:⁴ 1) s-plane transformation,⁵ 2) regression startup procedure,^{6,7} 3) output error method with artificial stabilization,^{2,3} 4) equation decoupling,⁸ and 5) multiple-shooting based on solving a two-point boundary value problem.^{7,9} The problems due to the correlated inputs and correlated motion variables can be mitigated by directly deflecting the control surfaces, a technique often called in the literature separate surface excitation.^{4,10} The separate surface excitation may, however, be a complex procedure requiring hardware modifications and often flight certification. Otherwise, the alternative approach would be to attempt parameter estimation based on detection of data-collinearity and mixed estimation. In such cases, however, it may be possible to obtain unbiased

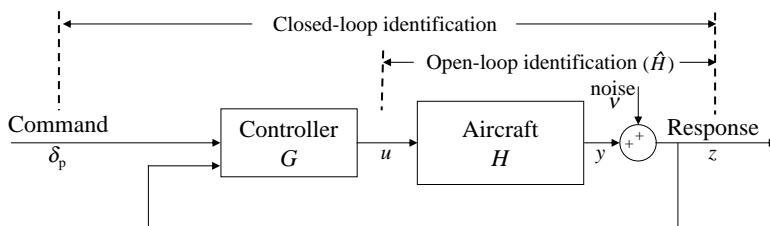


Fig. 9.1 Closed-loop and open-loop identification of unstable aircraft.

estimates of only a subset of parameters or combined derivatives. Moreover, the basic problems of insufficient excitation still persist.

In this chapter we deal with the various issues related to identification of unstable or highly augmented aircraft, and describe several variants of parameter estimation methods applicable to unstable aircraft and techniques to overcome the difficulties encountered in the process. They employ different techniques with different justifying arguments and with varying degrees of complexity to overcome the numerical difficulties. These time domain methods are applicable to both linear and nonlinear systems. In some cases the linearized form of the system equations is used, but only for the sake of clarity. We cover in some depth the issues related to parameter identifiability and suggest possible solutions. Various estimation techniques are applied to the simulated unstable aircraft data, which are provided along with this book for verification. In the majority of the cases, we use the algorithms and software already discussed and provided in the book. A more complex example is presented to bring out a practical application.

II. Basics of Unstable Aircraft Identification

The two possible approaches to unstable aircraft identification, namely the closed-loop and open-loop identification, are shown in Fig. 9.1. The open loop plant is denoted H , the controller G , pilot commands δ_p , and the input to the aircraft (i.e., output of the controller) u . The process noise (atmospheric turbulence) has not been explicitly shown in the figure. The measured system response y is assumed to be corrupted by noise v . Open-loop identification implies determination of the best approximation \hat{H} of the plant model with y as the output and u as the input, that is, input/output relation y/u . The plant input u usually corresponds to the measured control surface deflections, and output y to the aircraft motion variables such as angular rates, attitude angles, flow angles, and accelerations. On the other hand, closed-loop identification implies characterization of the overall system as a lumped model, that is, from pilot input δ_p to the measured output y . This lumped model combines the open loop plant and controller, without explicit representation of each of them. Such a lumped model characterizing the overall response is commonly termed the equivalent model.

As elaborated in Chapters 2 and 4, adequate excitation of the system and exogenous inputs (i.e., inputs generated independent of the system output) are necessary for reliable and accurate parameter estimation. In the present case of unstable aircraft, first we try to gain more insight into these basic requirements, because feedback may dominate the system performance. In the following discussion we draw heavily on the several related aspects presented in Ref. 1.

Considering the linear system representation, we can represent the basic aircraft dynamics as

$$\dot{x} = Ax + Bu + w \quad (9.1)$$

where x is the state vector, u the input (control surface deflections), and w the process noise. Assuming a proportional feedback, the control input u comprises

a linear combination of state feedback and pilot command:

$$u = Kx + \delta_p \quad (9.2)$$

where K is a feedback matrix. Now, rewriting Eq. (9.2) as $Kx + \delta_p - u = 0$, multiplying it with an arbitrary matrix Λ , adding the product to Eq. (9.1) and simple rearranging yields

$$\dot{x} = (A + \Lambda K)x + (B - \Lambda)u + (w + \Lambda\delta_p) \quad (9.3)$$

When open loop identification is attempted from the measurements of x and u , the estimation method treats the last term in Eq. (9.3), that is, the sum $(w + \Lambda\delta_p)$, as noise. Provided the pilot input δ_p is large, the elements of Λ become very small and may, therefore, be neglected in all terms of Eq. (9.3). In such a case, it is then approximately the same as Eq. (9.1) and the feedback, consequently, has only a slight influence on the estimation results. In other words, this implies that the basic aircraft parameters (matrices A and B) can be estimated.

On the other hand, if the pilot input δ_p during the observation period is small or restricted to very short duration, the term $\Lambda\delta_p$ will be negligible compared with the system noise w , even for rather large values of Λ , which cannot then be neglected. In such a case, Eq. (9.3) still has a structure similar to that of Eq. (9.1), but the estimation procedure would yield the combined matrices $(A + \Lambda K)$ and $(B - \Lambda)$, with any arbitrary Λ , instead of the desired A and B .

These difficulties are caused by the fact that, for small δ_p , the input u (control surface deflections) is linearly dependent on the states x , see Eq. (9.2). It can be shown that, in the worst case, with $\delta_p = 0$, it becomes possible to estimate only the feedback gain matrix and not the system matrices.¹¹ In practice, where usually both pilot inputs and process noise are present, the estimation of an open-loop system from closed-loop responses results in biased estimates. The ratio of the aircraft response due to pilot inputs and that due to noise from various sources (for example, feedback noise, atmospheric turbulence or that generated by canards) plays a very important role and is crucial to extracting the aerodynamic derivatives accurately and reliably. Based on numerical studies, a minimum ratio of 3 is necessary for this purpose.^{11,12}

Similar difficulties may arise when aircraft responses are correlated or become linear combinations. This correlation can be expressed in the following form:

$$x = Kx, \quad \text{with } K \neq I \quad (9.4)$$

Multiplying Eq. (9.4) with an arbitrary matrix Λ , and adding the product to Eq. (9.1) yields

$$\dot{x} = [A + \Lambda(K - I)]x + Bu + w \quad (9.5)$$

It is obvious that an arbitrary matrix Λ is involved in the system matrix, the first term in the square brackets on the right-hand side of Eq. (9.5). Following the arguments as in the previous case, here again it becomes difficult to determine the basic system matrix A in the presence of correlated system responses.

The above procedure can also be applied to investigate the influence of other types of controllers, like those with differential or integrating feedback. Having obtained the basic understanding of the issues involved, we do not go into their details, except to state that the problem becomes a little more complex. How to deal with such identifiability problems is discussed in the following sections.

As already pointed out in Sec. I, besides the identifiability issues just discussed, the choice of estimation method is also important for unstable aircraft. In practice, measurement noise is always present and the feedback of such noise-corrupted measured outputs acts as a stochastic input to the system. This stochastic input introduced via feedback acts in addition to the true process noise, which may be caused by atmospheric turbulence or flow disturbances resulting from flight at high angles of attack. It leads to a much more complex situation. The simplest approach to identification of unstable aircraft is to use the regression technique or the maximum-likelihood method implemented in the frequency domain. Several variants of the classical output error method in the time domain as well as the extended Kalman filter are possible, and of course the more advanced approach of the filter error method.

III. Least Squares Method

The least squares method discussed in Chapter 6 can be readily applied to unstable aircraft, because it characterizes the input–output behavior solely from the measured data, and does not involve integration of the state equations. As such the question of numerical divergence of the open-loop plant integration resulting from the system instability does not arise. We may recall from Chapter 6, Sec. II that the LS estimates of the regression equation

$$Y = X\theta + \varepsilon \quad (9.6)$$

are given by the noniterative solution

$$\hat{\theta} = (X^T X)^{-1} X^T Y \quad (9.7)$$

where $Y = [y(1) \ y(2) \ \dots \ y(N)]^T$ is the $N \times 1$ vector of dependent variable y , $\varepsilon = [\varepsilon(1) \ \varepsilon(2) \ \dots \ \varepsilon(N)]^T$ vector of residuals, and X the $(N \times n_q)$ matrix of n_q independent variables; N is the number of data points. Besides the simple noniterative solution provided by Eq. (9.7), the least squares method has another advantage in the present case, namely it accounts for the process noise (that is, system excitation through stochastic input), which may result from stochastic vortex shedding at the aircraft nose. As already pointed out in Chapter 6, Sec. II.B, we once again specifically draw the attention to the fact that the LS estimation accounts for the process noise, because the measured variables contain responses due to deterministic inputs as well as those due to nonmeasurable stochastic inputs. Process noise is different from measurement noise. The measurement noise in the dependent variables is accounted for, but not in the independent variables. This is possible in the total least squares estimation.

In Chapter 6, Sec. VIII we addressed some of the problems encountered in the least squares estimation, indicating that data collinearity (that is, correlation between two independent variables) leads to poor regression results.¹³ In the case of unstable aircraft, data collinearity is pronounced due to the feedback. The integrated flight system causes high correlations between the aircraft states and controls. Hence, special care needs to be exercised while selecting the proper independent variables in the regression equation.

A. Detection of Data Collinearity

Collinearity in the independent variables is mostly detected using eigensystem analysis or singular value decomposition. The correlation matrix of the independent variables or the coefficient of determination (see Chapter 6, Sec. VIII.B) is also sometimes used for this purpose. Theil's measure of multicollinearity is based on the coefficient of determination (i.e., R^2) from multiple partial regressions.¹⁴ It is given by

$$R^2 - \sum_{j=1}^{n_q} (R^2 - R_{-j}^2) \quad (9.8)$$

where R^2 is the coefficient of determination given by Eq. (6.62) for the full model with n_q independent variables, and R_{-j}^2 that for the same model excluding the j th independent variable. The difference $(R^2 - R_{-j}^2)$ is a measure of the contribution of the j th independent variable. Values of Theil's measure close to zero are indicators of absence of collinearity. In this connection, we might recollect some of the limitations of the R^2 criterion discussed in Chapter 6, Sec. VIII.B, which are likely to indirectly affect the above measure of multicollinearity. The correlation coefficients defined in Eq. (6.54) are also an indicator of the individual contributions. However, the correlation matrix of the independent variables is rarely used because the computed correlation coefficients are not reliable in the presence of multiple co-existing near dependencies, and the prediction of data collinearity from such a correlation matrix may not be accurate and reliable.

Eigensystem analysis based on the condition number is a more efficient and reliable approach to assess data collinearity. Here, we may recall from Chapter 4, Sec. X.C the concept of dominant directions based on eigenvalues to avoid the ill-conditioning of the information matrix used in the Gauss–Newton optimization method. Eigenvalues smaller than ϵ (a small positive number), indicating so-called bad directions, were neglected to improve the convergence properties. Data collinearity is usually the most common cause of such ill-conditioning. In the present case we apply the same philosophy to detect the ill-conditioning of the data matrix $X^T X$, which is the information matrix for the LS problem. The $(n_q \times n_q)$ data matrix $X^T X$ being symmetric, it can be written as follows:

$$X^T X = T \Lambda T^{-1} \quad (9.9)$$

where Λ is the diagonal matrix of eigenvalues and T is the matrix of eigenvectors. In the present case, very small or close to zero eigenvalues indicate high correlation among the independent variables.

Alternatively, we can use the condition number which is a well-known measure of a matrix. It is defined as the ratio of the largest to the smallest (nonzero) eigenvalue, of the given matrix. In the present case the ratio C_i of the largest eigenvalues to the eigenvalues pertaining to each independent variable

$$C_i = \frac{|\lambda_{\max}|}{|\lambda_i|} \quad (9.10)$$

which represents a partial condition number and is used for the same purpose. Equation (9.10) yields n_q such individual condition numbers, which can be used to determine the co-existence of multiple dependencies. Condition numbers close to 1 are the most desirable, and those very much greater than 1 are indicators of an ill-conditioned matrix, that is, data collinearity in the present case. We hesitate here to specify a fixed number or to suggest a rule of thumb to decide on the ratio as being large or not, because condition number is relative and a function of the problem size. A wide range of values ranging from 100 to 100,000 or higher have been suggested for the ratio. Thus, some care and engineering judgment is necessary in using this criterion to avoid discarding a perfectly valid independent variable from the regression model.

It is also possible to use the singular values of the matrix X of the independent variables, instead of the eigenvalues of $X^T X$. Following the notation used in Chapter 6, Sec. V, we can write

$$X = U\Sigma V^T \quad (9.11)$$

where $\Sigma = [\text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{nq})]$ is the $(n_q \times n_q)$ diagonal matrix of singular values such that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{nq}$; U and V are the $(N \times n_q)$ and $(n_q \times n_q)$ size left and right singular matrices, respectively. The condition index defined as the ratio of the largest singular value (σ_1) to the individual (nonzero) singular values is given by

$$C_i = \frac{\sigma_1}{\sigma_i} \quad (9.12)$$

Once again, a condition index close to 1 suggests a well-conditioned matrix or the absence of data collinearity, and large values suggest otherwise. As in the previous case of condition numbers based on eigenvalues, here too the choice of a proper decision value is relative and not unique. Although there is no hard and fast rule to judge the condition index, ratios larger than 50 mostly indicate strong data collinearity.

It is apparent from the issues discussed in this subsection that different criteria are possible to investigate the data collinearity, but in an absolute sense none is conclusive. Proper care is necessary in the use of these and other similar test criteria to prevent elimination of a possible independent variable from the regression model.

B. Estimation in the Presence of Data Collinearity

The general approach to deal with the data collinearity is to diagnose the cause and then react accordingly. The two main reasons leading to data collinearity

are: 1) chosen maneuvers (data) do not contain independent excitation of the independent variables, and 2) variables may be linearly dependent, for example α , Θ , γ .

The possible approaches to overcome these difficulties are 1) analyze multiple maneuvers simultaneously or other data sets, 2) reformulate the model to reduce the size, that is, fewer parameters (fewer independent variables), probably leading to combined derivatives, 3) use a priori information about parameter values, if available, and 4) redesign the experiment with better system excitation.

The first choice is a trivial one, because if additional data were available, it should anyway be used to improve the estimates. Additional data collected from similar flight conditions using similar inputs are not likely to improve the estimates, because the cause of difficulties, data collinearity, will persist. The second choice based on estimating a reduced number of derivatives is possible, but it will lead to combined derivatives; one may not obtain estimates of individual derivatives, unless some of them are kept fixed at predefined values. Yet another option which incorporates a priori information is commonly called mixed estimation, which will be addressed mainly in this section. The last option of redesigning the experiment with better system excitation is probably the best choice to overcome the data collinearity problem. It is no doubt more involved and will be treated separately in Sec. XIV.

The basic principle behind mixed estimation is very similar to that of Bayesian estimation introduced in Chapter 4, Sec. XVI, where the probabilistic formulation is used to incorporate the a priori values together with the associated error covariance matrix in the maximum likelihood function. Since the least square estimation is not based on probabilistic theory (see Chapter 6, Sec. II), we resort to an indirect method of constrained optimization to include prior information in the least squares cost function. Referring to Eq. (9.6), if we assume that the a priori values are available for n_{ap} number of parameters ($n_{ap} < n_q$), this prior knowledge is formulated as

$$\theta^* = A\theta + s \quad (9.13)$$

where A is a $(n_{ap} \times n_q)$ matrix of known constants based on the a priori information available, s is a vector of random variables with $E\{s\} = 0$ and $E\{ss^T\} = \sigma^2 W$, and W is a known weighting matrix. Combining Eqs. (9.6) and (9.13) leads to the mixed regression model:

$$\begin{bmatrix} Y \\ \theta^* \end{bmatrix} = \begin{bmatrix} X \\ A \end{bmatrix} \theta + \begin{bmatrix} \varepsilon \\ s \end{bmatrix} \quad (9.14)$$

Assuming σ^2 to be known [possibly estimated using Eq. (6.21)], applying the ordinary least squares procedure to Eq. (9.14), the estimate of θ is obtained as

$$\hat{\theta}_{ME} = (X^T X + A^T W^{-1} A)^{-1} (X^T Y + A^T W^{-1} \theta^*) \quad (9.15)$$

where subscript ME is introduced to denote the mixed estimates of parameters θ . A thorough treatment of mixed estimation including properties of such estimates

is found in Ref. 15. We do not go any further into these details, because of the practical limitations of applying this method, which are discussed next.

Addition of the a priori information helps to reduce the variance of the estimates and thereby improve the reliability of the estimates. Thus, mixed estimation is an indirect way to handle the problems encountered due to data collinearity. Although mixed estimation appears to be a viable approach, we may recollect our pragmatic discussion in Chapter 4, Sec. XVI on the limitations of such an approach, namely specifying the weighting matrix and a priori values correctly. In the absence of precise knowledge about θ^* , σ and W , Eq. (9.15) yields biased estimates.¹³

IV. Total Least Squares Method

As studied in Chapter 6, Sec. V, the TLS method is an extension of the ordinary least squares method, and accounts for measurement noise in both the dependent variables as well as in the independent variables (regressors).^{16,17} The TLS solution is also obtained by an analytical noniterative procedure, and does not involve any integration of the state equations, and hence can be readily applied to an unstable aircraft. As formulated in Eq. (6.48), the regression equation modified to account for the noise in X is given by

$$Y = (X_m - \mu)\theta + \varepsilon \quad (9.16)$$

which is solved by rank reduction using the singular value decomposition of the compounded matrix $[X \ Y]$:

$$[X \ Y] = U\Sigma V^T \quad (9.17)$$

where $\Sigma = [\text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{n_q+1})]$ is the $(n_q + 1 \times n_q + 1)$ diagonal matrix of singular values such that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{n_q+1}$; U and V are the left and right singular matrices. The total least squares solution is then obtained from the column corresponding to the smallest singular value, that is, the last column of V and is given by

$$\hat{\theta}_{\text{TLS}} = -v/\lambda \quad (9.18)$$

where λ is the last component and v is the vector of the first n_q elements of the last column of V .

Since TLS accounts for measurement noise in both the dependent variable as well as in the independent variables, theoretically it is likely to yield better results compared with the LS method. However, in practice the level of measurement noise introduced by the high-quality sensors is very low. Furthermore, stochastic vortex shedding at the aircraft nose may be dominant, which acts as process noise and not as measurement noise. However, we know that the measured variables automatically account for the response due to such stochastic inputs. Thus, from both these aspects, in practice the improvements obtained from TLS compared with LS may turn out to be only marginal. As regards identifiability of parameters, the problems due to presence of data collinearity persist, whether we use OLS or TLS.

V. Combined Output Error and Least Squares Approach

In Chapter 4, Sec. XV we introduced the regression startup procedure to generate better initial values for the output error method.^{6,7} The approach was based on the general idea of using measured states in combination with parameters which were to be estimated, so that starting values did not need to be specified for these parameters. For the other constant terms or fixed parameters appearing in the system model the integrated states were used, without affecting the performance. The use of measured states makes the algorithm behave like the least squares method, and since the LS method does not diverge, inclusion of such state measurements in the output error method tends to prevent the numerical difficulties encountered otherwise. The same method or its variant as below can be applied to unstable aircraft. Although not guaranteed in all cases, the approach seems to work in practice.

Assuming that we know the primary state variable resulting in divergence of a particular mode, for example vertical speed component w associated with longitudinal static instability, and an independent measurement of that state variable is available, the system equations postulated in Eqs. (4.75) and (4.76) can be reformulated as:

$$\dot{x}(t) = A_S x(t) + [B(\beta); \quad A_U(\beta)] \begin{bmatrix} u(t) \\ x_m(t) \end{bmatrix} \quad (9.19)$$

$$y(t) = C_S x(t) + [D(\beta); \quad C_U(\beta)] \begin{bmatrix} u(t) \\ x_m(t) \end{bmatrix} \quad (9.20)$$

The state matrices A and C are resolved into two matrices each, $(A_S; A_U)$ and $(C_S; C_U)$. The matrices with the subscripts "S" and "U" contain the parameter corresponding to stable and unstable states, respectively. This is in contrast to the regression startup procedure of Chapter 4, Sec. XV, in which the system matrices were partitioned based on fixed and free (to be estimated) coefficients. Usually the parameters leading to system instability are fewer than the total number of parameters to be estimated. Thus, the partitioning based on stable and unstable states uses measured states to a lesser extent. In other words it will tend to behave more like the output error method. Accordingly, good initial starting values will be required for such an approach. The regression startup presented in Chapter 4, Sec. XV is more robust from this viewpoint.

VI. Equation Decoupling Method

The basic idea behind this method is as follows: with the aid of state measurements, reformulate the state model in such a way that each differential equation can be integrated independently.^{7,8} Continuing with the linear system representation of Eq. (4.73), partition the system matrix A into two submatrices denoted by A_D and A_{OD} , where A_D is diagonal containing the diagonal elements of A and the matrix A_{OD} the off-diagonal elements. Now, augmenting the control input vector u with the measured states x_m , the state equations can be rewritten as

$$\dot{x}(t) = A_D(\beta)x(t) + [B(\beta); \quad A_{OD}(\beta)] \begin{bmatrix} u(t) \\ x_m(t) \end{bmatrix} \quad (9.21)$$

Since the reformulated system is decoupled, as implied by the diagonal system matrix A_D , it may change the original unstable system into a decoupled stable system. Although, strictly speaking, such a reformulation does not guarantee a stable decoupled system, in general it is found that it does lead to stable systems in practical cases. It is possible to consider various degrees of decoupling, depending upon the resolution of matrix A .

To illustrate the concept on a typical example, we consider a system with four state variables and two control inputs. We denote the corresponding system as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (9.22)$$

For notational simplicity we assume that we have measurements of just four state variables. Thus, the observation equation is given by

$$[y_1 \quad y_2 \quad y_3 \quad y_4]^T = [x_1 \quad x_2 \quad x_3 \quad x_4]^T \quad (9.23)$$

and the corresponding measurement vector by

$$[z_1 \quad z_2 \quad z_3 \quad z_4]^T = [y_1 + v_1 \quad y_2 + v_2 \quad y_3 + v_3 \quad y_4 + v_4]^T \quad (9.24)$$

where v is the measurement noise.

Following the principle of equation decoupling introduced in Eq. (9.21), Eq. (9.22) can now be rewritten as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 \\ 0 & 0 & a_{33} & 0 \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & a_{12} & a_{13} & a_{14} \\ a_{21} & 0 & a_{23} & a_{24} \\ a_{31} & a_{32} & 0 & a_{34} \\ a_{41} & a_{42} & a_{43} & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \quad (9.25)$$

Since the measurements z are corrupted by noise [see Eq. (9.24)], from Eq. (9.25) it is obvious that the off-diagonal matrix A_{OD} introduces process noise. For this reason, application of the output error method accounting for measurement noise only (described in Chapter 4) may pose some difficulties; at least it will affect the accuracy of the estimates, unless the measurement noise is so low that it can be neglected. Furthermore, in a few cases, it will also involve a consideration based on the model definition and measurements available; for example when the postulated model contains the angle of attack or of sideslip as a state variable. These vane measurements are usually less accurate and noisy.

To overcome these uncertainties resulting from process noise, one may consider applying the filter error method that we studied in Chapter 5. This is more likely to yield accurate estimates.¹⁸ However, even under such a case we face a theoretical formulation difficulty. From Eqs. (9.24) and (9.25), one can infer that the process noise introduced is cross correlated. The reader may recall from Chapters 4 and 5 the assumption of independent process and measurement noise, which is not strictly met here. Nevertheless, since noise statistics are of secondary importance and also because the filter error method is relatively insensitive to the actual distribution of process and measurement noise, the filter error method applied to equation decoupling is likely to perform well. The necessity of applying the filter error method, however, raises another interesting aspect that is seldom discussed. As soon as we have to apply the filter error method, then there is no necessity of equation decoupling to integrate the states, because the filter error method, as will be discussed in Sec. VIII, is numerically stable due to inherent feedback. Similarly, equation decoupling implemented in the frequency domain is superfluous, because the output error method implemented in the frequency domain, as will be discussed in Sec. XIII, is by nature stable and does not warrant any further measures to overcome numerical difficulties encountered in the time domain implementation.¹⁹

The equation decoupling method, although illustrated on a linear system, is equally valid and applicable to nonlinear systems. The equation decoupling method, as becomes evident, is a variation of the regression method.

VII. Eigenvalue Transformation Method

One of the earliest approaches to identification of unstable aircraft is based on an extension of a controller design philosophy applied to unstable systems. It is based on finding a suitable s-plane transformation to transform the basic open-loop unstable plant into a stable plant. Eigensystem analysis is one way to investigate the system stability; we know that the eigenvalues and eigenvectors of the system state matrix completely characterize the system behavior; the eigenvalues determine the nature of the transient response whereas the eigenvectors determine the amplitude of that response. For an unstable plant, one or more eigenvalues have positive real parts and lie in the right half of the $\sigma-j\omega$ plane. Shifting the eigenvalue corresponding to the largest positive real part to the left half of the s-plane and scaling the complete system proportionately would lead to a transformed system that is open-loop stable.⁵ This principle of eigenvalue transformation is shown in Fig. 9.2.

Selecting a real positive σ_T larger than the largest real part of the most unstable eigenvalue, the state variables x , the observation variables y , and the control variables u are transformed through an exponential function:

$$\tilde{x}(t) = e^{-\sigma_T t} x(t), \quad \tilde{y}(t) = e^{-\sigma_T t} y(t), \quad \tilde{u}(t) = e^{-\sigma_T t} u(t) \quad (9.26)$$

where \tilde{x} , \tilde{y} , \tilde{u} are the transformed state, observation, and input variables respectively. Since we are dealing with measured data containing noise, it is also imperative to transform the measurement equation and for the same reason the process

UNSTABLE AIRCRAFT IDENTIFICATION

307

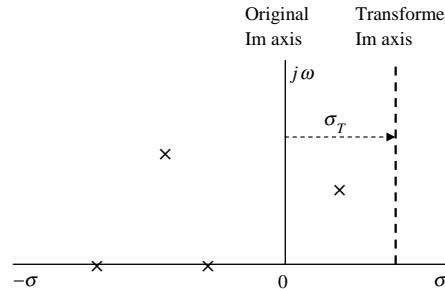


Fig. 9.2 Eigenvalue transformation approach.

noise affecting the state variables. Accordingly,

$$\tilde{z}(t) = e^{-\sigma_T t} z(t), \quad \tilde{w}(t) = e^{-\sigma_T t} w(t), \quad \tilde{v}(t) = e^{-\sigma_T t} v(t) \quad (9.27)$$

Substituting Eqs. (9.26) and (9.27) in the system model postulate of Eqs. (3.24) and (3.25) and through simple derivation, it can be shown that the transformed system can be represented as

$$\dot{\tilde{x}}(t) = (A - \sigma_T I)\tilde{x}(t) + B\tilde{u}(t) + F\tilde{w}(t) \quad (9.28)$$

$$\tilde{y}(t) = C\tilde{x}(t) + D\tilde{u}(t) \quad (9.29)$$

$$\tilde{z}(t_k) = \tilde{y}(t_k) + G\tilde{v}(t_k) \quad (9.30)$$

The above transformation modifies the eigenvalues, but the relative location in the s-plane is not altered. It is obvious that the system matrix $\tilde{A} = (A - \sigma_T I)$ will have all stable eigenvalues, that is, having negative real parts, because we have chosen σ_T larger than the eigenvalue with largest positive real part.

Thus, the transformed system being stable, any standard estimation algorithm, for example the output error method discussed in Chapter 4, can be applied without encountering numerical difficulties. For parameter estimation purposes, the given set of flight data, that is, time histories at discrete time points, of the output variables z and of the input variables u are multiplied by $e^{-\sigma_T t}$. To these transformed time histories we fit the model given by Eqs. (9.28)–(9.30). Since we are interested in applying the output error method, we neglect the process noise $w(t)$. Estimation accounting for process noise requires the filter error method which is, as will be discussed in Sec. VIII, by its formulation numerically stable, and will not require the technique of eigenvalue transformation. It is necessary only when we would like to apply the output error method.

In such a case, the estimates of the parameters are obtained by applying the Gauss–Newton method:

$$\sum_{k=1}^N \left[\frac{\partial \tilde{y}}{\partial \Theta} \right]^T \tilde{R}^{-1} \left[\frac{\partial \tilde{y}}{\partial \Theta} \right] \Delta \Theta = \sum_{k=1}^N \left[\frac{\partial \tilde{y}}{\partial \Theta} \right]^T \tilde{R}^{-1} [\tilde{z}(k) - \tilde{y}(k)] \quad (9.31)$$

where $\Delta\Theta$ contains the elements of the matrices $\tilde{A}(=A - \sigma_T I)$, B , C , D and possible biases. The covariance matrix of the noise \tilde{v} is obtained as

$$\tilde{R} = \frac{1}{N} \sum_{k=1}^N [\tilde{z}(t_k) - \tilde{y}(t_k)][\tilde{z}(t_k) - \tilde{y}(t_k)]^T \quad (9.32)$$

which is related to the measurement covariance matrix R through $\tilde{R} = e^{-2\sigma_T t} R$.

From Eqs. (9.28) and (9.29) we note that only the diagonal elements of state matrix A are modified through σ_T . This needs to be properly taken care of while coding the state and the observation equations. If the state equations are programmed as $\tilde{x} = \tilde{A}\tilde{x} + \tilde{B}\tilde{u}$, then some care needs to be taken while coding the observation equations because the matrix C in Eq. (9.29) contains the original parameters, whereas the parameters in \tilde{A} appearing in Eq. (9.28), which will be estimated, are for the transformed system matrix. Therefore, if any diagonal elements appearing in the state matrix also appear in the matrix C , then they should be back-transformed before using them in the computation of the observation variables. Such a case arises when we consider accelerations as the observation variables. Failure to do so will lead to a mismatch in those particular parameters in the state and observation equations. After the estimation, for further use the parameters of the original state matrix can be obtained by back-transformation, $A = \tilde{A} + \sigma_T I$. Alternatively, we can program the state equations exactly as in Eq. (9.28), as $\tilde{x} = (A - \sigma_T I)\tilde{x} + \tilde{B}\tilde{u}$, in which case the data will be processed in terms of the transformed system, but the numerical estimates will correspond directly to those of the original system. We will follow the second approach, which is more direct, in the example presented in Sec. XVI.A.

Although the approach looks promising, it has one limitation, namely the data transformation is based on an exponentially decaying function $e^{-\sigma_T t}$, which approaches zero as t becomes larger. This limits the time-length of data that can be analyzed using this method,⁵ not so much because of the noise properties, but mainly due to the fact that the transformed variables \tilde{z} and \tilde{u} approach zero and thereby tend to numerically lose information content. The larger the instability, the larger σ_T will be, and the smaller the record length will be that can be appropriately analyzed using this method.

VIII. Filter Error Method

The filter error approach to parameter estimation introduced in Chapter 5 is the most general one, which basically caters for the need to analyze flight data gathered in a turbulent atmosphere.^{20,21} Two versions considering steady-state and time-varying filters were introduced and their applicability to general nonlinear systems brought out. Since the exact algorithm has already been presented in full detail, we recapture here just the essentials of the state estimator using a steady-state filter discussed in Chapter 5, Sec. V.A. The state estimator consists of a prediction and a correction step given by

$$\tilde{x}(t_0) = x_0 \quad (9.33)$$

$$\tilde{y}(t_k) = g[\tilde{x}(t_k), u(t_k), \beta] \quad (9.34)$$

$$\hat{x}(t_k) = \tilde{x}(t_k) + K[z(t_k) - \tilde{y}(t_k)] \quad (9.35)$$

$$\tilde{x}(t_{k+1}) = \hat{x}(t_k) + \int_{t_k}^{t_{k+1}} f[x(t), \bar{u}(t), \beta] dt \quad (9.36)$$

where \tilde{x} is the predicted state vector, \hat{x} the updated state vector, and K the filter-gain matrix. The zero shifts Δu and Δy have been not shown in the above equation, but as discussed in the previous chapters will have to be accounted for during the estimation.

As evident from Eq. (9.35), the state estimators incorporate measurements of the observation variables $z(t_k)$. A feedback proportional to the fit error updates the predicted state variables. This feedback stabilizes numerically the filter algorithm and also helps to improve the convergence properties of the estimation algorithm. Owing to this stabilizing property of the filter error algorithm, it can be readily applied to unstable systems without encountering numerical divergence problems. The filter feedback matrix K need not bear any relation to the control system feedbacks that actually help fly the unstable aircraft.

Thus, due to its basic formulation, the filter error method is very attractive for unstable systems for two reasons: 1) it accounts for the stochastic input (process noise) resulting from stochastic vortex shedding at the aircraft nose, for example, and 2) without any additional modifications it helps to overcome the divergence problems which are likely to be encountered in the numerical integration. Since the filter error method allows estimation of process noise as well as measurement noise and simultaneously the systematic errors in the measurements as well, it provides a complete solution to the estimation problem. In practical applications, however, the approach has certain limitations, discussed in Chapter 5, Secs. XII and VII, namely 1) good response match tends to mask the modeling discrepancies, and 2) usually the method is applicable to single maneuver or to maneuvers carried out under similar atmospheric conditions. In the present case the first limitation will, as in a general case, affect the model development process. The second restriction is not a serious drawback because the major problem we are addressing is to avoid numerical divergence due to system instability during analysis of maneuvers carried out in calm air or similar weather conditions.

IX. Extended and Unscented Kalman Filters

For the very same reasons that we have just discussed in the case of filter error method, namely incorporation of a feedback proportional to the residual error, the extended or unscented Kalman filter can also be applied to open-loop identification of unstable aircraft. The difference between the filter error approach and the nonlinear filtering approach is that the former is a direct approach that minimizes iteratively the maximum likelihood cost function applying the Gauss–Newton method, whereas EKF/UKF is an indirect approach to parameter estimation treating the unknown parameters as additional states; see Chapter 7, Sec. III.

Any suitable implementation of the EKF and UKF would serve the purpose here, for example as given in Eqs. (7.52)–(7.61). Of course, the problems of

specifying the measurement noise and process noise distribution matrices still persist in this case too. Improper values of such noise distribution matrices may influence the estimates adversely. Nevertheless, it is a viable approach that does not face numerical problems.

X. Output Error Method

The output error method is undoubtedly the most commonly used parameter estimation method for stable systems. Its application to unstable systems encounters many practical difficulties, not due to the theoretical formulation but mainly due to the numerical integration of unstable system in an open loop, which is sensitive to the errors in the initial conditions and system parameters. In our case, since the aerodynamic parameters are unknown, added to the unknown or inaccurately known initial conditions, it may lead to a diverging solution, more often than not exceeding the numerical range of the computer. This behavior is caused by characteristic attributes of the single-shooting approach, that is, of the initial-value problems, incorporated in the output error method.

Although extremely laborious in practice, one can attempt to apply the output error method to unstable systems by analyzing a given maneuver in multiple steps. Starting from a very small initial portion, estimating the parameters and then successively using the improved estimates and extending the duration of the maneuver being analyzed, in some cases it does become possible to apply the output error method to sensitive cases such as unstable systems. In some rare cases such an approach has been applied.²² Besides the approach being laborious, it does not guarantee convergence in every case and it is also difficult to apply to multirun evaluation. A more convenient approach for routine applications is desirable.

XI. Output Error Method with Artificial Stabilization

To overcome the practical difficulties just discussed in the previous section, by analogy to the filter error methods possessing stabilizing properties, the output error method is extended to provide artificial stabilization.^{2,3} Although such an artificial stabilization is arbitrary and does not have the sound theoretical basis of the filter error method, the approach is seen as a practical solution to extend the commonly accepted output error method for parameter estimation to unstable aircraft.

Such an algorithm is parallel to the filter error algorithm of Eq. (9.35), except that the updated state vector is now computed according to

$$\hat{x}(t_k) = x(t_k) + S[z(t_k) - y(t_k)] \quad (9.37)$$

where S is an arbitrarily selected stabilization matrix of size $(n_x \times n_y)$, assumed to be independent of the system parameters and initial conditions. This overcomes the need for computation of the filter gain matrix, which is the computationally complex and time-consuming part of the filter error method.

Two limiting values of S are the null and the identity matrix, provided the units of states x , measurements z , and model output y variables are compatible with

each other. For $S = 0$ the algorithm corresponds to the output error method. In the other case of $S = I$ it reduces to the equation error method, provided only states are used as the observation variables. This is obvious from Eq. (9.37), in which substitution of $y = x$ on the right-hand side leads to $\hat{x} = z$, the measured states. In a more general case, when the observation vector y contains not only the states but other variables as well, in the limiting case the matrix S will not be an identity matrix, but will have to be specified appropriately.

Introduction of the stabilization matrix S affects the parameter estimates. This influence, however, will be small provided the elements of S are selected to be small. Its influence will in any case be minimized when the output error is small, that is, when modeling errors are small. Furthermore, the choice of the type of stabilization, that is, which fit error to be fed back to which state variable, is also neither defined nor obvious. This may call for some engineering judgment. Usually, fit error in the states as outputs is fed back to the states being integrated, thereby increasing the static stability or the system damping. The choice of variables to be fed back may also depend on the degree of confidence in the particular measurement. The measurements of rates are more reliable than those of the flow angles. Stabilization by feeding back other variables is possible, but not so direct. Some additional care will be required in specifying the values for elements of S , if the states, measurements, and model outputs differ in their units.

In addition to stabilizing the system solution, stabilization is also necessary of the solutions to the perturbed state equations used in numerically approximating the sensitivity coefficients required in the optimization (see Chapter 4, Sec. VIII).^{3,7} By partially differentiating Eq. (9.37), it can be shown that the same stabilization matrix can be used to stabilize the perturbed state solutions.

XII. Multiple Shooting Method

The commonly applied technique to solve estimation problems is fitting the unknown parameters in an iterative procedure incorporating a solution to an initial-value problem (single shooting). In this approach, optimization of a scalar function does not make efficient use of the primary information available to the estimation procedure, which is defined by the measured trajectories. In contrast, the multiple shooting approach accounts more efficiently for the provided information.

The basic concept of the multiple shooting approach is to subdivide the integration interval $[t_0, t_f]$ by a suitable choice of a grid:

$$g^m: \quad \{t_0 = \tau_1 < \tau_2 < \dots < \tau_m = t_f\} \quad (9.38)$$

into $(m - 1)$ intervals; see Fig. 9.3. The corresponding initial-value problems

$$\dot{x}(t) = f(t, x, \Theta), \quad x(\tau_j) = \sigma_j, \quad t \in [\tau_j, \tau_{j+1}] \quad (9.39)$$

with the additional variables $(\sigma_1, \dots, \sigma_m)$ as estimates of the states $x(\tau_j)$ can be solved independently using classical integration methods, such as the Runge-Kutta algorithm. In order to ensure continuity of the final solution over the

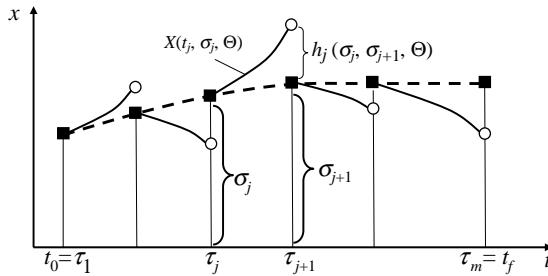


Fig. 9.3 Basic principle of multiple shooting method.

entire time interval, additional constraints

$$h_j(\sigma_j, \sigma_{j+1}, \Theta) = x(\tau_{j+1} | \sigma_j, \Theta) - \sigma_{j+1} = 0; \quad j = 1, \dots, (m-1) \quad (9.40)$$

have to be satisfied. This technique transforms the original identification problem into a constrained least squares problem:

$$\min_{\tilde{\Theta}} J(\tilde{\Theta}) \quad \text{subject to} \quad h_j(\sigma_j, \sigma_{j+1}, \Theta) = 0 \quad (9.41)$$

where $\tilde{\Theta} = [\sigma_1^T \ \sigma_2^T \ \dots \ \sigma_m^T \ \Theta^T]^T$ is the extended unknown parameter vector and the equality constraints h_j are defined in Eq. (9.40).

A direct solution to this problem is provided by constrained optimization. A more efficient approach is, however, to recognize the special structure and to apply the Gauss–Newton method in combination with a condensation algorithm. The advantage of the multiple shooting method is that, by defining additional grid points (σ_j), the computed solution remains close to the observed data. Exponential growth of the parasitic components in the solution of unstable differential equations can be restricted. The choice of grid points is one critical aspect in the application of this method, influencing both estimates and the convergence. The grid points may be equidistant and can be either specified a priori or can also be determined automatically by specifying a tolerance limit for the error growth.

The foregoing approach of subdividing the integration interval into several segments is not the same as that of evaluating simultaneously multiple time records. In the multiple run evaluation applying the output error method, the individual time segments are treated independently, whereas the multiple shooting approach enforces continuity of the solution at the grid points. Although multiple shooting is a viable approach, for the reasons to be elaborated in Sec. XV, we have restricted the description here to indicate the approach without going into specific methodical details. A detailed description of the formulation is found in Refs. 9, 23, and 24.

XIII. Output Error Method in Frequency Domain

Since the main difficulties of the output error method when applied to unstable systems originate from the diverging solution resulting from the integration of the state equations, an alternative approach would be to implement the method in the frequency domain. The transformation of the system equations to the frequency domain leads to a set of algebraic equations. Thus, there is no integration of the states involved in the frequency domain, which automatically eliminates the numerical divergence problems resulting from the unstable eigenvalues. Assuming periodic signals, $x(0) = x(T)$, applying the Fourier transformation, the system equations (3.24) and (3.25) are transformed into^{4,25,26}

$$j\omega x(\omega) = A(\beta)x(\omega) + B(\beta)u(\omega) \quad (9.42)$$

$$y(\omega) = C(\beta)x(\omega) + D(\beta)u(\omega) \quad (9.43)$$

$$z(\omega_l) = y(\omega_l)v(\omega_l), \quad l = 1, \dots, N \quad (9.44)$$

where $x(\omega)$, $u(\omega)$, $y(\omega)$ are the Fourier transformed state, input and observation variables, $\omega_l = 2\pi l/T$ is the l th discrete frequency and $T = N\Delta t$. The cost function to be minimized is then given by

$$J_{\text{FR}} = \sum_{l=1}^M [z(\omega_l) - y(\omega_l)]^* S_{vv}^{-1} [z(\omega_l) - y(\omega_l)] + \log |S_{vv}| \quad (9.45)$$

where M is the number of frequencies to be evaluated and S_{vv} is the spectral density matrix of the measurement noise; the subscript FR is appended to the cost function to distinguish between the frequency domain and time domain. Minimization of Eq. (9.45) by the Gauss–Newton method yields the maximum likelihood estimates of the parameters β , fitting the Fourier coefficients of the measured data and the model response. Note that the estimated parameters β are still the same stability and control derivatives as in the time domain method, and have the same physical definition. They do not represent or correspond in any form to transfer function models, which are commonly used for model representation in the frequency domain. The distinctive features of such an approach are that

- 1) Optimization amounts to fitting the frequency spectra, involving algebraic equations; hence instabilities do not affect the estimation.
- 2) Without affecting the estimation results, the zero frequency can be neglected in the evaluation, which is advantageous in eliminating the need to estimate a large number of initial conditions and bias terms, and thereby help to reduce the problems of correlation between the bias parameters and aerodynamic derivatives. For multirun evaluations, bias parameters and initial conditions often far exceed the number of aerodynamic derivatives.
- 3) As recorded flight data usually do not meet the requirements of being periodic, extensions to Eqs. (9.42)–(9.45) are necessary to include corrections terms that allow use of nonperiodic data.^{26,27}

Although the estimated parameters are directly the stability and control derivatives, the computed responses are in the frequency domain. For comparison of time histories, they have to be back-transformed into time domain, which is not critical because it does not involve numerical integration. Although the output error method implemented in the frequency domain has some advantages, the major limitation is that it is applicable only to linear systems. For unstable aircraft, however, it provides a useful alternative approach.

XIV. Separate Surface Excitation

As pointed out in Sec. III.B, better system excitation is probably the best choice to reduce data collinearity. Better excitation of unstable aircraft for the specific purpose of parameter estimation is possible through separate surface excitation (SSE), also sometimes termed as single surface excitation.¹⁰ The basic idea behind this approach is to feed separate inputs, bypassing the flight control system, to directly deflect the individual control surfaces. In most of the cases it is expected that, with minor modifications or through judicious use of existing hardware, it would be possible to inject such controlled inputs. For example, in the case of the X-31A aircraft, the flutter test box was used advantageously for this purpose by integrating a dedicated signal generating card; see Fig. 9.4. Although theoretically it would be possible to inject extra inputs simultaneously to multiple control surfaces, it is preferable to excite them one at a time for the very reason of reducing collinearity.

Reduction of data collinearity through SSE is shown in Fig. 9.5. This figure shows time history plots of 1) pitch and PID command, 2) angle of attack, 3) canard deflection, and 4) thrust-vector (TV) deflection and symmetric trailing edge flaps for three independent flight maneuvers, demarcated by vertical lines. The first maneuver pertains to a pitch doublet commanded by the pilot. It is seen that the TV deflection and symmetric trailing edge flaps are completely correlated with each other and also with the canard deflection. The second and the third maneuvers show SSE inputs, first to the canard and then to the symmetric trailing edge flaps (that is, elevator). From the second maneuver it is observed

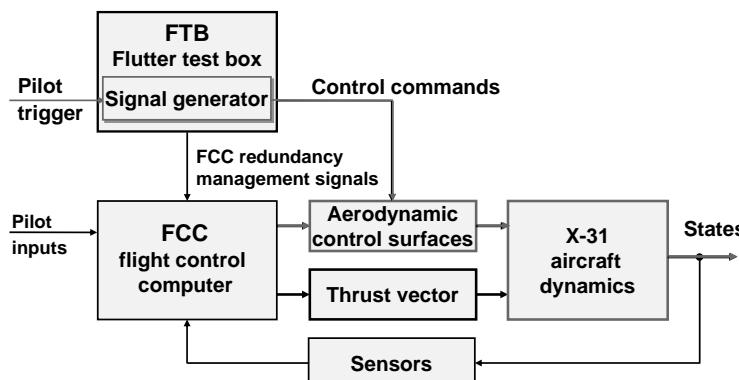


Fig. 9.4 Realization of separate surface excitation.

X-31A:
pilot input and
separate surface
excitation

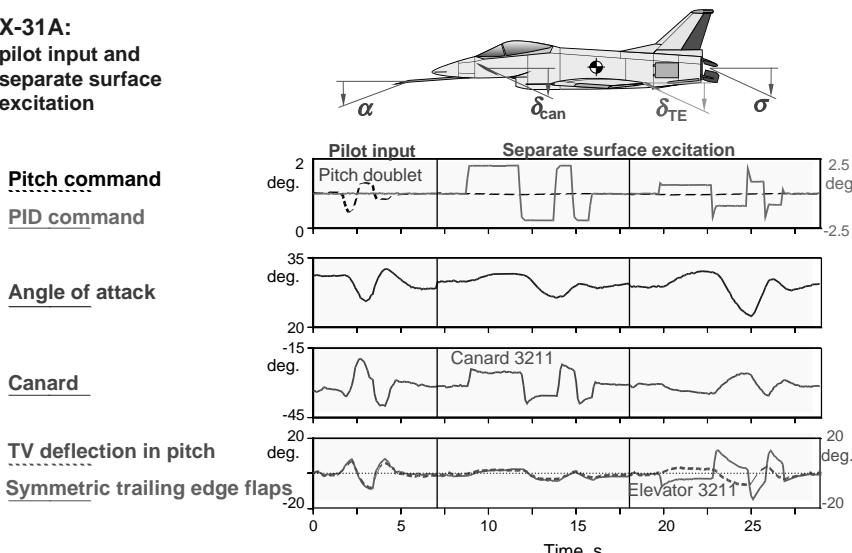


Fig. 9.5 Reduction of data collinearity through separate surface excitation.

that the canard deflection is now different compared with the TV and symmetric trailing edge flaps. During the third maneuver a separate input to trailing edge flaps forces elevator deflection uncorrelated with the TV deflection. Thus, combining these two SSE maneuvers provides adequate information to estimate the respective control surface effectivenesses.

It is obvious that during such SSE input maneuvers the feedback is active, which will react in due time to damp out the dynamic motion resulting from SSE and also result in correlated inputs and motion variables. However, initial aircraft response to the directly fed inputs is characteristically different compared with that if the same inputs were fed through the flight control system. This initial response and decoupled control surface deflections allow accurate estimation of the individual control surface effectiveness. Furthermore, improved estimation of the control surface effectiveness combined with reduced correlation between the motion variables and control surface deflections leads to improved estimation of the other derivatives as well. Figure 9.6 shows the estimates $C_{m\delta\text{can}}$, pitching moment due to canard, obtained from pilot input maneuvers and from SSE input maneuvers.^{4,10} The vertical bars denote the standard deviations. It is clearly seen that the pilot input maneuvers yield estimates with large standard deviations and the scatter in the estimates is large too. On the other hand, SSE input maneuvers provide more accurate and reliable estimates as implied by the low standard deviations (vertical bars), and significantly reduced scatter.

The brief discussion of SSE provided in this section is typical of the issues pertaining to excitation of highly augmented unstable aircraft. For a detailed treatment of various practical issues in connection with the X-31A aircraft, the reader is referred to Refs. 10, 28 and 29.

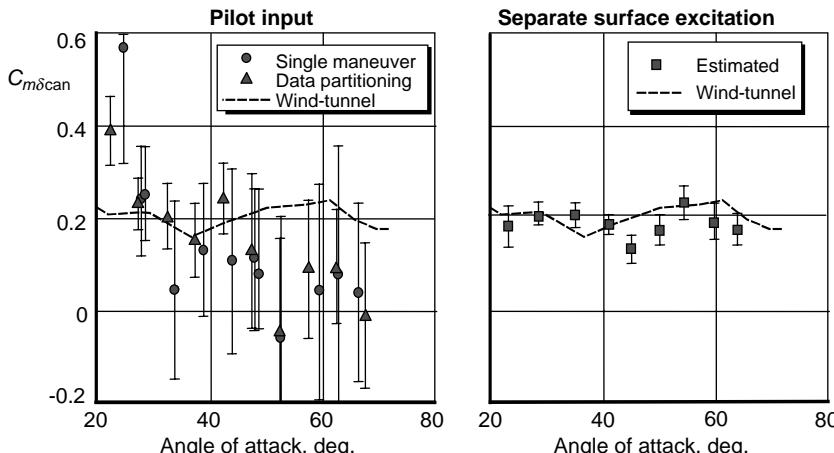


Fig. 9.6 Improved estimation through separate surface excitation.

XV. Programming Considerations

In order to compare the complexities of modifying existing estimation software or writing new to provide capabilities of analyzing unstable systems, it is reasonable to assume that software for the output error algorithm is available, including the one provided in this book. Application of regression startup or of the equation decoupling method does not warrant any estimation program modifications. Only the postulated model needs to be properly coded. This will be highlighted on a test example presented in the next section. To include artificial stabilization in the output error method requires some software modifications. These changes are comparatively minor and can be incorporated in the existing software without much developmental time. The basic output error software described in Chapter 4, Sec. XIX and provided with the book has been extended to provide such an option through a flag “iArtifStab.” For the default value of iArtifStab=0, the program runs in the mode of classical output error method. Artificial stabilization is invoked by setting this flag to any value greater than zero. In such cases it is also necessary to specify the stabilization matrix S(Nx,Ny) appearing in Eq. (9.37). The direct and simplest approach is to feed back errors in the observed states to the respective states.

On the other hand, the filter error algorithm may not be universally available. New development of such software and its validation is time-consuming and costly. However, a few estimation software packages do exist,^{20,21} and experience gained can be used to minimize the developmental costs. A basic version of the software for the filter error method has been provided here. The multiple shooting method for aircraft parameter estimation is a relatively new approach. The programming complexity of the multiple shooting method with condensation algorithm is of the same order as that of the filter error method. Although some investigations help to establish the feasibility of the approach,^{7,9} we do not go into any further details of such an approach or software using this technique because other options are more direct and well established.

XVI. Examples

The various algorithms discussed in the foregoing sections are applied to two test cases. The first pertains to simulated unstable aircraft responses to critically evaluate the performance of the estimation methods under a controlled environment. As in the case of Chapter 5, Sec. XI.A, dealing with analysis of data in turbulence, we once again resort to simulated data, mainly because 1) it provides data generated under known external initial conditions and noise characteristics, and 2) no modeling errors are involved in the postulated model, both of which affect the estimates. This is all the more true in the case of unstable systems. The second example pertains to estimation of lateral-directional derivatives of X-31A, which is a highly control-augmented fighter with enhanced maneuverability.

A. Simulated Unstable Aircraft Response Data

A simplified model pertaining to the short period motion of the test aircraft de Havilland DHC-2 "BEAVER" is considered here.³⁰ The nominal values of the aerodynamic derivatives used to generate the responses were obtained by parameter estimation from flight data in a typical flight condition. The static stability parameter M_w is, however, so adjusted as to result in an unstable system with a time-to-double of 1 s. To provide realistic control inputs, the pilot input actually applied in a particular flight are used.³⁰ A total of 12.5 s of data with a sampling time of 0.05 s were generated.³ The state and observation equations are given by *State equations*

$$\begin{aligned}\dot{w} &= Z_w w + (U_0 + Z_q)q + Z_{\delta_e} \delta_e \\ \dot{q} &= M_w w + M_q q + M_{\delta_e} \delta_e\end{aligned}\quad (9.46)$$

Observation equations

$$\begin{aligned}a_z &= Z_w w + Z_q q + Z_{\delta_e} \delta_e \\ w &= w \\ q &= q\end{aligned}\quad (9.47)$$

where w denotes the vertical velocity, U_0 the stationary forward speed, q the pitch rate, a_z the vertical acceleration and δ_e the elevator deflection. Numerical simulation of such an unstable aircraft is feasible only through incorporation of a suitable controller. A feedback proportional to the vertical velocity is used.

$$\delta_e = \delta_p + Kw \quad (9.48)$$

where δ_p denotes the pilot input. The feedback K is so chosen as to result in the original degree of stability for the closed loop system.³

Although the data is generated through a closed loop simulation, it is attempted to identify the system in open loop, by treating δ_e , the elevator deflection, as the control input. The controller is not considered in the estimation procedure. The nominal (true) values of the parameters are provided in the second column of Table 9.1. The eigenvalues of the system matrix

$$A = \begin{bmatrix} Z_w & U_0 + Z_q \\ M_w & M_q \end{bmatrix} \quad (9.49)$$

Table 9.1 Summary of estimates and model definitions for various estimation methods: simulated unstable short period motion

	Nominal value	LS	TLS	LS/OEM	EqDecpl	EigTransf	FEM	EKF	UKF	OEM	SOEM
Z_w	-1.4249	-1.4249	-1.4249	-1.4249	-1.4277	-1.4294	-1.42780	-1.4252	-1.3086	-1.4268	
Z_q	-1.4768	-1.4768	-1.4768	-1.4768 ^a	-1.4768 ^a	(0.01) ^b	(0.01)	(0.0)	(2.48)	(0.01)	
$Z_{\dot{w}}$	-6.2632	-6.2632	-6.2632	-6.2632	-6.1681	-6.12825	-6.16658	-6.2622	-1.4768 ^a	-1.4768 ^a	
M_w	0.2163	0.2148	0.2148	0.21634	0.21616	(0.01)	(0.01)	(0.01)	-9.9477	-6.1711	
M_q	-3.7067	-3.6906	-3.6907	-3.71215	-3.7086	-3.72645	-3.71967	-3.7373	-3.7343	-3.7201	
$M_{\dot{w}}$	-12.784	-12.7056	-12.7059	-12.7687	-12.773	-12.8262	-12.8152	-12.8247	-12.8122	-12.815	
Eigenvalues	0.6934 -5.8250	0.6887 -5.8041	0.6887 -5.8043	0.6919 -5.8262	0.6900 -5.8262	(0.01) (0.01)	(0.01) (0.01)	(0.02) (0.02)	(1.87) (0.01)	(0.01)	
Program name	uAC_regTLS	uAC_regTLS	uAC_regTLS	ml_oem	ml_oem	ml_oem	ml_fem	mainRPE	mainRPE	ml_oem	ml_oem
Test case	—	—	—	6	7	10	8	8	8	9	9
Function name for state and observation equations	—	—	xdot_TC06_uAC_RegSt	xdot_TC07_uAC_EqDecou	xdot_TC08_uAC_EigT	xdot_TC08_uAC	xdot_TC08_uAC	xdot_TC08_uAC	xdot_TC08_uAC	xdot_TC08_uAC	xdot_TC08_uAC
Nx; Ny; Nu	0; 1; 3 3; 3	0; 1; 3 3; 3	2; 3; 3 6	2; 3; 3 6	2; 3; 3 6	2; 3; 1 6	2; 3; 1 6	2; 3; 1 6	2; 3; 1 6	2; 3; 1 6	2; 3; 1 6
NparSys											

^aParameter held fixed; ^brelative standard deviation in percent.

for the nominal speed U_0 of 44.57 m/s are also provided in the same table. The two modes of the simulated system have the eigenvalues (0.6934, -5.8250), one of them being positive corresponding to the unstable mode, resulting in a time-to-double of roughly 1 s, given by

$$T_2 = \frac{\ln 2}{\sigma_T} \quad (9.50)$$

To this data we now apply the various estimation methods discussed in Secs. III–XI, namely 1) least squares, 2) total least squares, 3) combined least squares and output error, 4) equation decoupling (EqDecpl), 5) eigenvalue transformation (EigTransf), 6) filter error method (FEM), 7) extended Kalman filter, 8) unscented Kalman filter, 9) classical output error method, and 10) artificially stabilized output error method (SOEM). The results are provided in the upper part of Table 9.1, giving the numerical estimates of the derivatives appearing in Eqs. (9.46) and (9.47) and the corresponding eigenvalues. The lower part of the table provides details required to run the respective software programs. These programs are started from different directories as follows:

LS	/FVSysID/chapter09/uAC_regLS	
TLS	/FVSysID/chapter09/uAC_regTLS	
LS/OEM	/FVSysID/chapter04/ml_oem	test_case = 6
EqDecpl	/FVSysID/chapter04/ml_oem	test_case = 7
EigTransf	/FVSysID/chapter04/ml_oem	test_case = 10
FEM	/FVSysID/chapter05/ml_fem	test_case = 8
EKF	/FVSysID/chapter07/mainRPE	test_case = 8
UKF	/FVSysID/chapter07/mainRPE	test_case = 8
OEM	/FVSysID/chapter04/ml_oem	test_case = 8
SOEM	/FVSysID/chapter04/ml_oem	test_case = 9

The data file containing the time histories is found under the directory /FVSysID/flt_data/unStabAC_sim.asc.

The programs uAC_regLS and uAC_regTLS are specially tailored for the specific case being analyzed applying the least squares and total least squares techniques. These programs are fairly straightforward to follow and do not need any further explanation. The least squares and total least squares method do not need starting values to be specified. From Table 9.1 it is observed that the LS and TLS methods yield numerically the same estimates. This is not really surprising, because in the absence of noise in the independent variables TLS is equivalent to LS. These estimates agree very well with the nominal parameter values.

To run the other programs, it is necessary to specify test_case the integer flag for test case number, Nx the number of states, Ny the number of observation variables, Nu the number of input (control) variables, and NparSys the number of system parameters. The total number of parameters to be estimated is given by Nparam = NparSys and in the case of EKF or UKF applying the program "mainRPE" it is Nxp = Nx + Nparam. In each case the model definition is

provided by the user defined function “mDefCase**.m,” where as given earlier ** is set to 6–10 as the case may be. For example, to apply the equation decoupling method described in Sec. VI, we specify the various details as follows:

test_case = 7	% flag for test case
state_eq	% function for state equations
= ‘xdot_TC07_uAC_EqDecoup’;	
obser_eq = ‘obs_TC08_uAC’;	% function for observation equations
Nx = 2;	% number of states
Ny = 3;	% number of observation variables
Nu = 3;	% number of input variables
NparSys = 6;	% number of system parameters
Nparam = NparSys;	% total number of parameters
dt = 0.05;	% sampling time
param	% starting values for parameters
parFlag	% flags for free/fixed parameters

For the other methods a similar procedure is to be followed, each time specifying the details as provided in Table 9.1. In all these cases the same initial guess values for the parameters were used, which were about 50% of the nominal values. In most cases, due to high correlation it was necessary to keep the parameter Z_q fixed, hence the integer flag “parFlag” has to be defined appropriately.

From Table 9.1 it is obvious that the programs FEM, EKF, UKF, OEM, and SOEM use the same functions, namely xdot_TC08_uAC for the state equations and obs_TC08_uAC for the observation equations. Thus, once the postulated model is coded in these two functions, different techniques can be readily applied. The function xdot_TC07_uAC_EqDecoupl called by the equation decoupling method is a modified version of xdot_TC08_uAC using integrated state values for the diagonal terms only and measured states for the off-diagonal terms; see Eq. (9.21). Accordingly, in this function the derivatives Z_q and M_w are multiplied by q_m and w_m , that is, by the respective measured quantities which are provided as additional input variables; except for these two differences, there are no other changes. The function xdot_TC10_uAC_EigT called by the eigenvalue transformation method includes, as elaborated in Sec. VII, the transformation σ_T in the diagonal elements of state matrix, whereas the observation function obs_TC08_uAC in terms of original system parameters is used. In the combined LS/OEM method using regression startup procedure, we make more use of the measured variables by multiplying all the free derivatives with the corresponding measured states, and not just the off-diagonal terms as in the case of the equation decoupling method. Accordingly, we use the function xdot_TC06_uAC_RegSt. In a similar way, except for the regression startup, all other methods call the same function, namely obs_TC08_uAC, to compute the observation variables a_z , w , and q according to Eq. (9.47).

Figure 9.7 shows for two cases, namely the OEM and SOEM, a comparison of data to be matched (continuous lines) and model output (dashed lines) for the specified starting parameter values. A diverging solution resulting from the integration in the classical OEM is observed in Fig. 9.7a (obtained by running ml_oem, for test_case=8 and niter_max=0). On the other hand the artificial

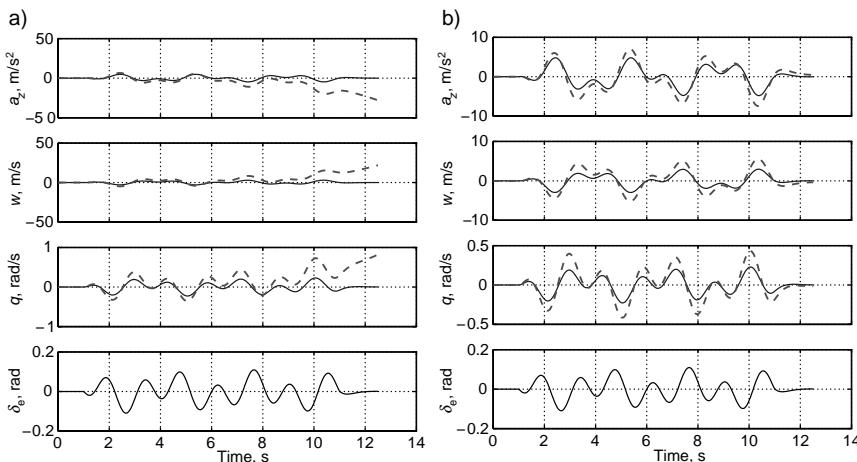


Fig. 9.7 Comparison of system (—) and estimated (----) responses for starting parameter values. a) Output error method, b) Stabilized output error method.

feedback prevents such a divergence; see Fig. 9.7b (obtained by running `ml_oem`, for `test_case=9` and `niter_max=0`). The filter error method provided a similar match to that in Fig. 9.7b. In the case of the other methods too, starting from the specified initial parameter values, the tendency to diverge was not observed. Next, by setting `niter_max` to 10, various cases were run to estimate the parameters. The various methods, except for the classical output error, provided a very good match between measured data and model responses. Since there were no observable differences in the quality of match, a single typical plot is provided in Fig. 9.8.

In the case of the eigenvalue transformation method, we preprocess the output and input variables prior to estimation by multiplying them by $\exp(-0.7^*t)$ which corresponds to the transformation of Eq. (9.26) with $\sigma_T = 0.7$ which is slightly higher than the largest positive eigenvalue. This is done in the model definition function “`mDefCase10.m`.” We note that the vertical acceleration a_z is one of the observation variables. To properly account for this general case, as elaborated in Sec. VII, we adopt the alternative means of programming the state equations by incorporating σ_T in the diagonal elements in the state equations. This was carried out in the function “`xdot_TC10_uAC_eigT.m`” for the parameters Z_w and M_q . In such a case no further back-transformation is necessary, and accordingly, these estimates are provided in Table 9.1. From Fig. 9.9 it can be observed that the exponential function tends to reduce the information content as time increases. The eigenvalues of the transformed system matrix, that is, of $(\tilde{A} = A - \sigma_T I)$, were found to be $(-0.0064; -6.5495)$ implying a stable transformed system.

In the case of the extended Kalman filter (invoked by calling the program “`/FVSysID/chapter07/mainRPE`,” with `test_case = 8`), the measurement noise

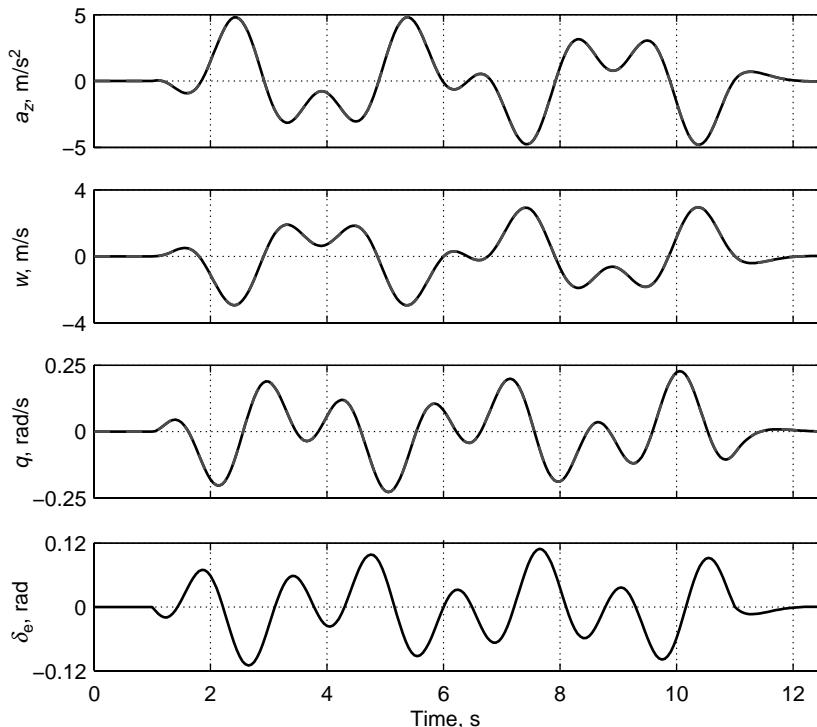


Fig. 9.8 Comparison of system (—) and estimated (----) responses applying a typical estimation method for unstable systems (simulated data).

covariance matrix is specified in "mDefCase08.m" to be diagonal with values of 5×10^{-8} and the process noise covariance matrix is also specified as diagonal with values of 1×10^{-9} for states and zero for the extended states, that is, parameters, according to Eq. (7.49). In the present case, specification of these noise matrices was no problem, because we are analyzing simulated data generated without any process or measurement noise; very small nonzero values were used to avoid numerical inconsistencies. In general, in the presence of measurement and process noise, inaccurate specification of their covariances will affect the estimates. The convergence plot of the estimates is provided in Fig. 9.10; it is observed that convergence is achieved within the first few seconds. An inset shows an enlarged plot of the last few seconds for just two parameters. Table 9.1 gives values which are the averages of the last 10 data points, obtained from an arbitrarily specified initial covariance matrix P . Different starting values for initial P were used; they provided very comparable estimates.

Using the same values for the noise covariances as defined above for the EKF, the UKF is applied by setting the flag method=UKF. This is done in mainRPE. From Table 9.1, it is observed that the estimates from UKF and EKF are comparable.

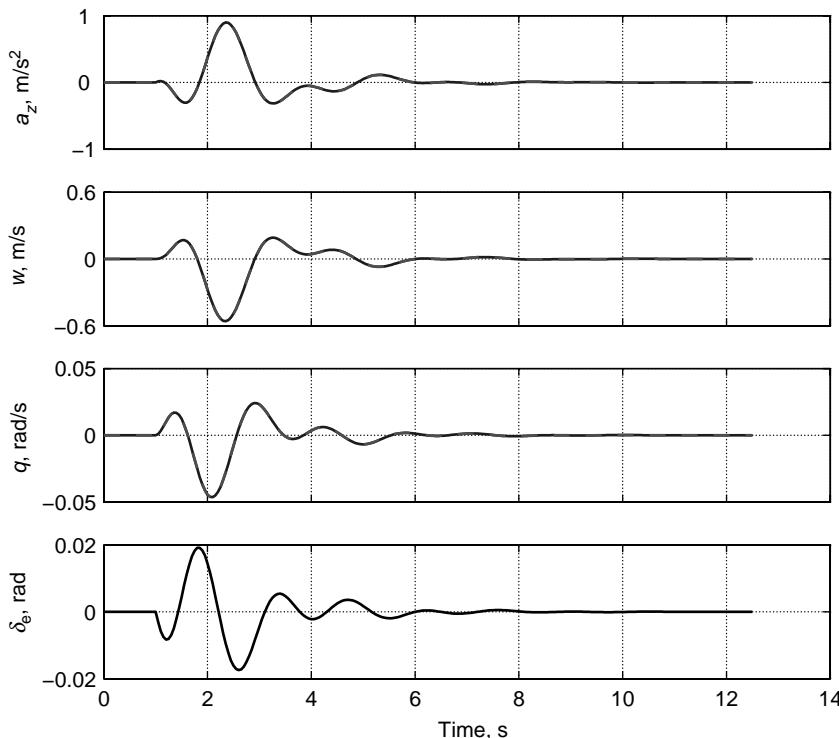


Fig. 9.9 Comparison of system (—) and estimated (----) responses applying the equation decoupling method.

The filter error method is applied by calling the program “/FVSysID/chapter05/ml_fem” with `test_case = 8`. In the present case, elements of the process noise distribution matrix F cannot be estimated, because the simulated data is generated without process and measurement noise. Attempts to estimate F -elements lead to numerical problems. They are kept fixed at very small values of 10^{-6} , practically zero. The results are tabulated in the eighth column of Table 9.1. As expected, no numerical problems were encountered. The estimates match fairly well with the expected values.

The standard output error method, described in Chapter 4, Sec. VI, is applied by calling the program “/FVSysID/chapter04/ml_oem” and setting `test_case = 8`. For several sets of starting values, some even closer to the nominal values, the algorithm faced numerical divergence problems in the very first iteration, making further estimation impossible. For a few sets of starting values the algorithm did converge, but to wrong values. One such case provided the estimated parameters shown in the 10th column. It is observed that the numerical values of all the parameters are completely wrong. The algorithm somehow manipulates the parameter values leading to a convergence and matches the response variables, see Fig. 9.11, but to completely wrong parameter

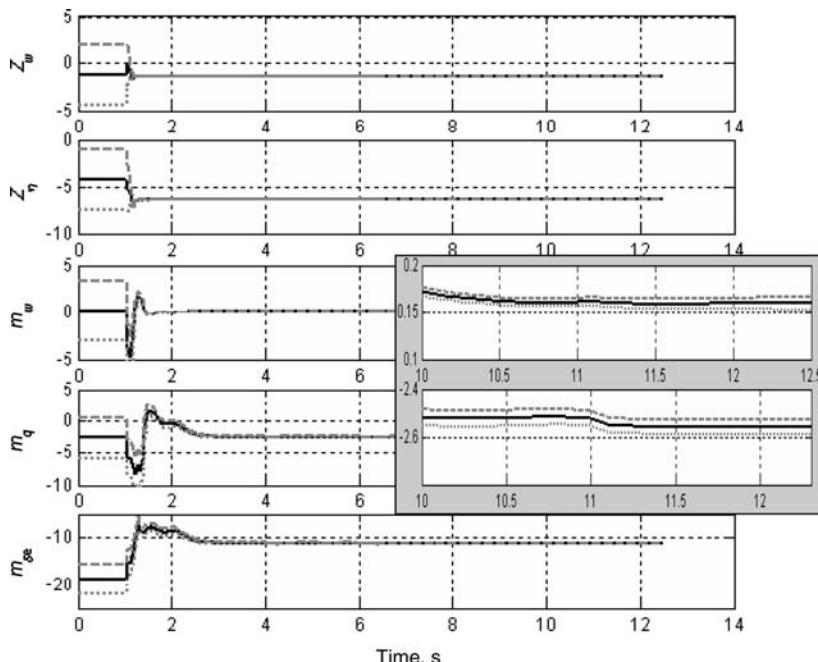


Fig. 9.10 Estimates of parameters applying EKF (—, estimate, -----, estimate \pm standard deviation).

values. The eigenvalues corresponding to the final estimates, as observed from Table 9.1 are -0.0873 and -3.1490 , both of which are not correct. They suggest a stable system, whereas the system being analyzed is unstable. The standard deviations of the estimates provided in the same column range from roughly 2 to 9%, and at first glance may not appear to be too large. However, as the data being analyzed is simulated data with no process and measurement noise, they are very high. All other methods yielded estimates with very very low standard deviations of less than 0.02%. Standard deviation is a measure of the accuracy of the estimates.

The artificially stabilized output error method described in Sec. XI, is applied by calling the same program, namely “ml_oem,” with the flag `test_case=9` and by setting the flag `iArtiStab` and defining the artificial stabilization matrix as follows:

```
iArtifStab = 1; % artificial stabilization
StabMat = [0 0.05 0; 0 0 0]; % stabilization matrix
```

The above values correspond to the stabilization matrix S of Eq. (9.37) given by

$$S = \begin{bmatrix} 0 & 0.05 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (9.51)$$

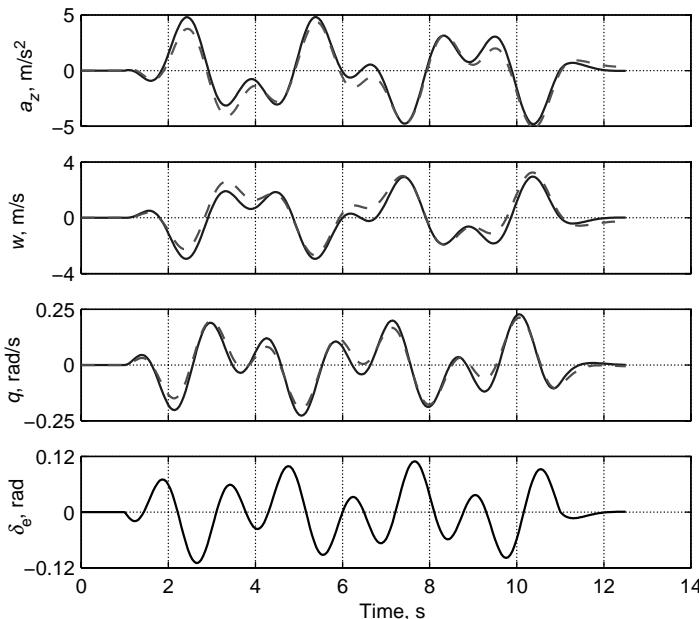


Fig. 9.11 Comparison of system (—) and estimated (----) responses applying a output error method.

for the two states (w and q) and three output variables (a_z , w , q) according to Eqs. (9.46) and (9.47). This implies that we stabilize the integrated state variable w by feeding back the error in the observation variable w . A very low stabilization factor is sufficient to overcome the numerical problems. The same choice of stabilization matrix was found to be adequate for several different sets of starting parameter values. The response match was as shown in Fig. 9.8.

The eigenvalues for the models identified applying different algorithms are computed using the program “/FVSysID/chapter09/eig_uAC_ID.m,” and the results provided as the eighth row of Table 9.1. It is clearly observed that all the methods, except for the classical OEM, yield eigenvalues close to the expected values of the unstable simulated system. The foregoing discussion and results presented in this section corroborate the finer aspects which were brought out while describing the algorithms in Secs. III–XI. It is apparent that these methods use the measured states to varying degrees. Through minor modifications of the function for the state equations, it can be easily verified that the feedback of state variable w is quite sufficient for this example. This is not really surprising, because we know that the instability was introduced in the simulation through the derivative M_w .

B. Identification of X-31A Lateral-directional Motion

X-31A is a highly control-augmented aircraft with enhanced maneuverability. Post-stall maneuvering is enabled, applying advanced technologies such as

high-angle-of-attack aerodynamics and flight control system integrated thrust vectoring. The challenges encountered in the X-31A system identification are typical of unstable aircraft, namely 1) inherently unstable open-loop plant, 2) highly correlated control surface deflections, thrust vectors vanes, and states due to integrated flight control laws, and 3) insufficient and non-optimized excitation. The first and the second aspect have already been dealt with in the various foregoing sections, bringing out the utility and applicability of several estimation algorithms to inherently unstable systems. The improvements through the separate surface excitation were highlighted in Sec. XIV. In this section we further address the last aspect of estimating the parameters from pilot input maneuvers. A bank-to-bank roll around the velocity vector at 54° angle of attack, executed by the pilot through a manual roll command input is considered here.²⁸

The dynamic model describing the lateral-directional motion is formulated as follows without the usual simplifications. It enables to account for larger sideslip angles and possible cross coupling effects. The state equations are given by

$$\begin{aligned} \dot{\beta} &= \frac{g}{V} \left\{ -[(n_x - \sin \theta) \cos \alpha + (n_z + \cos \phi \cos \theta) \sin \alpha] \sin \beta \right. \\ &\quad \left. + \left[\frac{1}{mg} (\bar{q} S C_Y + Y_{\text{Flneng}}) + \sin \phi \cos \theta \right] \cos \beta \right\} + p \sin \alpha - r \cos \alpha \\ I_x \dot{p} - I_{xz} \dot{r} &= \bar{q} S b C_\ell^{\text{CG}} + L_{\text{Flneng}} + I_{xz} pq + (I_y - I_z) qr \\ -I_{xz} \dot{p} + I_z \dot{r} &= \bar{q} S b C_n^{\text{CG}} + N_{\text{Flneng}} - I_{xz} qr + (I_x - I_y) pq \\ \dot{\phi} &= p + (q \sin \phi + r \cos \phi) \tan \theta \\ \dot{\psi} &= (q \sin \phi + r \cos \phi) / \cos \theta \end{aligned} \quad (9.52)$$

Since most of the variables appearing in Eq. (9.52) are according to the standard notation, we do not explain each of them explicitly here. The forces and moments due to the engine, Y_{Flneng} , L_{Flneng} , and N_{Flneng} , were computed using the appropriate measured variables prior to parameter estimation and treated as additional input variables. They include components due to engine inlet and position only, and not those due to the thrust vector deflections. The variables pertaining to the longitudinal motion, V , α , q , θ , n_x , n_z , and \bar{q} , are treated as additional pseudo-control inputs. As discussed in Chapter 3, Sec. VI, the approach of pseudo-control inputs helps to account for the effects of other motion variables which are not treated as state variables.

The coefficients of rolling and yawing moments in Eq. (9.52) are referred to the center of gravity, as implied by the superscript CG. They are given by:

$$\begin{aligned} C_\ell^{\text{CG}} &= C_\ell^{\text{AC}} - C_Y \frac{z_{\text{ACCG}}}{b}; \\ C_n^{\text{CG}} &= C_n^{\text{AC}} + C_Y \frac{x_{\text{ACCG}}}{b} \end{aligned} \quad (9.53)$$

where (x_{ACCG}, z_{ACCG}) denote the position of the aerodynamic center AC with respect to the CG. We consider here the aerodynamic model referred to AC to facilitate comparison of the estimated moment derivatives with the wind-tunnel derived aerodynamic data set (ADS) and also of estimates obtained from different tests performed with different CG locations. Considering linear model postulates, the side force, rolling, and yawing moment coefficients are modeled as

$$\begin{aligned} C_Y &= C_{Y0} + C_{Y\beta}\beta + C_{Yp}\frac{pb}{2V} + C_{Yr}\frac{rb}{2V} + C_{Y\delta_a}\delta_a + C_{Y\delta_r}\delta_r + C_{Y\kappa}\frac{\kappa F}{qS} \\ C_\ell^{AC} &= C_{\ell0} + C_{\ell\beta}\beta + C_{\ell p}\frac{pb}{2V} + C_{\ell r}\frac{rb}{2V} + C_{\ell\delta_a}\delta_a + C_{\ell\delta_r}\delta_r \\ C_n^{AC} &= C_{n0} + C_{n\beta}\beta + C_{np}\frac{pb}{2V} + C_{nr}\frac{rb}{2V} + C_{n\delta_a}\delta_a + C_{n\delta_r}\delta_r + C_{n\kappa}\frac{\kappa F}{qS} \end{aligned} \quad (9.54)$$

It is now our goal to estimate the various parameters C_{Y0} , $C_{\ell0}$, C_{n0} appearing in Eq. (9.54) by matching the observation variables n_y , β , p , r , ϕ , and ψ . Additionally, the angular accelerations, \dot{p} and \dot{r} (not measured directly, but generated by numerical differentiation of the rates) are also matched as observation variables, because inclusion of such angular accelerations is found to speed up the convergence significantly (recall our arguments in Chapter 2, Sec. V on this aspect). Besides the already pointed out pseudo inputs, the primary input variables are the aileron deflection (asymmetric trailing edge flap deflection) δ_a , the rudder deflection δ_r , and κ the thrust deflection angle in yaw.

Attempts to estimate the various parameters appearing in Eq. (9.54) were not successful; estimates were either highly correlated or were not reliable, as implied by large standard deviations. This was expected, because, as evident from the upper four plots of Fig. 9.12a, aileron deflection δ_a and thrust deflection angle in yaw κ are highly correlated. Likewise, the roll rate p is correlated with yaw rate r . Furthermore, the rudder deflection δ_r is zero, because the rudder is faded out by the flight control laws for angles of attack above 40 deg. At such flight conditions the lateral control is possible only through the thrust vector deflection. Based on these inferences, we now reduce the model of Eq. (9.54) to analyze the specific flight maneuver as follows: 1) remove the derivatives due to rudder without affecting the estimation, because rudder deflection is anyway zero; 2) keep thrust-vector effectiveness in yaw $C_{n\kappa}$ fixed at values identified from other flight maneuvers to overcome problems due to correlated δ_a and κ , thereby enabling estimation of $C_{n\delta_a}$; and 3) introduce the following combined derivatives, because correlated roll and yaw do not allow independent estimation of these derivatives

$$\begin{aligned} C_{Yp}^* &= C_{Yp} + C_{Yr}r/p \\ C_{\ell p}^* &= C_{\ell p} + C_{\ell r}r/p \\ C_{nr}^* &= C_{nr} + C_{np}p/r \end{aligned} \quad (9.55)$$

The combined derivatives defined in Eq. (9.55) are, in general, valid only for cases when the roll rate p and yaw rate r are correlated.

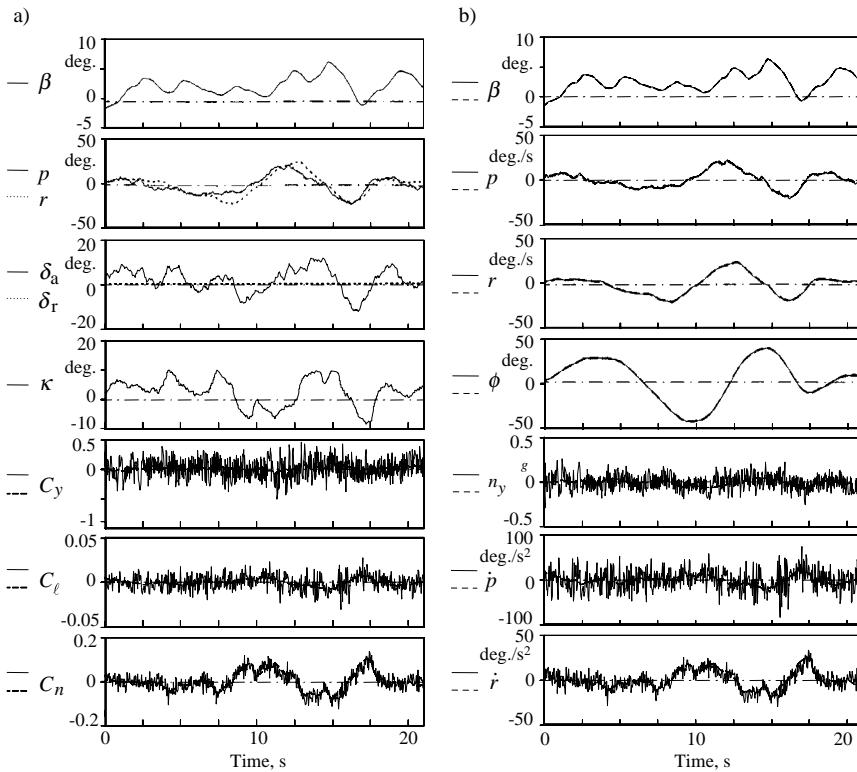


Fig. 9.12 Identification of a bank-to-bank maneuver at 54 deg. angle of attack (—, flight-measured data; -·-, model output). a) Regression method, input and output variables, b) filter error method, output variables.

Thus, a judicious combination of mixed estimation concept and of combined derivatives leads to the following aerodynamic model:

$$\begin{aligned}
 C_Y &= C_{Y0} + C_{Y\beta}\beta + C_{Yp}^* \frac{pb}{2V} + C_{Y\delta_a}\delta_a + C_{Y\kappa} \frac{\kappa F}{qS} \\
 C_\ell^{\text{AC}} &= C_{\ell0} + C_{\ell\beta}\beta + C_{\ell p}^* \frac{pb}{2V} + C_{\ell\delta_a}\delta_a \\
 C_n^{\text{AC}} &= C_{n0} + C_{n\beta}\beta + C_{nr}^* \frac{rb}{2V} + C_{n\delta_a}\delta_a + C_{n\kappa} \frac{\kappa F}{qS}
 \end{aligned} \tag{9.56}$$

Detailed investigations reported in Ref. 28 clearly bring out the limitations of the artificially stabilized output error method. Moreover, as observed from the time history of roll rate p in Fig. 9.12a, it is corrupted by turbulence. The roll rate p has a considerable influence on the sideslip angle β . Although for conventional aircraft the derivative C_{Yp} is not important and often neglected, in the present

case it is necessary to include it. As a consequence, owing to the presence of noise in p , the sideslip angle modeling was also not accurate. This has been confirmed in a separate step by treating the roll rate as an input variable instead of state, which provided a good match for the sideslip angle.²⁸ These considerations indicate the necessity of accounting for process noise in the estimation. Accordingly, we apply three parameter estimation methods to this maneuver, namely 1) least squares, 2) filter error method, and 3) extended Kalman filter.

Filter error and EKF methods can be applied directly to the model postulated in Eqs. (9.52)–(9.56). However, application of the LS method requires computation of aerodynamic forces and moments; the general procedure to compute them from the measured linear accelerations and angular rates has been described in Chapter 6, Sec. IX.A. Figure 9.12 shows the complete bank-to-bank roll evaluated applying a) the least squares method and b) the filter error method.^{28,29} Depending upon the method being applied, different variables are plotted in Fig. 9.12a and b. In the first case, the aerodynamic coefficients C_Y , C_e , and C_n are matched as seen from the three plots on the bottom of Fig 9.12a (shown by continuous and dashed lines). The four plots on the top show measured β , p , r , δ_a , δ_r and κ (continuous and dotted lines), which are input (independent) variables in the least squares method. On the other hand, the filter error method matches the output variables n_y , \dot{p} , \dot{r} , β , p , r , and ϕ . The flight-measured and model estimated time histories for these variables are shown in Fig. 9.12b by continuous and dashed lines. In the case of the filter error method, the fit between flight test data and model output is perfect for all variables. The match in the lateral, roll, and yaw accelerations is of the same quality as the match in the corresponding aerodynamic coefficients obtained from regression.

Figure 9.13 shows the time history plots of the derivatives estimated applying the EKF from the same bank-to-bank maneuver at 54° angle of attack.²⁹ It is

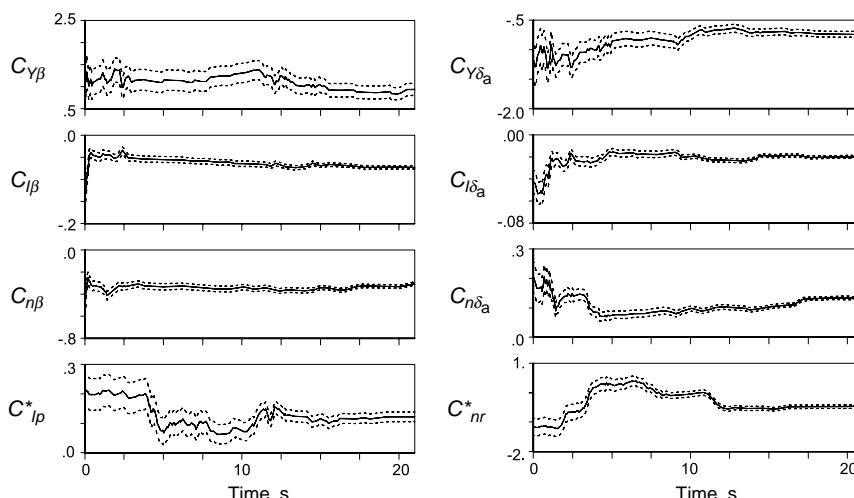


Fig. 9.13 Identification of a bank-to-bank maneuver at 54 deg. angle of attack applying EKF (—, estimate; - - -, estimate \pm standard deviation).

clearly observed that the lateral force derivatives converge poorly. This was to be expected, because as already pointed out the flight control laws suppress the lateral force and as such the data contain practically no information that is necessary to enable estimation of these derivatives.

Having discussed the problems encountered in the estimation of aerodynamic parameters from pilot input maneuvers at high angles of attack, we now turn our attention to study the performance of the least squares and filter error method over a wide range of 30–70 deg. angle of attack in the post-stall regime. Identification results are presented in Fig. 9.14. For comparison, the ADS predictions are also plotted. The sideforce due to sideslip $C_{Y\beta}$ shows large scatter and large uncertainty levels, indicated by the error bars. These estimates are most unreliable. As already pointed out, this is to be expected as there is nearly no excitation in the lateral acceleration. The estimates of dihedral effect $C_{\ell\beta}$ and of aileron effectiveness $C_{\ell\delta_a}$ show some deviations compared with the ADS wind-tunnel predictions. The aileron effectiveness was the best identifiable parameter. The estimates indicate that the decline in aileron effectiveness starts at about 30 deg. angle of attack, which is earlier than predicted at about 40 deg. The combined yaw damping, C_{nr}^* , also has larger scatter due to insufficient excitation of the aircraft eigenmotion resulting from pilot input, because, as pointed out in Sec. I, the controller suppresses the oscillatory motion.

The low quality of the estimation results is attributed to the fact that the aircraft motion is insufficiently excited. Effective excitation of dynamic motion suitable for parameter estimation at high angles of attack from pilot inputs is prevented

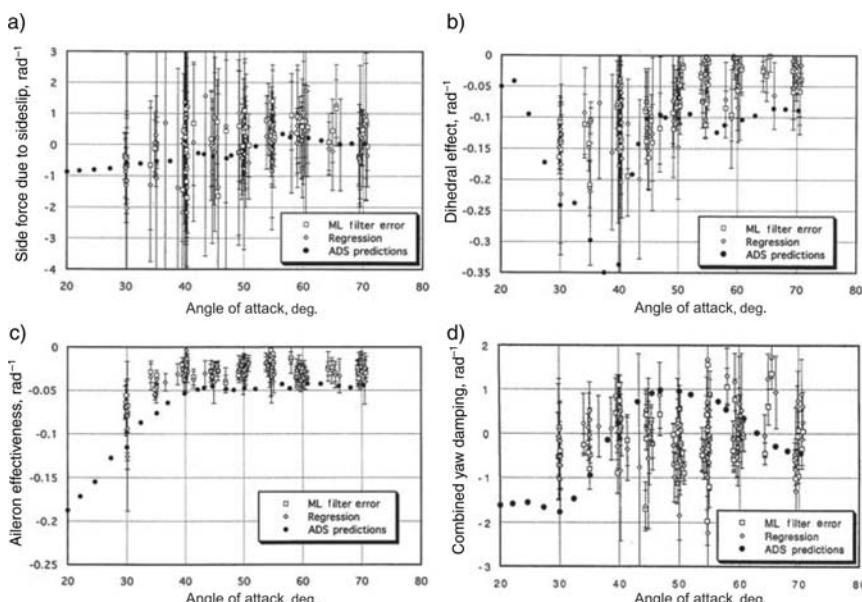


Fig. 9.14 Estimates of aerodynamic derivatives. a) Sideforce due to sideslip, b) dihedral effect, c) aileron effectiveness, d) combined yaw damping.

by the X-31A flight control laws and would be achieved only through separate surface excitation. This has already been substantiated through typical results in Sec. XIV. It is observed that no significant differences can be observed between the results obtained from regression and filter error method, apart from the fact that the uncertainty levels from regression are somewhat larger. This indicates that the simplest and the most sophisticated algorithm produce approximately the same results. This similarity in the two results suggests that the process noise must be dominant compared with the measurement noise. The process noise results from atmospheric turbulence and in the present case from stochastic shedding of vortices, whereas measurement noise results from the sensors. High-quality sensors have fairly low noise levels. It may be recalled that the least squares method accounts for the process noise, but not for the measurement noise in the independent variables, whereas the filter error method accounts for both the process and measurement noise. Some more results pertaining to the X-31A aerodynamic model validation and update will be discussed in Chapter 12, Sec. IX. The parameter estimation techniques described in this chapter have also been applied to other unstable and high angle of attack aircraft, for example the X-29A and F-18 High Angle of Attack Research Vehicle.^{2,31-34}

XVII. Concluding Remarks

In this chapter we have clearly demonstrated that the difficulties encountered in the unstable aircraft parameter estimation in open-loop are related to practical issues of applying system identification, and not to the issues of estimation methodology being applicable or not. This is evident from the fact that we use mostly the same estimation algorithms that we had already covered in the previous chapters, or some suitable variants of them based on the principle of using measured data within the estimation algorithm itself. Several applicable techniques have been elucidated, bringing out advantages and limitations each time. From the sample example analyzing simulated unstable aircraft responses, it is found that all of them serve the intended purpose.

While analyzing flight data, a diligent choice of an algorithm from those presented in this chapter would aid the data analysis. It has been clearly shown that an algorithm accounting for process noise is generally preferable. Such algorithms are 1) least squares method, 2) filter error method, 3) extended Kalman filter, and 4) unscented Kalman filter. Since tuning of the noise matrices in EKF/UKF may be laborious, and a wrong choice might adversely affect the estimates, we prefer the filter error method or the least squares method. It is interesting to note that the simple least squares technique provided estimates comparable to those from the complex filter error method. As a general trend, the least squares approach, combined with data partitioning if necessary, appears to be adequate, provided that the data analyzed is verified for consistency and corrected for systematic errors.

The problems resulting from correlated states and inputs are more difficult to handle. The best choice is separate surface excitation, that is, introducing inputs directly after the controller. The substantial improvements in the accuracy of the estimates obtained from SSE have been clearly demonstrated, but it may be difficult to implement SSE in practice. In the absence of SSE, accurate and reliable estimation of all the unknown derivatives may not be possible. The alternative would be to apply techniques based on detecting data collinearity and subsequent

least squares estimation in terms of combined derivatives or mixed estimation incorporating a priori information. The difficulties encountered in the parameter estimation of pilot input maneuvers have been brought out by analyzing flight maneuvers at high angles of attack in the post-stall regime.

References

- ¹Koehler, R. and Wilhelm, K., "Closed Loop Aspects of Aircraft Identification," AGARD LS-104, Paper 10, Nov. 1979.
- ²Maine, R. E. and Murray, J. E., "Application of Parameter Estimation to Highly Unstable Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 11, No. 3, 1988, pp. 213–219.
- ³Plaetschke, E., "Identifizierung instabiler flugmechanischer Systeme mit einem Ausgangsfehlerverfahren," *ZFW*, Vol. 12, No. 4, 1988, pp. 233–240 (in German).
- ⁴Hamel, P. G. and Jategaonkar, R. V., "Evolution of Flight Vehicle System Identification," *Journal of Aircraft*, Vol. 33, No. 1, 1996, pp. 9–28.
- ⁵Lebacqz, J. V. and Govindaraj, K. S., "Implicit Model Following and Parameter Identification of Unstable Aircraft," *Journal of Guidance and Control*, Vol. 3, No. 2, 1980, pp. 119–123.
- ⁶Jategaonkar, R. V., "Startup Algorithm for Maximum Likelihood Estimation of Parameters in Nonlinear Systems," DFVLR-IB 111-86/67, Dec. 1986.
- ⁷Jategaonkar, R. V. and Thielecke, F., "Evaluation of Parameter Estimation Methods for Unstable Aircraft," *Journal of Aircraft*, Vol. 31, No. 3, 1994, pp. 510–519.
- ⁸Preissler, H. and Schäufele, H., "Equation Decoupling—A New Approach to the Aerodynamic Identification of Unstable Aircraft," *Journal of Aircraft*, Vol. 28, No. 2, 1991, pp. 146–150.
- ⁹Thielecke, F., "Identifizierung instabiler flugmechanischer Systeme mit Hilfe der Mehrzielmethode," DLR-IB 111-91/27, Oct. 1991 (in German).
- ¹⁰Weiss, S., Friehmelt, H., Plaetschke, E., and Rohlf, D., "X-31A System Identification Using Single-Surface Excitation at High Angles of Attack," *Journal of Aircraft*, Vol. 33, No. 3, 1996, pp. 485–490.
- ¹¹Kaletka, J. and Fu, K.-H., "Frequency-domain Identification of Unstable Systems Using X-31A Aircraft Flight Test Data," AIAA Paper 93-3635, Aug. 1993.
- ¹²Tischler, M. B., "Frequency Response Identification of XV-15 Tilt-Rotor Aircraft Dynamics," NASA TM-89428, May 1987.
- ¹³Klein V., "Estimation of Aircraft Aerodynamic Parameters from Flight Data," *Progress in Aerospace Sciences*, Pergamon, Oxford, Vol. 26, No. 1, 1989, pp. 1–77.
- ¹⁴Theil, H., *Principles of Econometrics*, John Wiley & Sons, Hoboken, NJ, 1971.
- ¹⁵Toutenburg, H., *Prior Information in Linear Models*, John Wiley & Sons, New York, 1982.
- ¹⁶Golub, G. H. and Van Loan, C. F., "An Analysis of the Total Least Squares Problem," *SIAM Journal of Numerical Analysis*, Vol. 17, No. 6, 1980, pp. 883–893.
- ¹⁷Laban, M. and Masui, K., "Total Least Squares Estimation of Aerodynamic Model Parameters from Flight Data," *Journal of Aircraft*, Vol. 30, No. 1, 1993, pp. 150–152.
- ¹⁸Naihong, L., Yaohua, W., and Pingyuan, C., "Comment on 'Equation Decoupling—A New Approach to the Aerodynamic Identification of Unstable Aircraft,'" *Journal of Aircraft*, Vol. 30, No. 3, 1993, p. 431.

- ¹⁹Singh, J. and Raisingshani, S. C., "An Equation Decoupling Technique for Identification of Unstable Aircraft with Unsteady Aerodynamic Modelling," *Aeronautical Journal*, Vol. 97, No. 969, Nov. 1993, pp. 321–327.
- ²⁰Maine, R. E. and Iliff, K. W., "Formulation and Implementation of a Practical Algorithm for Parameter Estimation with Process and Measurement Noise," *SIAM Journal of Applied Mathematics*, Vol. 41, No. 3, 1981, pp. 558–579.
- ²¹Jategaonkar, R. V. and Plaetschke, E., "Algorithms for Aircraft Parameter Estimation Accounting for Process and Measurement Noise," *Journal of Aircraft*, Vol. 26, No. 4, 1989, pp. 360–372.
- ²²Balakrishna, S. and Niranjana, T., "Wind Tunnel Dynamic Flying Study of the Pitching Moment Derivatives of the Standard Dynamic Model in Active Control," AIAA Paper 87-2626, Aug. 1987.
- ²³Stoer, J. and Bulirsch, R., *Numerische Mathematik 2*, 3rd ed., Springer, Berlin, 1990.
- ²⁴Bock, H. G., "Numerical Solution of Nonlinear Multipoint Boundary Value Problems with Applications to Optimal Control," *ZAMM*, Vol. 58, 1978, pp. 407–409.
- ²⁵Klein, V., "Aircraft Parameter Estimation in the Frequency Domain," AIAA Paper 78-1344, Aug. 1978.
- ²⁶Fu, K.-H. and Marchand, M., "Helicopter System Identification in the Frequency Domain," *9th European Rotorcraft Forum*, Stresa, Paper 96, Sept. 1983.
- ²⁷Marchand, M. and Fu, K.-H., "Frequency Domain Parameter Estimation of Aeronautical Systems Without and With Time Delays," *Proceedings of the 7th IFAC Symposium on Identification and System Parameter Estimation*, York, 1985, pp. 669–674.
- ²⁸Plaetschke, E., Weiss, S., and Rohlff, D., "Identification at High Angles of Attack Applied to X-31A Flight Test Data," *DLR-Mitt.* 93-14, Dec. 1993, pp. 181–198.
- ²⁹Plaetschke, E., Jategaonkar, R. V., Rohlff, D., and Weiss, S., "Methoden zur Schätzung der Parameter eines Flugzeugs im Post-Stall-Bereich," *Proceedings of the 4th Aerospace Symposium on Safety in Air Traffic*, SFB-212 Technical University of Braunschweig, Germany, 13–15 Sept. 1994, pp. 297–306 (in German).
- ³⁰Plaetschke, E., Mulder, J. A., and Breeman, J. H., "Results of Beaver Aircraft Parameter Identification," DFVLR-FB 83-10, March 1983.
- ³¹Klein, V., Cobleigh, B. R., and Noderer, K. D., "Lateral Aerodynamic Parameters of the X29 Aircraft Estimated From Flight Data at Moderate to High Angles of Attack," *NASA TM 104155*, Dec. 1991.
- ³²Iliff, K. W., "X-29A Lateral-Directional Stability and Control Derivatives Extracted From High-Angle-of-Attack Flight Data," *NASA TP-3664*, Dec. 1996.
- ³³Iliff, K. W. and Wang, K. C., "Flight-determined Subsonic Longitudinal Stability and Control Derivatives of the F-18 High Angle of Attack Research Vehicle (HARV) With Thrust Vectoring," *NASA/TP-97-206539*, Dec. 1997.
- ³⁴Iliff, K. W. and Wang, K. C., "Flight-determined, Subsonic, Lateral-directional Stability and Control Derivatives of the Thrust-Vectoring F-18 High Angle of Attack Research Vehicle (HARV), and Comparison to the Basic F-18 and Predicted Derivatives," *NASA/TP-1999-206573*, Jan. 1999.

This page intentionally left blank

Chapter 10

Data Compatibility Check

I. Introduction

IN GENERAL, parameter estimation methods assume availability of measured data containing adequate information about the cause–effect relationship with a minimum amount of corruption caused by systematic errors like scale factor, zero shift biases, and time lags. The process to generate experimental data with adequate information contents has been dealt with in Chapter 2 on data gathering. In this chapter we deal with the means of checking and improving the quality of the recorded data. The presence of noise is treated differently by the various methods. As discussed in Chapter 6, the least squares estimates are sensitive to such systematic errors and noise in the independent variables. The output error method of Chapter 4 and filter error method of Chapter 5 principally allow estimation of noise statistics and incorporation of corrections for systematic errors as unknown parameters, but it may lead to correlations among the initial conditions or other aerodynamic derivatives, which affects the convergence and accuracy of the estimates. Thus, a data check, independent of system parameter estimation, is usually desirable. Furthermore, in the context of aerodynamic characterization of an aircraft, a large number of variables is usually measured and recorded during flight test programs. Before using the raw flight test data, it is often necessary and time-saving to verify whether the recorded data are compatible or not. The basis for verifying the compatibility of measured data is the use of kinematic relationships. Such a general procedure, which in addition allows for estimation of systematic instrument errors,^{1,2} is commonly called compatibility checking.

The aim of a data compatibility check is to ensure that the measurements used for subsequent aerodynamic model identification are consistent and error free. For example, the measured angle of attack and angle of sideslip must match with those reconstructed from the measured linear accelerations and angular rates. Such a verification is possible in the case of flight data because the well-defined kinematic equations of aircraft motion provide a convenient means to bootstrap the information through a numerical procedure. The primary goals of this data preprocessing step, also termed as flight path reconstruction (FPR), are twofold:

- 1) Determine systematic instrument errors such as scale factors, zero shifts, and time delays in the measurements of flight variables.

2) Generate accurate aircraft states, which basically implies state estimation. The states to be estimated may comprise, for example, the velocity components, and other motion variables.

The FPR is also a necessary first step of the two-step approach using the least squares technique to estimate aerodynamic derivatives,³ and of the similar approach referred to as estimation-before-modeling.^{4–6}

In general, there are two approaches to flight path reconstruction: 1) a rigorous one in the stochastic framework based on the extended Kalman filter, and 2) a simpler one in the framework of deterministic systems based on the output error method, which we found adequate for many practical cases and suitable for more routine use. In the context of methods belonging to the general class of output error, state estimation is a part of the estimation procedure, and hence, during the data preprocessing step that we are currently addressing, it is usually sufficient to focus on the first goal only. Commonly, it aims at calibrating flow variables from flight data, because accurate knowledge about the flow angles, that is, angle of attack and angle of sideslip, is necessary for parameter estimation and for other applications such as in-flight simulation incorporating model following control, gain scheduling, and direct flight control augmentation. These variables, conventionally measured using vanes or five-hole probes, are prone to calibration, bias, and alignment errors and, as a rule, are of lower quality. Laboratory calibration is mostly inadequate, making necessary the determination of calibration characteristics, which can also be configuration-dependent.

In this chapter we adapt the kinematic equations of aircraft motion to include systematic errors, postulate sensor models, and address both the aforementioned techniques of state and parameter estimation. We then deal with the estimation-before-modeling approach. This is followed by discussions on calibrating a five-hole air flow probe, and on determination of the position error in static pressure measurements from dynamic and tower flyby maneuvers. Finally, a more recent approach to calibrating the same variables from a special flight maneuver called wind-box maneuver applying Kalman filter technique and incorporating GPS information is presented. We also demonstrate the procedure of kinematic consistency checking and calibrating flow angles applying the output error method developed in Chapter 4, and describe the details of model definition for the software and flight data provided.

II. Kinematic Equations

The kinematic model for the data compatibility check is derived from the equations of translational motion of an aircraft.^{7,8} For the purpose of simplification, it is assumed that the aircraft is a rigid body. The translational equations of motion in the body fixed axes system with origin at the aircraft center of gravity are given by:

$$\begin{aligned} m(\dot{u} + qw - rv) + mg \sin \theta &= F_X \\ m(\dot{v} + ru - pw) - mg \cos \theta \sin \phi &= F_Y \\ m(\dot{w} + pv - qu) - mg \cos \theta \cos \phi &= F_Z \end{aligned} \quad (10.1)$$

where F_X , F_Y , F_Z are the external (aerodynamic and thrust) forces. Rewriting the above equation in a state variable form, suitable for parameter estimation purposes, yields

$$\begin{aligned}\dot{u} &= -qw + rv - g \sin \theta + a_x \\ \dot{v} &= -ru + pw + g \cos \theta \sin \phi + a_y \\ \dot{w} &= -pv + qu + g \cos \theta \cos \phi + a_z\end{aligned}\quad (10.2)$$

The terms a_x , a_y , and a_z in Eq. (10.2) are the same as F_X/m , F_Y/m , and F_Z/m respectively. They represent the accelerations along the three axes. The angles ϕ and θ denote the roll and pitch angles, respectively.

The additional kinematical equations relating the Euler angles ϕ , θ , ψ to the body-fixed rotational rates p , q , r are given by

$$\begin{aligned}\dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ \dot{\theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= q \sin \phi \sec \theta + r \cos \phi \sec \theta\end{aligned}\quad (10.3)$$

The last set of kinematic relations for the position is given by:

$$\begin{aligned}\dot{x}_g &= u \cos \psi \cos \theta + v(\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi) \\ &\quad + w(\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \\ \dot{y}_g &= u \sin \psi \cos \theta + v(\sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi) \\ &\quad + w(\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \\ \dot{h} &= u \sin \theta - v \cos \theta \sin \phi - w \cos \theta \cos \phi\end{aligned}\quad (10.4)$$

Equations (10.2)–(10.4) represent the complete set of kinematic relationships used for FPR. These nonlinear equations are exact in terms of motion variables (u, v, w) , (ϕ, θ, ψ) , (x_g, y_g, h) , (p, q, r) , and (a_x, a_y, a_z) , because the aerodynamic force and moment coefficients in terms of derivatives do not explicitly appear in these equations. Instead, we use equivalently the linear accelerations and angular rates. The first three triples are the state variables, whose values will be obtained from Eqs. (10.2)–(10.4). Assuming that the measurements of the last two triples are available, and hence known, we have all the necessary information required for state estimation. Initial conditions are either estimated or determined from the measured data, as discussed in Chapter 3, Sec. VII. Thus, through the use of measured linear accelerations (a_x, a_y, a_z) and angular rates (p, q, r) , it becomes possible to estimate the state variables $(u, v, w, \phi, \theta, \psi, x_g, y_g, h)$ using the above-defined kinematic equations.⁹

Once the states of the aircraft motion are known, it is fairly straightforward to derive other variables, such as angle of attack α , angle of sideslip β , true

airspeed V , and dynamic pressure \bar{q} .

$$\begin{aligned} V &= \sqrt{u^2 + v^2 + w^2} \\ \alpha &= \tan^{-1}(w/u) \\ \beta &= \sin^{-1}(v/V) \\ \bar{q} &= \frac{1}{2}\rho V^2 \end{aligned} \tag{10.5}$$

This approach was initially used in dynamic flight testing to eliminate the need for direct measurement of angle of attack.¹ The density of air ρ may be assumed constant, if the variations in altitude during the test maneuver are small and can be obtained from standard atmosphere tables. Otherwise, it can be computed from the actual measurement of the static pressure p_s , using the universal gas law:

$$\rho = \frac{p_s}{RT_s} \tag{10.6}$$

where R is the gas constant and T_s the static temperature. The procedure to calibrate static pressure measurements will be discussed in Sec. VII.

In many cases the measurements of flow angles α and β are available. However, as already mentioned in the preceding section, these variables, measured generally using vanes or a five-hole probe, are prone to calibration, bias and alignment errors, and are inherently of lower quality. Since effects of flight configuration under operating conditions cannot be tested realistically in a laboratory, calibration from flight data is necessary to eliminate the errors. This is possible by appending a sensor model to the variables appearing in Eq. (10.5). Considering a simple sensor model in terms of scale factor, bias, and time delay as elaborated in Chapter 3, Secs. III and IV, any general measured variable, say y , can be expressed as:

$$y_m(t) = K_y y(t - \tau) + \Delta y \tag{10.7}$$

where K_y is the calibration factor, Δy the unknown instrument bias, and τ the time delay in the recorded signal. The subscript m on the left-hand side refers to the measured quantity. Ideally, the scale factor should be unity and the bias and time delay negligible. The instrument errors are usually assumed to be constant over the period of observation, that is, over the length of the time segment being analyzed. Thus, by considering the variables V , α , β , and \bar{q} as the output variables, and incorporating the sensor models similar to Eq. (10.7) in Eq. (10.5), we have a state space model suitable for parameter estimation. Apart from the variables V , α , β , and \bar{q} , the variables ϕ , θ , ψ , and h are also additionally considered as output variables. They are also necessary to enable estimation of the corresponding initial conditions, which may be unknown and, hence, will have to be estimated as well. The sensor model of Eq. (10.7) can be applied to output variables as well as input variables of the state space model.

We also notice that state equations for geodetical positions x_g and y_g are decoupled, that is, these variables do not appear in the other equations in Eqs. (10.2)–(10.4). Hence, they can be omitted without affecting the results for the classical flight path reconstruction purposes pertaining to variables of motion with respect to atmosphere only. Similarly, the ψ equation is also not explicitly coupled into the other equations. For the same reason as before, it would be possible to neglect this variable too, but it is usually retained as it aids estimation of bias in the yaw rate r . The state vector thus reduces to $[u \ v \ w \ \phi \ \theta \ \psi \ h^T]$.

There is one subtle point that is rarely discussed in connection with the above model. As postulated in Eqs. (10.2) and (10.5), it is a mix of inertial and aerodynamic coordinate frames; integration of the inertial accelerations yields incremental changes in an inertial coordinate frame, but they are used to compute the true airspeed in aerodynamic frame. This is valid only for constant wind, which would be accounted for in the initial condition. Under such an assumption, the equations are compatible for the intended purpose of determining instrument errors. An alternative would be to use ground speed. Likewise, in deriving the kinematical relations, apart from the assumption of rigid body, additional assumptions of flat and stationary earth were implicitly incorporated. It is possible to include the effect of flexible aircraft and of a spherical rotating earth. For an extended use of flight path reconstruction, for example verification of the onboard navigation, however, the full set of kinematic equations will be necessary, including a spherical rotating earth. However, such extensions are not considered here, because we are primarily addressing the problems dealing with information necessary for identification of aerodynamic model parameters.

The different motion variables in the kinematic equations (10.2)–(10.5) are with respect to the center of gravity. In Chapter 6, Sec. IX.A we have already covered the practical aspects of linear accelerometers mounted at off-CG locations, and transformations to compute accelerations referred to the CG. The choice of CG as the reference point in Eqs. (10.2)–(10.5) was just for the sake of simplicity; any other arbitrary fixed point can be used as well, but then accounting for the geometrical locations of the various sensors from this chosen reference point, instead of from the CG. We stick to the CG as the reference point, because it is easier to comprehend. Recalling from Chapter 6, Sec. IX.A, the linear accelerations ($a_x^{CG}, a_y^{CG}, a_z^{CG}$) at the CG are computed from the accelerations ($a_{xm}^{AS}, a_{ym}^{AS}, a_{zm}^{AS}$) measured by the acceleration sensor (AS) at a point away from the CG through the following relation:

$$\begin{aligned} a_x^{CG} &= a_{xm}^{AS} + (q^2 + r^2)x_{ASCG} - (pq - \dot{r})y_{ASCG} - (pr + \dot{q})z_{ASCG} - \Delta a_x \\ a_y^{CG} &= a_{ym}^{AS} - (pq + \dot{r})x_{ASCG} + (p^2 + r^2)y_{ASCG} - (qr - \dot{p})z_{ASCG} - \Delta a_y \\ a_z^{CG} &= a_{zm}^{AS} - (pr - \dot{q})x_{ASCG} - (qr + \dot{p})y_{ASCG} + (p^2 + q^2)z_{ASCG} - \Delta a_z \end{aligned} \quad (10.8)$$

where the subscript m stands for the measured quantities and $(x_{ASCG}, y_{ASCG}, z_{ASCG})$ denote the position of the accelerometer with respect to the CG in the body-fixed coordinates; see Fig. 10.1. The exact location of the CG during the flight is determined from the instantaneous fuel quantity and weight and balance sheet. The biases in the measurements of $(a_{xm}^{AS}, a_{ym}^{AS}, a_{zm}^{AS})$ are denoted by

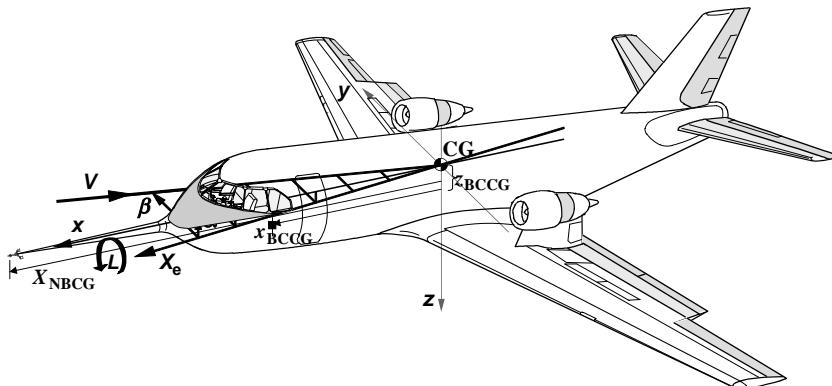


Fig. 10.1 Locations of off-CG sensors.

$(\Delta a_x, \Delta a_y, \Delta a_z)$, and the angular rates (p, q, r) appearing in Eq. (10.2) are given by $(p_m - \Delta p, q_m - \Delta q, r_m - \Delta r)$ obtained from the measured rates (p_m, q_m, r_m) corrected for the biases $(\Delta p, \Delta q, \Delta r)$. The variables $(\dot{p}, \dot{q}, \dot{r})$ required in Eq. (10.8) are obtained by numerical differentiation of the measured angular rates.

Besides the above computations to account for the off-CG location, another type of correction that may have to be performed results from the accelerometer misalignment. If the accelerometers are not aligned exactly along the body-fixed coordinates, it leads to cross coupling between the axes. An error in the orientation causes acceleration along the axis of sensitivity and along the other orthogonal axes as well to be measured. This is schematically depicted in Fig. 10.2 for the alignment error of ε_θ in the pitch axis. It is obvious that the correct accelerations along the body-fixed coordinates will be given by

$$\begin{aligned} a_x^B &= a_{xm}^{AS} \cos \varepsilon_\theta + a_{zm}^{AS} \sin \varepsilon_\theta \\ a_z^B &= -a_{xm}^{AS} \sin \varepsilon_\theta + a_{zm}^{AS} \cos \varepsilon_\theta \end{aligned} \quad (10.9)$$

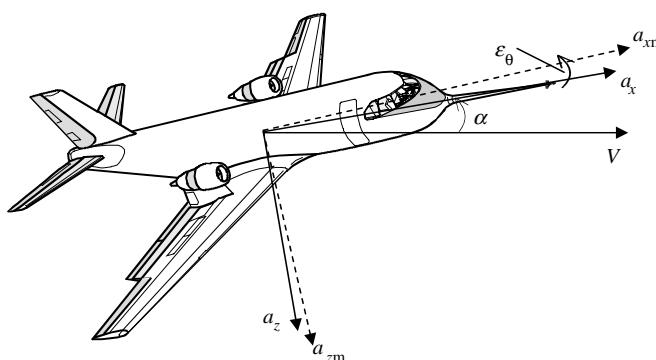


Fig. 10.2 Alignment error in pitch axis.

where the superscript B denote the accelerations along the body-fixed coordinates. Denoting the misalignments with respect to the three body-fixed X, Y, and Z axes as ε_ϕ , ε_θ , and ε_ψ , respectively, and for small angles in radians approximating the sine and cosine function through 0 and 1, accelerations along the body-fixed coordinates can be obtained from the measurements provided by accelerometer triple as

$$\begin{bmatrix} a_x^B \\ a_y^B \\ a_z^B \end{bmatrix} = \begin{bmatrix} 1 & -\varepsilon_\psi & \varepsilon_\theta \\ \varepsilon_\phi \varepsilon_\theta + \varepsilon_\psi & 1 - \varepsilon_\phi \varepsilon_\theta \varepsilon_\psi & -\varepsilon_\phi \\ \varepsilon_\phi \varepsilon_\psi - \varepsilon_\theta & \varepsilon_\theta \varepsilon_\psi + \varepsilon_\phi & 1 \end{bmatrix} \begin{bmatrix} a_{xm}^{AS} \\ a_{ym}^{AS} \\ a_{zm}^{AS} \end{bmatrix} \quad (10.10)$$

The first term on the right-hand side of Eq. (10.10) is the coordinate transformation matrix for misalignments defined with respect to the body-fixed coordinates; see Fig. 10.2. It is usually possible to mount the triaxis accelerometer very precisely; the misalignments are extremely small. Furthermore, in many cases these corrections components are smaller than the measurement noise and vibrations. For these reasons, in practice, we usually do not consider this correction for accelerations. The above principle of correcting for alignment errors can be applied to any sensor. Like accelerometers, the alignments errors in rate gyro are also very small and negligible. If at all, exact alignment of an external long noseboom may pose some difficulties. If such an orientation error is known, it is best to correct it using the above equation. It would not be possible to estimate such alignment errors applying estimation methods, as they are correlated with the constant bias errors included in Eq. (10.7).

Strictly speaking, we should also include the scale factors for linear accelerations and angular rates as well, but they are omitted in Eq. (10.8) for the reasons of identifiability; they tend to become highly correlated with biases. Furthermore, scale factor errors in the inertial measurements from high-precision linear accelerometers and rate gyros are very small and their laboratory calibration more precise. For the same reason the bias errors are also very small, but we include them in the model to avoid drifts in the integrated variables.

Thus, flight path reconstruction amounts to state estimation using kinematic relations (10.2)–(10.5), subject to systematic errors as in Eq. (10.7), and accounting for sensor locations as represented in Eq. (10.8). The complete set of nonlinear system equations is obtained by including, as just discussed, the sensor models in Eqs. (10.2)–(10.4), which leads to the following state equations:^{9,10}

$$\begin{aligned} \dot{u} &= -(q_m - \Delta q)w + (r_m - \Delta r)v - g \sin \theta + a_x^{CG}, & u(t_0) &= u_0 \\ \dot{v} &= -(r_m - \Delta r)u + (p_m - \Delta p)w + g \cos \theta \sin \phi + a_y^{CG}, & v(t_0) &= v_0 \\ \dot{w} &= -(p_m - \Delta p)v + (q_m - \Delta q)u + g \cos \theta \cos \phi + a_z^{CG}, & w(t_0) &= w_0 \end{aligned}$$

$$\begin{aligned}
 \dot{\phi} &= (p_m - \Delta p) + (q_m - \Delta q) \sin \phi \tan \theta + (r_m - \Delta r) \cos \phi \tan \theta, & \phi(t_0) &= \phi_0 \\
 \dot{\theta} &= (q_m - \Delta q) \cos \phi - (r_m - \Delta r) \sin \phi, & \theta(t_0) &= \theta_0 \\
 \dot{\psi} &= (q_m - \Delta q) \sin \phi \sec \theta + (r_m - \Delta r) \cos \phi \sec \theta, & \psi(t_0) &= \psi_0 \\
 \dot{h} &= u \sin \theta - v \cos \theta \sin \phi - w \cos \theta \cos \phi, & h(t_0) &= h_0
 \end{aligned} \tag{10.11}$$

where $(a_x^{CG}, a_y^{CG}, a_z^{CG})$ are defined in Eq. (10.8). The measurement equations are given by:

$$\begin{aligned}
 V_m &= \sqrt{u^2 + v^2 + w^2} \\
 \alpha_{NB,m} &= K_\alpha \tan^{-1} \left(\frac{w_{NB}}{u_{NB}} \right)_\tau + \Delta \alpha_{NB} \\
 \beta_{NB,m} &= K_\beta \sin^{-1} \left(\frac{v_{NB}}{\sqrt{u_{NB}^2 + v_{NB}^2 + w_{NB}^2}} \right)_\tau + \Delta \beta_{NB} \\
 \phi_m &= \phi \\
 \theta_m &= \theta \\
 \psi_m &= \psi \\
 h_m &= h
 \end{aligned} \tag{10.12}$$

The velocity components along the three body-fixed axes at an off-CG location, say at the noseboom (subscript NB), denoted by (u_{NB}, v_{NB}, w_{NB}) , are computed from (u, v, w) as follows:

$$\begin{aligned}
 u_{NB} &= u - (r_m - \Delta r)y_{NBCG} + (q_m - \Delta q)z_{NBCG} \\
 v_{NB} &= v - (p_m - \Delta p)z_{NBCG} + (r_m - \Delta r)x_{NBCG} \\
 w_{NB} &= w - (q_m - \Delta q)x_{NBCG} + (p_m - \Delta p)y_{NBCG}
 \end{aligned} \tag{10.13}$$

where $(x_{NBCG}, y_{NBCG}, z_{NBCG})$ denote the offset distances from the center of gravity to the flow angle sensor mounted on the noseboom.

The quantities appearing on the right-hand side of Eq. (10.12) are either the state variables $(u, v, w, \phi, \theta, \psi, h)$ obtained by integration of the state equations (10.11) or quantities derived (u_{NB}, v_{NB}, w_{NB}) using Eq. (10.13). The parameters K_α and K_β denote the scale factors for the angle of attack and angle of sideslip, and the biases in these variables are denoted $\Delta \alpha_{NB}$ and $\Delta \beta_{NB}$ respectively. If the noseboom is not sufficiently long, these scale factors may be configuration dependent. We can also add bias corrections to other observation variables, but usually it is not possible because initial conditions are estimated.

In Eq. (10.12), the subscript τ implies that the computed angles of attack and sideslip, $\alpha_{NB,C} = \tan^{-1}(w_{NB}/u_{NB})$ and $\beta_{NB,C} = \sin^{-1}(v_{NB}/V_{NB})$, are to be time-shifted through delays τ_α and τ_β , respectively. Using the notation of Eq. (3.14),

the exact representation, for example, for angle of attack would be $\alpha_{NB,m}(t) = \alpha_{NB,C}(t - \tau_\alpha) + \Delta\alpha_{NB}$, but for a notational simplicity we have dropped the time t in the above set of equations. Usually, time delays are found in the measurements of attitude angles ϕ , θ , ψ and also in the flow variables, V , α , β , and \bar{q} . The angular rates may also contain time delays, although not shown explicitly. For the estimation of delays, a time reference is necessary. This is provided by assuming that the linear accelerations have no time delays. This is a reasonable assumption, because they are measured accurately and generally recorded directly. Thus, time delays are usually estimated relative to the time frame of the acceleration measurements.

Independent estimation of scale factor and bias parameters using Eq. (10.12) from a limited set of flight maneuvers needs some consideration, because of the differences in order of magnitude between the trim (nominal) value and bias error. Small variations in scale factor (which is usually around 1) multiplied by the large trim value [for example true airspeed or angle of attack appearing on the right-hand sides of Eq. (10.12)] can compensate for small zero shifts without affecting the match significantly. This leads to a correlation between the two parameters. One possible approach to avoiding such a correlation is to evaluate simultaneously several maneuvers at different trim conditions with good dynamic variations. As an example, Fig. 10.3 shows a combination of maneuvers that provided very good information to estimate the scale factors and biases independently; here we observe dynamic maneuvers at different trim points and quasi-steady maneuvers. The other alternative would be to

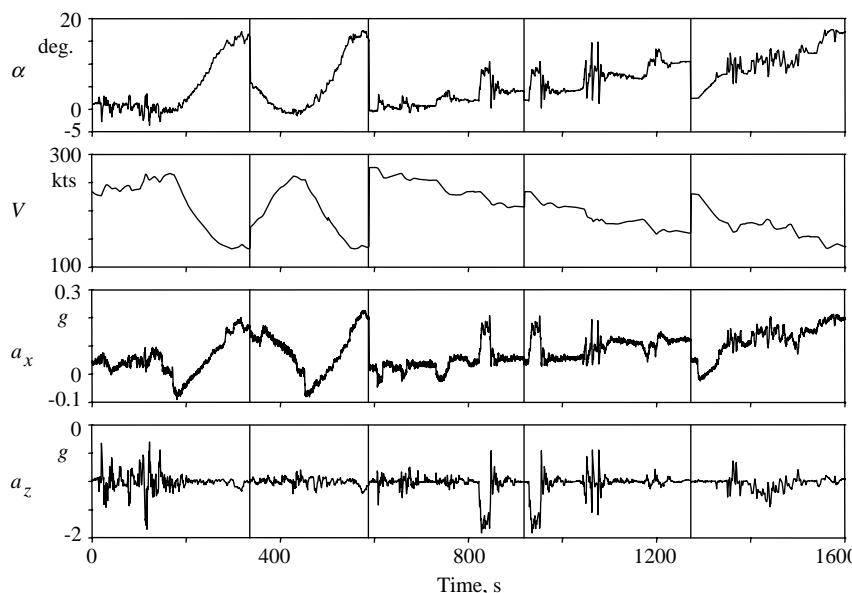


Fig. 10.3 Maneuver combination to avoid scale factor and bias correlation.

reformulate the observation equations to include trim values explicitly and associate the scale factors with the variations about the nominal values. For example, the observation equation for angle of attack can be written as

$$\alpha_m = K_\alpha(\alpha - \alpha_0) + \alpha_0 + \Delta\alpha \quad (10.14)$$

where we have now introduced α_0 to represent the trim value. As elaborated above, the angle of attack α will have to be computed appropriately from integrated states w and u , accounting for position of the sensor. The nominal value of the angle of attack, α_0 , is predefined for each time section being analyzed and held fixed. It can be obtained from the measured data (see Chapter 3, Sec. VII on treatment of initial conditions). A better approach would be to obtain it as $\alpha_0 = \tan^{-1}(w_0/u_0)$ from the initial conditions of states which will be estimated along with the other instrument errors. As a caveat, the estimates of the scale factors in the two cases, namely Eqs. (10.12) and (10.14), will be different. A similar formulation can be applied to other variables. The same approach can be adopted in the estimation of aerodynamic derivatives to overcome the correlation among the various derivatives, particularly that with the aerodynamic zero term.¹¹

Equation (10.12) is the minimum set of observations, provided we perform checking based on the six degrees of freedom equations. Where further variables are to be checked, for example calibrated airspeed, then one can add these as further observation variables, which may need some additional quantities like static air temperature, static pressure, and density of air.

III. Flight Path Reconstruction Techniques

Equations (10.11) and (10.12) are in the general form of a state space model, defined in Chapter 3. Hence, it is evident that we are dealing with a problem that has been addressed a number of times in the preceding chapters. To recapitulate, the problem is to obtain the estimates of the instrument errors Θ and in some cases also of the state vector x subject to the postulated model of Eqs. (10.11)–(10.13). Depending upon the assumptions we make regarding the noise in the input variables ($a_x, a_y, a_z, p, q, r, \dot{p}, \dot{q}$, and \dot{r}), we need either the extended Kalman filter or the output error method to solve the above problem.

A. Deterministic Approach

In many practical applications, particularly when high-precision low-noise accelerometers and rate gyros are used to measure linear accelerations and angular rates, noise in these measurements is very small. Neglecting the noise in these variables, which are input variables in the model for flight path reconstruction, makes the model for kinematic checking deterministic. It is apparent from the parameter estimation methods discussed in Chapters 4 and 5 that deterministic systems are much easier to handle. The state estimation problem reduces to simple numerical integration. Under such an assumption, the model

for estimation of instrument errors is given by

$$\begin{aligned}\dot{x}(t) &= f[x(t), u(t), \Theta], \quad x(t_0) = x_0 \\ y(t) &= g[x(t), \Theta] \\ z(t_k) &= y(t_k) + v(t_k)\end{aligned}\tag{10.15}$$

For such a system model, the state, input and observation vectors for data compatibility are given by

$$\begin{aligned}x &= [u \ v \ w \ \phi \ \theta \ \psi \ h]^T \\ u &= [a_x \ a_y \ a_z \ p \ q \ r]^T \\ y &= [V \ \alpha \ \beta \ \phi \ \theta \ \psi \ h]^T\end{aligned}\tag{10.16}$$

The unknown parameters are

$$\Theta = [\Delta a_x \ \Delta a_y \ \Delta a_z \ \Delta p \ \Delta q \ \Delta r \ K_\alpha \ \Delta \alpha \ K_\beta \ \Delta \beta]^T\tag{10.17}$$

In addition, it may be required to estimate the initial conditions $x_0 = [u_0 \ v_0 \ w_0 \ \phi_0 \ \theta_0 \ \psi_0 \ h_0]^T$.

To this state space model we can readily apply the output error method for nonlinear systems discussed and developed in Chapter 4. The iterative procedure allows estimation of the parameter vector of Eq. (10.17) and also of the covariance matrix of the measurement noise R . Thus, in this approach the problem of data compatibility check is solved as a parameter estimation problem considering a parameter vector consisting of unknown initial conditions and the instrument errors.^{9,12,13} Since the output error method has been elaborately presented and already applied to a few examples, we do not go into further discussion of the algorithm.

B. Stochastic Approach

A rigorous approach to the flight path reconstruction is based on the EKF that accounts for measurement noises in both input and output variables.^{14–20} Introduction of the noise in the input variables makes the system represented in Eq. (10.15) stochastic. In such a case, the model of Eq. (10.15) is modified to

$$\begin{aligned}\dot{x}(t) &= f[x(t), u_m(t) - w(t), \Theta], \quad x(t_0) = x_0 \\ y(t) &= g[x(t), \Theta] \\ z(t_k) &= y(t_k) + v(t_k)\end{aligned}\tag{10.18}$$

where u_m is the measured input vector, $w(t)$ represents the noise in the measured inputs (a_x, a_y, a_z, p, q, r) and $v(t_k)$ that in the observation variables. The noise processes are characterized by Gaussian white noise with zero mean and covariance

matrices Q and R , respectively, and the two noise processes are uncorrelated:

$$\begin{aligned} E\{w(t_k)\} &= 0, & E\{w(t_k)w(t_\ell)^T\} &= Q\delta_{k\ell} \\ E\{v(t_k)\} &= 0, & E\{v(t_k)v(t_\ell)^T\} &= R\delta_{k\ell} \\ E\{w(t_k)v^T(t_\ell)\} &= 0 \end{aligned} \quad (10.19)$$

Before going into the details of an estimation procedure, let us try to correlate a few variables appearing in Eq. (10.18) to those which we have used so far. Hitherto, the measurements of input variables were assumed to be noise-free, hence u_m was simply denoted as u in Eq. (10.15). Now, in the present case, we try to account for the noise in the inputs, hence the term $u_m(t) - w(t)$ instead of $u(t)$ only. Furthermore, the noise $w(t)$ in this context is not the same as the process noise (atmospheric turbulence) considered in Chapter 5, although we have denoted both by the same variable w . In the present case, similar to the measurement noise covariance matrix R of the outputs, we use the covariance matrix Q to characterize the input noise. On the other hand, the atmospheric turbulence was characterized by process noise distribution matrix F , which was assumed unknown and to be identified.

There are different approaches to solving the above general problem of state and parameter estimation. In the simplest form, we prespecify the noise statistics appropriately and apply the extended Kalman filter to the augmented state vector consisting of state variables of the kinematic equations and the various instrument errors, that is, scale factors and bias parameters. A more advanced procedure is based on a variational solution of a nonlinear, fixed-interval smoothing problem, minimizing a quadratic cost function in terms of P_0 , R , and Q , which allows estimation of the initial conditions x_0 and the noise covariance matrix Q as well.²¹ Here, we restrict ourselves to the simpler form based on prespecifying the noise covariances.

As already discussed in Chapter 7 and Appendix F, defining additional auxiliary differential equations $\Theta = 0$ for the unknown parameters, which are assumed constant over the period of observation, and augmenting Θ to x leads to the augmented state vector:

$$\begin{aligned} x_a &= [x^T \ \Theta^T]^T \\ &= [u \ v \ w \ \phi \ \theta \ \psi \ h \ \Delta a_x \ \Delta a_y \ \Delta a_z \ \Delta p \ \Delta q \ \Delta r \ K_\alpha \ \Delta \alpha \ K_\beta \ \Delta \beta]^T \end{aligned} \quad (10.20)$$

The extended Kalman filter for this nonlinear model consists of²²

Prediction step

$$\begin{aligned} \tilde{x}_a(k+1) &= \hat{x}_a(k) + \int_{t_k}^{t_{k+1}} f_a[\hat{x}_a(t), \bar{u}_m(k)] dt \\ \tilde{P}(k+1) &= \Phi(k+1)\hat{P}(k)\Phi^T(k+1) \\ &\quad + \Psi(k+1)B(k)Q(k)B^T(k)\Psi^T(k+1) \end{aligned} \quad (10.21)$$

with initial conditions $\hat{x}_a(0) = x_{a0}$ and $\hat{P}(0) = P_0$.

Correction step

$$\begin{aligned}
 K(k) &= \tilde{P}(k)C^T(k)\{C(k)\tilde{P}(k)C^T(k) + R(k)\}^{-1} \\
 \hat{x}_a(k) &= \tilde{x}_a(k) + K(k)\{z(k) - g_a[\tilde{x}_a(k)]\} \\
 \hat{P}(k) &= \{I - K(k)C(k)\}\tilde{P}(k) \\
 &= \{I - K(k)C(k)\}\tilde{P}(k)\{I - K(k)C(k)\}^T + K(k)R(k)K^T(k)
 \end{aligned} \tag{10.22}$$

for $k = 1, 2, \dots, N$

where \bar{u} is average or interpolated input (see Chapter 3, Sec. VIII.A); the notation of “~” (tilde) and “^” (hat) followed in Chapter 5 has been used in Eqs. (10.21) and (10.22) to denote the predicted and updated states, respectively. In the other commonly used notation, they are denoted $\hat{x}(k|k-1)[=\tilde{x}(k)]$ and $\hat{x}(k|k)[=\hat{x}(k)]$. Likewise, the error covariances are denoted $\hat{P}(k|k-1)[=\tilde{P}(k)]$, $\hat{P}(k|k)[=\hat{P}(k)]$, and $\Phi(k|k-1)[=\Phi(k)]$ and $\Psi(k|k-1)[=\Psi(k)]$. The state transition matrix Φ and its integral Ψ at each discrete point are given by Eqs. (3.38)–(3.39), which are repeated here for convenience:

$$\begin{aligned}
 \Phi(k+1) &= I + A(k)\Delta t + A^2(k)\frac{\Delta t^2}{2!} + \dots \\
 \Psi(k+1) &= I\Delta t + A(k)\frac{\Delta t^2}{2!} + A^2(k)\frac{\Delta t^3}{3!} + \dots
 \end{aligned} \tag{10.23}$$

The Jacobian matrices appearing in Eqs. (10.21)–(10.23) are defined as

$$\begin{aligned}
 A(k) &= \left(\frac{\partial f_a}{\partial x_a}\right)_{\hat{x}_a(k-1), u(k-1)} & B(k) &= \left(\frac{\partial f_a}{\partial u}\right)_{\hat{x}_a(k-1), u(k-1)} \\
 C(k) &= \left(\frac{\partial g_a}{\partial x_a}\right)_{\hat{x}_a(k)}
 \end{aligned} \tag{10.24}$$

These matrices are approximated using the central difference formula discussed in Chapter 3, Sec. V.C. The equation for \tilde{P} is slightly different compared with Eq. (7.53). This is because of the differences we have already discussed at the beginning of this section, namely between input noise and atmospheric turbulence.

The initial values for $\hat{x}_a(t_0)$ and P_0 need to be specified to complete the algorithm. The initial values for the states are derived by averaging over the first few data points. Determination of P_0 may be obtained from the stationary part of the time segment, preceding the maneuver being filtered. An alternative would be to specify it arbitrarily; usually a larger value of the initial state error covariance is necessary to allow tracking of the parameters, whose initial values may differ considerably from the final estimates.

The state estimation applying the extended Kalman filter is often followed by a smoothing technique applied to reduce the inherent bias in the EKF estimates.²³ From the three types of smoothing techniques, namely fixed-interval, fixed-point,

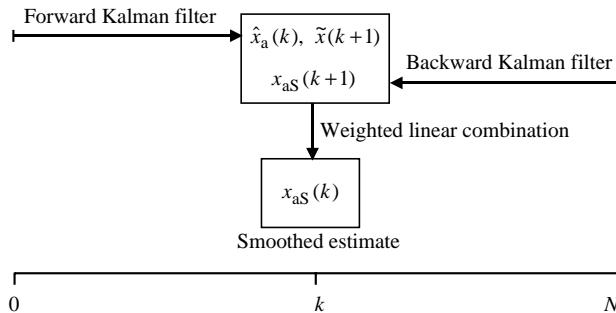


Fig. 10.4 Schematic of filtering and smoothing procedures.

and fixed-lag smoothing,²² the Rauch–Tung–Striebel smoother^{24,25} belonging to the fixed-interval type has been mostly used in the aircraft state estimation, mainly because we perform state estimation for a fixed length of recorded time corresponding to specific dynamic maneuvers. The smoothing procedure is based on using the state estimates from all N data points, and not just up to the k th point as done in the Kalman filter. It leads to a two-pass procedure, the first pass being the regular extended Kalman filter that we have just discussed, called here forward Kalman filter, and the second pass is also a Kalman filter, but running backwards from $N - 1$ to 1. Figure 10.4 depicts the schematic of the combined filtering-smoothing technique. The smoothed estimate is a weighted linear combination of the forward and backward Kalman filter estimates.

Without going into the detailed derivation, we specify here just the final equations for the smoothed state estimates and the error covariance matrix propagation, which can be found in Ref. 24, or any standard book on filtering, for example Ref. 22 or 25:

$$\begin{aligned}
 K_S(k) &= \hat{P}(k)\Phi^T(k+1)\tilde{P}^{-1}(k+1) \\
 x_{aS}(k) &= \hat{x}_a(k) + K_S(k)\{x_{aS}(k+1) - \tilde{x}_a(k+1)\} \\
 P_S(k) &= \hat{P}(k) + K_S(k)\{P_S(k+1) - \tilde{P}(k+1)\}K_S^T(k) \\
 &= \{I - K_S(k)\Phi(k+1)\}\hat{P}(k)\{I - K_S(k)\Phi(k+1)\}^T \\
 &\quad + K_S(k)\{\tilde{P}(k+1) + \Psi(k+1)B(k)Q(k)B^T(k)\Psi^T(k+1)\}K_S^T(k)
 \end{aligned} \tag{10.25}$$

for $k = N - 1, N - 2, \dots$

where the subscript S is added to denote variables pertaining to the smoother, and thereby distinguish between the filter (forward pass) and smoother (backward pass). Accordingly, $x_{aS}(k)$, $K_S(k)$, and $P_S(k)$ denote the smoothed states, smoother gain matrix, and the corresponding error covariance matrix. As in the case of the

forward filter, these variables are also denoted in the literature as $x_a(k|N)$ and $P(k|N)$, where “ $|N$ ” implies reverse filter starting from the $N - 1$ th point.

It is apparent from the discrete time point index k running from $N - 1$ to 1 in Eq. (10.25) that the smoother works backwards in time, and requires variables $\hat{x}_a(k)$, $\tilde{x}_a(k + 1)$, $\hat{P}(k)$, $\tilde{P}(k + 1)$, $\Phi(k + 1)$, and $\Psi(k + 1)$ from the forward filter. As such, their time histories will have to be stored for the complete N time points during the first pass of forward filtering. As in the case of the forward filter (EKF), we have two forms for the covariance matrix of the state prediction error $P_S(k)$ in Eq. (10.25), the second one being numerically more stable for the reasons discussed in Chapter 7 and Appendix F.

Solutions based on the EKF techniques have been proposed in the literature for this combined problem of estimation of states and instrument errors.^{14–20} In the simplified approach presented here, the noise statistics are generally prespecified, which may not be quite accurate, and therefore may affect the estimates. As already pointed out in Chapter 5, Sec. V.B, reasonable information about the R and Q matrices can be obtained from the laboratory calibration of the various measurement sensors. Alternatively, these noise characteristics can also be obtained by applying Fourier smoothing techniques.²⁶ In the context of flight path reconstruction, it is once again pointed out that the input noise is not be confused with the atmospheric turbulence. It implies noise in the measurements of the linear accelerations and angular rates, which are in this case the input variables. If the EKF based approach is to be applied to consistency checking of flight data gathered in turbulent atmospheric condition, it requires not only an extended Kalman filter, but also simultaneous modeling of atmospheric turbulence. Prespecification of noise statistics in such cases is poor and may vary from case to case. Alternatively, as discussed in Chapter 5, the filter error method accounting for both process and measurement noise can be used.

The deterministic approach based on the output error method, presented in Sec. III.A, was suggested as an alternative to the EKF, which had exhibited convergence problems.^{12,13} Comparison of EKF and ML methods for flight path reconstruction indicates that both the techniques yield similar results.^{13,18,19} Furthermore, it is reported in Ref. 18 that for higher noise levels, the ML method appears to perform better compared with the EKF. From several aircraft cases investigated, it has been our experience that the maximum likelihood output error method accounting only for noise in the output variables is adequate for this purpose. It can be used more routinely than the EKF approach, which is more rigorous but is more involved, has some limitations as well and may perform poorly in the absence of accurate a priori knowledge about noise statistics.

As discussed in this section, the kinematic consistency checking provides an *a posteriori* means to verify recorded data and to estimate instrument errors pertaining to the aircraft motion variables. It is an off-line procedure which can be applied during data analysis after flight testing. Besides these motion variables, aerodynamic model identification also needs the measurement of control surface deflections. The measurement of these deflections and other control inputs also should be verified and corrected for instrument and recording errors. Unfortunately, there is no way to check the accuracy of these measurements, because there are no kinematic relations between them and other

variables. This is a fundamental problem, for which there is no easy solution. Hence, calibration of these independent inputs needs great care and considerable effort is spent on laboratory and in situ calibration of these measurements, and on re-verification at regular intervals.

IV. Estimation-before-modeling Approach

The estimation-before-modeling (EBM) methodology is once again a two-step procedure to model aircraft aerodynamics; the first step is generating smoothed time histories of the vehicle states and the second is identifying the aerodynamic model parameters, usually applying the linear regression technique discussed in Chapter 6. Compared with the conventional two-step approach based on the flight-path-reconstruction discussed in Sec. III followed by parameter estimation applying regression technique, the EBM differs primarily in the first step concerned with the state estimation only. In Sec. III, we treated the linear accelerations and angular rates as inputs to the kinematic equations, estimating constant scale factors and bias errors in them, and generating smoothed time histories for the states. The linear accelerations and angular rates (input variables) were not estimated and smoothed. In contrast, the EBM methodology, although conceptually very similar, generates smoothed estimated of the states and estimates the instrument errors, and additionally the optimal, smoothed time histories of the aerodynamic forces and moments. We pursue here the discussion of the first step only, because the second step of identification of aerodynamic model parameters by applying the regression method covered in Chapter 6 is the same in both the cases.

The development of the mathematical model for the state estimation in the present case is very similar to that discussed in Sec. III, the only difference being the inclusion of additional differential equations for the angular rates p , q , and r . Once again, considering the aircraft as a rigid-body, the equation of motion in the body-fixed frame referred to the CG can be formulated as^{4–6}

$$\begin{aligned}\dot{u} &= -qv + rv - g \sin \theta + X \\ \dot{v} &= -ru + pw + g \cos \theta \sin \phi + Y \\ \dot{w} &= qu - pv + g \cos \theta \cos \phi + Z \\ \dot{p} &= pqC_{11} + qrC_{12} + qC_{13} + L + NC_{14} \\ \dot{q} &= prC_{21} + (r^2 - p^2)C_{22} - rC_{23} + M \\ \dot{r} &= pqC_{31} + qrC_{32} + qC_{33} + LC_{34} + N \\ \dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ \dot{\theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= q \sin \phi \sec \theta + r \cos \phi \sec \theta \\ \dot{h} &= u \sin \theta - v \cos \theta \sin \phi - w \cos \theta \cos \phi\end{aligned}\tag{10.26}$$

DATA COMPATIBILITY CHECK

351

where (X, Y, Z) and (L, M, N) are the normalized (that is, divided by mass or appropriate moment of inertia) external aerodynamic forces and moments acting on the aircraft. The constant coefficients C_{ij} appearing in Eq. (10.26) are given by

$$\begin{aligned}
 C_{11} &= \frac{I_{xz}(I_z + I_x - I_y)}{I_x I_z - I_{xz}^2} & C_{21} &= \frac{I_z - I_x}{I_y} \\
 C_{12} &= \frac{I_z(I_y - I_z) - I_{xz}^2}{I_x I_z - I_{xz}^2} & C_{22} &= \frac{I_{xz}}{I_y} \\
 C_{13} &= 0 & C_{23} &= 0 \\
 C_{14} &= \frac{I_{xz}}{I_x} & & \\
 C_{31} &= \frac{I_x(I_x - I_y) + I_{xz}^2}{I_x I_z - I_{xz}^2} & C_{32} &= \frac{I_{xz}(I_y - I_z - I_x)}{I_x I_z - I_{xz}^2} \\
 C_{33} &= 0 & C_{34} &= \frac{I_{xz}}{I_z}
 \end{aligned} \tag{10.27}$$

The terms C_{13} , C_{23} , and C_{33} are zero for the case of neglecting the angular momentum due to engines. If the angular momentum due to rotating parts is considerable and cannot be neglected, then these three terms will have to be appropriately considered.⁴

Equation (10.26) can be reformulated in the general state variable form:

$$\dot{x}(t) = f[x(t), \chi(t)] + w(t) \tag{10.28}$$

where $w(t)$ is, as in the previous case, the Gaussian noise vector with zero mean and covariance matrix Q ; the state vector x and the vector χ consisting of external forces and moments are given by

$$\begin{aligned}
 x &= [u \ v \ w \ p \ q \ r \ \phi \ \theta \ \psi \ h]^T \\
 \chi &= [X \ Y \ Z \ L \ M \ N]^T
 \end{aligned} \tag{10.29}$$

In the previous approach discussed in Sec. III, we used for χ equivalently the measured accelerations and angular accelerations as inputs. Instead of treating them as inputs, in the present case we treat them as pseudo-state variables by modeling them stochastically. This is done for each component of χ by a third-order Gauss–Markov process, which can be represented as⁶

$$\dot{\chi}_i = L_i \chi_i + \xi_i(t); \quad i = 1, 2, \dots, 6 \tag{10.30}$$

where

$$L_i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (10.31)$$

and ξ is the white zero-mean Gaussian noise, generated using the random number generator. Equations (10.30) and (10.31) imply that for each component of χ we have three first-order differential equations. For example, the pitching moment M will consist of ($\dot{M} = M_1$; $\dot{M}_1 = M_2$; $\dot{M}_2 = 0$), where (M , M_1 , M_2) are the three pseudo-states. Without going into the statistical details of the Gauss–Markov process, we simply note here that it is used to model the external forces and moments. Equation (10.31) yields a quadratic interpolation polynomial as a function of time; the recursive algorithm updates the polynomial coefficients at each sample. Thus, a total of 18 additional states are introduced, corresponding to three for each of the six forces and moments.

Finally, as in the previous case, assuming the instrument errors, represented by the combined vector Θ , to be constant over the period of observation, they are modeled by the auxiliary dynamic system:

$$\dot{\Theta} = 0 \quad (10.32)$$

Combining equations (10.28), (10.30), and (10.32) leads to the augmented system

$$\begin{bmatrix} \dot{x} \\ \dot{\chi} \\ \dot{\Theta} \end{bmatrix} = \begin{bmatrix} f[x(t), \chi(t)] \\ L\chi(t) \\ 0 \end{bmatrix} + \begin{bmatrix} w \\ \xi \\ 0 \end{bmatrix} \quad (10.33)$$

consisting of 10 basic states of aircraft motion variables, 18 pseudo-states for the six aerodynamic forces and moments (each modeled as a third-order Gauss–Markov process), and auxiliary states whose number will depend upon the number of instrument errors to be estimated. The matrix L is block-diagonal consisting of six blocks, each of the form of Eq. (10.31).

To complete the model postulate for state and parameter estimation, we define the observation equations

$$\begin{aligned} y(t) &= g[x(t)] \\ z(t_k) &= y(t_k) + v(t_k) \end{aligned} \quad (10.34)$$

where the observation vector y consists of the following variables:

$$y = [V \ \alpha \ \beta \ \phi \ \theta \ \psi \ h \ p \ q \ r \ a_x \ a_y \ a_z]^T \quad (10.35)$$

The observation variables appearing in Eq. (10.35) are standard variables; their exact forms have already been defined earlier in the chapter; they can also be found in Refs. 6 and 8. We also specifically note that, since (a_x, a_y, a_z, p, q, r)

are now treated as output variables, biases in their measurements appear in the observation equations, and not in the state equation formulated in Eq. (10.26). This is in contrast to Eq. (10.11) of the foregoing approach elaborated in Sec. III. The vector of unknown parameters corresponding to the instrument errors is once again given by

$$\Theta = [\Delta a_x \Delta a_y \Delta a_z \Delta p \Delta q \Delta r K_\alpha \Delta \alpha K_\beta \Delta \beta]^T \quad (10.36)$$

Additional corrections may have to be introduced, but the choice of corrections given in Eq. (10.36) has proved adequate in most of the cases analyzed. Introducing additional corrections in the observation equations of Eq. (10.35) and appending the parameter vector of Eq. (10.36) and the auxiliary state equation of Eq. (10.32) is a trivial exercise. As with any parameter estimation problem, care needs to be exercised when adding new parameters to be identified, because all of them may not be identifiable; they may be correlated or there may not be adequate information in the data set being analyzed.

Now, we can apply the extended Kalman filter and smoother, Eqs. (10.21)–(10.25), discussed in Sec. III.B to the system defined by Eq. (10.33)–(10.36). This provides the optimal smoothed estimates of the basic states x , of the normalized aerodynamic forces and moments (X, Y, Z, L, M, N), which are elements of χ , and of the instrument errors Θ .

Augmentation of the basic aircraft states through 18 additional pseudo-states to model the aerodynamic forces and moments makes the model computationally intensive and time-consuming compared with the approach presented in the preceding section. Considerable saving in the computational time is possible by making optimum use of the block-diagonal structure of the L matrix appearing in Eq. (10.33), and also of the null matrix corresponding to the auxiliary state equations $\Theta = 0$, see the schematic of the system matrix of Eq. (10.33) shown in Fig. 10.5. The nonzero elements appear in blocks shown in gray with bold borders. The special matrix structure shown in the figure can be used advantageously to reduce the number of numerical computations while computing the state transition matrix Φ , its integral, and the Jacobian matrices defined in Eq. (10.24).

EBM methodology provides smoothed time histories of the external normalized forces and moments, which can be analyzed without any further processing in the second step to estimate aerodynamic derivatives. Thus, numerical differentiation of the angular rates is eliminated, which is otherwise necessary for the two-step approach based on the procedure of Sec. III. As elaborated in Chapter 6, Sec. IX.A, computation of aerodynamic forces and moments requires angular accelerations, which are not directly measured, but are generated through numerical differentiation. Such procedures may be inaccurate in the presence of noise, but the advantages of EBM from this aspect may be just marginal, because generally the noise in the angular rates is low and good digital symmetrical filters are available to smooth the data before numerical differentiation. Furthermore, approximation of the forces and moments using a third-order process is a compromise between total model order and anticipated time-rate-of-change in forces and moments.⁵

Basic states (9)							
		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
1, 2, ..., 9 Basic states	3 × (1 2,3,4,5 6)	Pseudo-states of Gauss–Markov processes	1 2,3,..., N_p	Auxiliary equations for instrument errors			

Fig. 10.5 Block structure of system matrix for state estimation using EBM.

V. Example

To demonstrate the procedure of data compatibility checking, we consider here a simple case of checking data recorded during typical flight tests carried out with the research aircraft ATTAS.¹⁰ Three dynamic maneuvers are analyzed, an elevator input maneuver exciting the short period motion, an aileron input maneuver resulting in bank-to-bank motion, and a rudder input maneuver leading to Dutch roll motion. The first maneuver is 25 s long and the other two are of 30 s duration each. These maneuvers were executed starting from the same nominal trim condition and performed manually by the pilot. The flight maneuvers were carried out under seemingly steady atmospheric conditions, when the turbulence was considered negligible. The data was recorded with a sampling time of 40 ms. To these three maneuvers, which were used previously in Chapter 6, we apply the deterministic approach presented in Sec. III.A, using the output error method and the corresponding estimation program “ml_oem” elaborated in Chapter 4.

The state, input, and observation variables for the kinematic model consisting of state and observation equations, Eqs. (10.11)–(10.12), and the corresponding functions to code the right hand sides are defined as

	No. of variables	Function name
States	7 $u, v, w, \phi, \theta, \psi, h$	“xdot_TC22_fpr”
Outputs	7 $V, \alpha_{NB}, \beta_{NB}, \phi, \theta, \psi, h$	“obs_TC22_fpr”
Inputs	9 $a_x, a_y, a_z, p, q, r, \dot{p}, \dot{q}, \dot{r}$	

To start with, we assume that the angle of attack and angle of sideslip have been correctly calibrated, and require no corrections. Accordingly, we introduce bias

DATA COMPATIBILITY CHECK

355

corrections for the input variables only. For such a case, the unknown parameter vector Θ consists of

$$\Theta = [\Delta a_x \ \Delta a_y \ \Delta a_z \ \Delta p \ \Delta q \ \Delta r]^T$$

Designating this test case as 22, the model definition provided by the function /FVSSysID/chapter10/mDefCase22.m, which is called from the main program “/FVSSysID/chapter04/ml_oem,” is as follows:

```
test_case = 22; % test case number
state_eq = 'xdot_TC22_fpr'; % function for state equations
obser_eq = 'obs_TC22_fpr'; % function for observation equations
Nx = 7; % number of states
Nout = 7; % number of observation variables
Ninp = 9; % number of input (control) variables
NparSys = 6; % number of system parameters
Nparam = NparSys; % total number of parameters
dt = 0.04; % sampling time
```

The starting values for the unknown parameters, that is, biases, are zeros and the integer flags parFlag are set to 1, indicating that all parameters are to be estimated.

The data to be analyzed, the arrays Z(Ndata,Nout) and Uinp(Ndata,Ninp), are loaded from three data files:

```
load ..\flt_data\fAttasElv1;
load ..\flt_data\fAttasAil1;
load ..\flt_data\fAttasRud1;
```

Since we are analyzing three separate maneuvers simultaneously, we additionally define

```
Nzi = 3; % number of time segments
izhf = [Nts1; Nts1 + Nts2; Nts1 % cumulative index
+ Nts2 + Nts3);
```

where Nts1, Nts2, and Nts3 are the number of data points in the three individual time segments and izhf is the cumulative index at which the maneuvers end when they are concatenated.

As in the test cases of the previous chapters, it is necessary to provide the input and output variables in proper units. This has been taken care of in the model definition function “mDefCase22.m,” immediately after loading the data files and while assigning the appropriate channels to the arrays Z and Uinp. The angular accelerations are generated by numerical differentiation of the measured angular rates using a first-order differentiator with eighth order filter

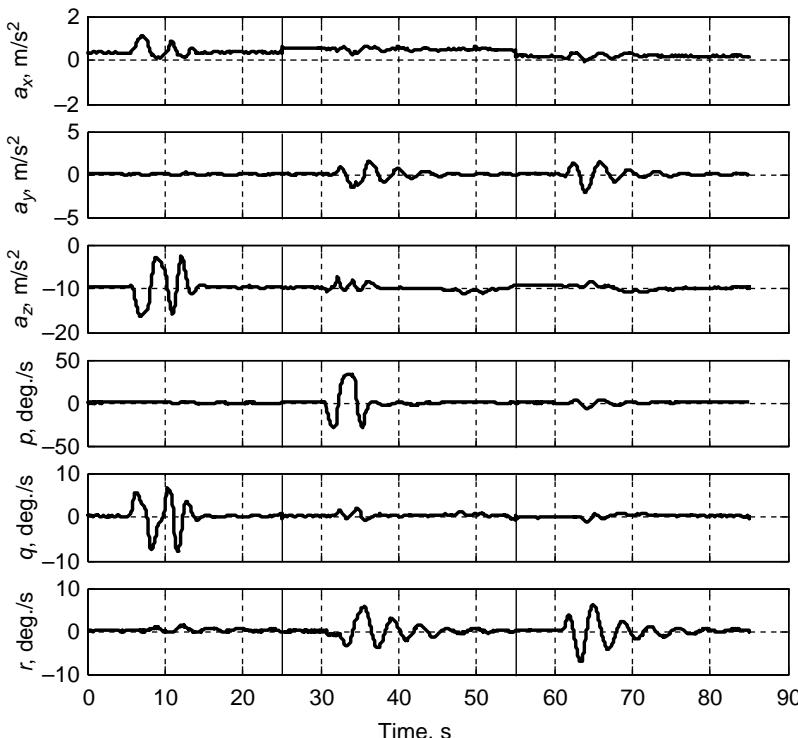


Fig. 10.6 Time histories of input variables for data compatibility check.

"ndiff_Filter08.m," see Chapter 2, Sec. V.B, called from the function "ndiff_pqr.m." The initial conditions (u_0, v_0, w_0) for the three time segments are set to $(129.4; -0.9; 5.05)$, $(129.4; -0.9; 5.15)$ and $(128.7; -0.9; 4.75)$ m/s, and those for the other states, namely (ϕ, θ, ψ, h) to the measured values of the respective first data point. The exact procedure can be easily traced in "mDefCase22m." Setting the starting values for biases Δa_x , Δa_y , Δa_z , Δp , Δq , and Δr to zero, and the counter for the maximum number of iterations niter_max to zero, a simulation is performed by running the program "ml_oem" from the directory /FVSysID/chapter04/. The time histories of the input variables, in the present case those of linear accelerations and angular rates, are shown in Fig. 10.6. Time histories for the output variables V , α_{NB} , β_{NB} , ϕ , θ , ψ , and h are plotted in Fig. 10.7, showing flight measured data in continuous lines, and those simulated using the kinematic equations for the specified initial conditions and without any corrections of the instrument errors in dashed lines. It is apparent that the specified initial conditions are well chosen, but the simulation shows drifts in all the variables which are attributed to biases in the input variables. Here, we recall our suggestion made in Sec. II that biases in the input variables always need to be considered to avoid drifts in the integrated variables, even when such biases may be very small.

DATA COMPATIBILITY CHECK

357

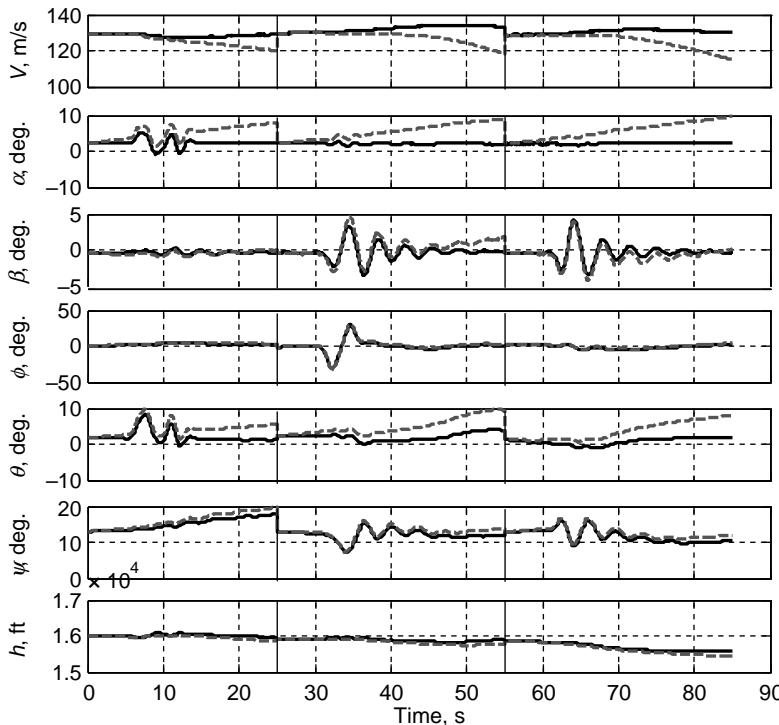


Fig. 10.7 Measured and simulated responses without instrument corrections (—, flight measured; - - -, reconstructed).

Now, setting the counter `niter_max` to a higher value, say 10, we restart the estimation program. It is observed that the estimation procedure starting with the initial cost function, $\det(R)$, value of 1.5138×10^{-12} converges to the minimum of 1.8792×10^{-26} within four iterations. The bias parameters are estimated to be

$$\begin{aligned}\Delta a_x &= 0.0092 \text{ m/s}^2 & \Delta a_y &= 0.00021 \text{ m/s}^2 & \Delta a_z &= 0.0960 \text{ m/s}^2 \\ \Delta p &= 0.0011 \text{ rad/s} & \Delta q &= 0.0033 \text{ rad/s} & \Delta r &= 0.0010 \text{ rad/s}\end{aligned}$$

None of the estimates were correlated (> 0.9), and their standard deviations were small, suggesting reliable values. The large relative standard deviation associated with Δa_y is because the bias in this measurement is practically zero. Several repeated checks at different flight conditions and from different maneuvers provided very similar estimates for these bias parameters. The resulting match shown in Fig. 10.8 between the flight measured and reconstructed responses for the variables V , α_{NB} , β_{NB} , ϕ , θ , ψ , and h is acceptable. An enlarged plot of

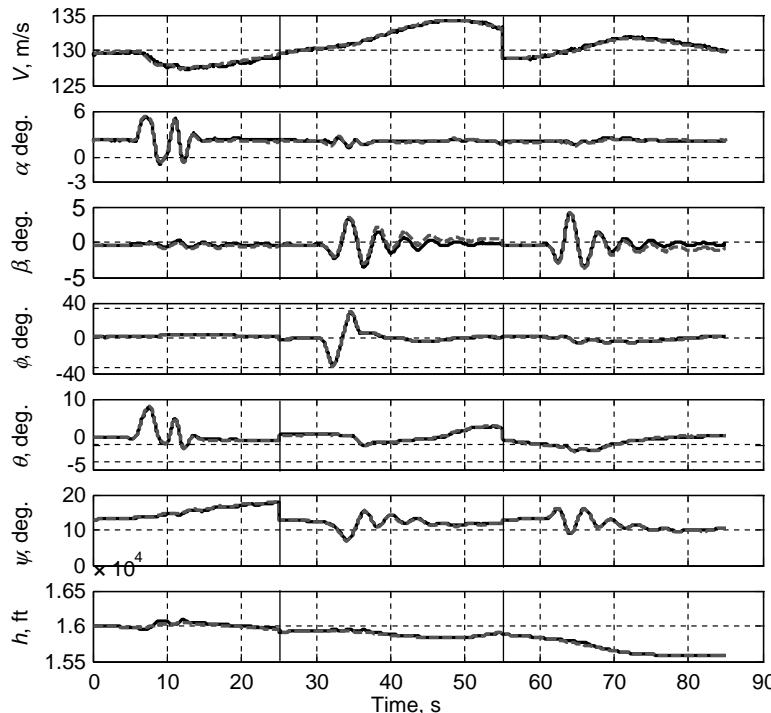


Fig. 10.8 Measured and simulated responses with bias corrections (—, flight measured; - - -, reconstructed).

the angle-of-sideslip (not shown here) shows some drift, which is attributed to two factors: 1) a small mismatch in the initial values which were only roughly specified and held fixed, and 2) estimating a single set of biases for the three flight maneuvers. Further experimentation with data checking by introducing additional parameters is left to the reader. The reconstructed flow angles matched the recorded data without any scale factor or bias corrections fairly well; this is attributed to the fact that the onboard recording system was calibrated and updated, incorporating such corrections which were already determined. Otherwise, usually scale factors and bias corrections are necessary for these measurements.

VI. Calibration of Five-hole Flow Angle Probe

Having studied different approaches to determine instrument errors in Secs. II and III and demonstrated the procedure on a simplified example in Sec. V, we now turn our attention to more detailed practical issues related to calibrating the flow angle sensors. This is one of the prime applications of the data compatibility check in general, and, as pointed out in Sec. I, particularly for identification of aerodynamic model parameters using the output error method.

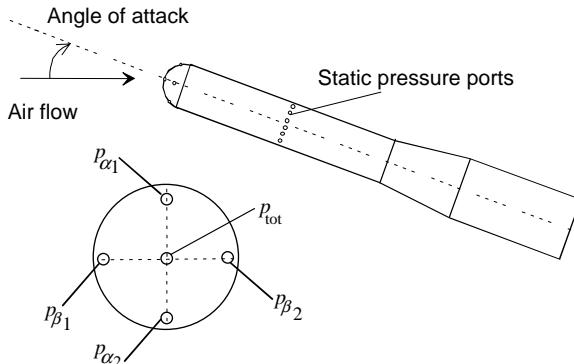


Fig. 10.9 Five hole probe-port configuration.

Different types of sensors, such as vanes, five-hole probe, or multihole system in the form of a flush air data sensor, are used to measure the flow angles. The procedures to calibrate a vane or a five-hole probe are quite similar except for the observation model. We address here the case of calibrating a five-hole probe from flight test data applying the data compatibility check.^{27,28}

A five-hole probe, Fig. 10.9, provides measurements of total pressure, p_{tot} , differential pressures for angle of attack, $p_{\alpha 1} - p_{\alpha 2}(= p_{d\alpha})$, and for angle of sideslip, $p_{\beta 1} - p_{\beta 2}(= p_{d\beta})$. Static pressure, p_s , is also measured very close to the tip of the noseboom. The angles of attack and sideslip are then given by^{29,30}

$$\alpha = \frac{p_{\alpha 1} - p_{\alpha 2}}{K p_{\text{dyn}}}; \quad \beta = \frac{p_{\beta 1} - p_{\beta 2}}{K p_{\text{dyn}}} \quad (10.37)$$

where $p_{\text{dyn}}(= p_{\text{tot}} - p_s)$ is the dynamic pressure and $K = f(Ma)$ is the sensitivity coefficient specified by the manufacturer, that is to be flight verified. Other corrections have to be determined, if necessary. In the present case we verify the raw data for the recorded $p_{d\alpha}$ and $p_{d\beta}$ directly without any onboard corrections or preprocessing. This is possible by reformulating Eq. (10.37) as follows:

$$\begin{aligned} p_{d\alpha m} &= K_\alpha p_{\text{dyn}} \alpha_{NB} + \Delta p_{d\alpha} \\ p_{d\beta m} &= K_\beta p_{\text{dyn}} \beta_{NB} + \Delta p_{d\beta} \\ p_{\text{dynm}} &= \frac{1}{2} \rho \sqrt{u_{NB}^2 + v_{NB}^2 + w_{NB}^2} \end{aligned} \quad (10.38)$$

where, as before, m denotes the measured quantities, K_α and K_β are the unknown sensitivity coefficients per degree, and Δs are the bias corrections in the respective measurements. The flow variables at the noseboom, namely the angle of attack α_{NB} , angle of sideslip, β_{NB} , and the dynamic pressure p_{dyn} , are obtained from the local velocity components at the noseboom, computed from the reconstructed states by accounting for the sensor location away from the center

of gravity. The reconstructed states are the states provided by the kinematic equations, in the present case by integration because we once again apply the deterministic approach, since the flight tests were carried out in steady atmospheric conditions, where the turbulence level is negligible.

To account for possible time delays resulting from the recording system, the quantities on the right-hand side of Eq. (10.38) are appropriately modified. For example, in the case of angle of attack, the exact representation will be

$$\begin{aligned} p_{d\alpha,C} &= K_\alpha p_{dyn} \alpha_{NB} \\ p_{d\alpha,m}(t) &= p_{d\alpha,C}(t - \tau_{pd\alpha}) + \Delta p_{d\alpha} \end{aligned} \quad (10.39)$$

where $\tau_{pd\alpha}$ is the time delay in the difference pressure $p_{d\alpha}$. The time delays $\tau_{pd\beta}$ and τ_{pdyn} were also introduced for the difference pressure $p_{d\beta}$ and dynamic pressure through similar equations. The time delays are in general not known, and, hence, have to be estimated along with the unknown scale factors and biases. Thus, the kinematic model consists of state equations (10.11) and observation equations (10.12) in which V , α , and β are replaced through $p_{d\alpha}$, $p_{d\beta}$, and p_{dyn} as modeled in Eq. (10.38). Such a model is applied to calibrate five-hole probes mounted on a noseboom on two aircraft: 1) VFW-614 ATTAS (Fig. 10.10a), and 2) Transall C-160 (Fig. 10.10b). The results of parameter estimation are provided in Figs. 10.11 and 10.12.^{27,28}

For VFW-614 ATTAS, Fig. 10.11 clearly shows that the flight estimates of sensitivity coefficients match reasonably well the manufacturer's data.^{29,30} To summarize, the estimated scale factors for the angle of attack and angle of sideslip are 0.0819 for Mach variation up with 0.55 compared with 0.079 specified by the manufacturer. For Mach 0.6 the scale factor for angle of attack is estimated to be 0.0781, and no Mach dependency could be estimated for angle of sideslip. The minor differences compared with the manufacturer's data are attributed to position error, possible errors in measured dynamic pressure, and noseboom length limited to roughly 1.35 times the fuselage diameter, which is probably not sufficient to reach free stream flow. An empirical rule suggests a boom length of 2.5–3 times the fuselage diameter. The sensitivity coefficients and

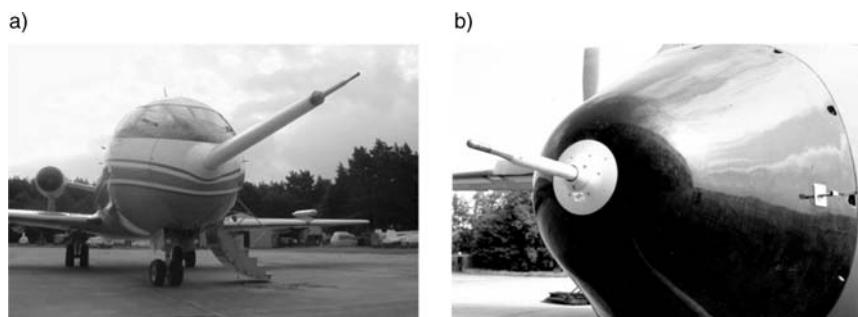


Fig. 10.10 Noseboom-mounted 5-hole probe for air data. a) Longer noseboom on ATTAS, b) short noseboom on Transall C-160.

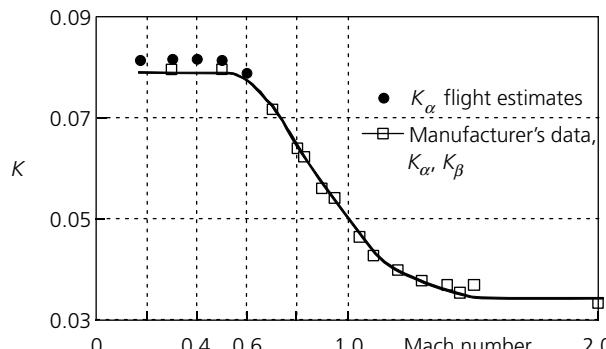


Fig. 10.11 Comparison of flight estimated scale factor for ATTAS with manufacturer's data.

biases were found to be independent of the flap setting and other configuration changes.

The measurement of difference pressure for angle of attack showed a bias of 131 Pa and for the angle of sideslip 200 Pa, resulting mostly from misalignment. As demonstrated in Ref. 28, the time delay in both the measurements was of the order of 130 ms, which was verified in a separate laboratory calibration to result from recording equipment. It is necessary to point out that the scale factors, biases and time delays are system installation-specific. Any generalization to other systems or configurations requires careful consideration.

In the other case, the same five-hole probe was mounted on a shorter noseboom on a Transall C-160 aircraft (Fig. 10.10b). The estimated parameters (sensitivity coefficients) are shown in Fig. 10.12, which clearly shows that the

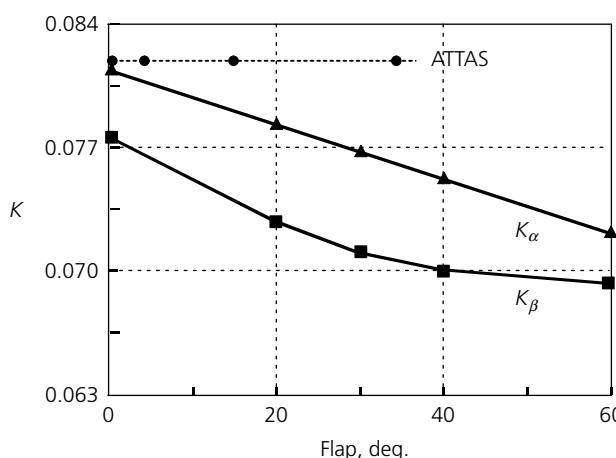


Fig. 10.12 Flight estimated calibration factors of Transall C-160.

scale factors are configuration-dependent. This is attributed to the fact that the noseboom length was just about 0.5 m. It is well known that the flow field around the aircraft is distorted. A large number of maneuvers were necessary to cover the variations with configuration change.

As already pointed out at the beginning of this section, the procedure just described is applied to calibrate flow angle vanes or a five-hole probe. Five-hole probes have faster response, and are more robust and reliable and better suited for operation in adverse weather conditions. It is known that the dynamics of the vane influences the measurements of the flow angles. Moreover, vanes have mechanical difficulties such as friction and grit in the bearings and are prone to malfunctioning in bad weather conditions. Both these types of sensors mounted on a boom located at a suitable location, such as nose, wingtip, or rudder tip, are used on a wide variety of research and prototype aircraft. Booms are sensitive to vibrations and alignment errors. Other aircraft have vanes mounted on the fuselage, typically at the pilot station. In general, they provide adequate measurement of flow angles in the low-to-medium range of angle of attack. In specific cases these two options to measure the flow angles may not be suitable or desirable because of 1) limited measurement range, 2) external fittings influencing the flow field affecting the aerodynamic performance, and 3) harsh thermal environment during supersonic speeds for entry vehicles. Furthermore, mounting a boom may not be possible due to radar equipment in the nosecone or due to interference with radar performance (radar backscatter). For stealth aircraft a boom is highly undesirable, because the pointed boom increases the visibility. In such cases a flush air data system (FADS) provides a better option to measure the flow variables.

The FADS concept is based on a nonprotruding probe, which consists of multiple pressure ports bored into the aircraft nose. They measure the local pressure distributions from which the airflow parameters are determined by applying suitable algorithms. Redundancy concepts and failure detection can be incorporated into these algorithms to yield reliable flow variables in the event of malfunctioning of some ports due to blockage or other reasons. These algorithms are derived from extensive wind-tunnel data and subsequently verified from flight data. FADS with multiple ports have also been used in the following applications: the SEADS (Shuttle Entry Air Data System) consisting of 20 flush orifices (14 primary and 6 supplementary orifices),³¹ or a 25-port HI-FADS suitable for high angle of attack tested on HARV,³² or RT-FADS with 11 orifices on F-18SRA (System Research Aircraft),³³ and FADS with 12 pressure ports (see Fig. 10.13) optimized for high angle-of-attack measurements on VECTOR.^{34,35} Common to all these systems is the basic layout of the orifices, which are located for a symmetric nose cap on concentric radial rings, and a single nose tip port. The arrangement of the individual ports on the nose cap defined by the cone and clock coordinate angle is optimized for specific aircraft. For nonsymmetric form of the nose cap, as in the case of Space Shuttle,³¹ the orifices are arranged in a cruciform array. A schematic of the pressure port locations of the FADS on VECTOR is shown in Fig. 10.13. The five pressure ports marked 1–5 are geometrically the same as those found on the five-hole probe discussed earlier; see Fig. 10.9. Such a FADS was calibrated by applying different algorithms.³⁴

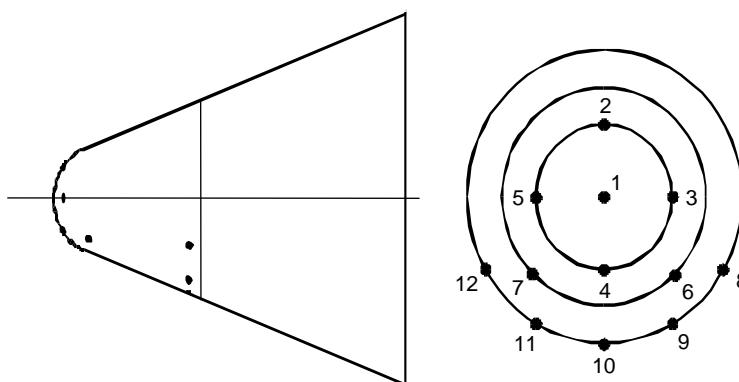


Fig. 10.13 Schematic of 12-hole flush air data system.

VII. Calibration of Static Pressure Ports

Calibration of flow angles measured by a five-hole probe needs the dynamic pressure, p_{dyn} , which is given by the difference between the total pressure p_{tot} measured by the central port, see Fig. 10.9, and the static pressure p_s , measured at a location very close by. Inaccuracies in the measurements of the static pressure at the noseboom directly affect the difference, that is, dynamic pressure, and subsequently the flow angles. Thus, accurate calibration of the static pressure port is a prerequisite for flow angle calibration. Similarly, deriving airspeed and altitude information from total and static pressures also requires accurate measurement of these pressures in free stream air. Besides indirectly influencing the other measured or computed variables, accurate static pressure information is also necessary for vertical spacing, for safety reasons. Generally, the static pressure measured by the pitot static system mounted on an aircraft in flight differs from the free stream pressure.^{36,37} This difference is primarily dependent on the location of the pressure ports, speed and altitude; secondary effects due to angle of attack and Reynold's number may also exist. Since the static pressure field varies along the length of an aircraft, see Fig. 10.14,³⁶ it is necessary to choose appropriately the location of the ports mounted on the fuselage. In the case of ports mounted on a noseboom, the length of the boom, which may be limited due to other considerations such as weight, structural vibrations and stability, is an important criterion.

The commonly adopted procedures to calibrate the static pressure ports are 1) trailing cone and 2) tower flyby.³⁸ The trailing cone procedure compares the static pressure measured by the aircraft's pitot static system with the free stream static pressure measured behind the aircraft by a static source. A sensor attached to a long cable or tube is necessary to sense the ambient pressure out of the aircraft's wake. The cone helps to stabilize the tube and to keep it taut. Some mechanism is necessary to let out the tube and cone in free atmosphere during flight, for example by opening the ramp door in the case of the military transport aircraft Transall C-160; see Fig. 10.15. This may not be possible in a

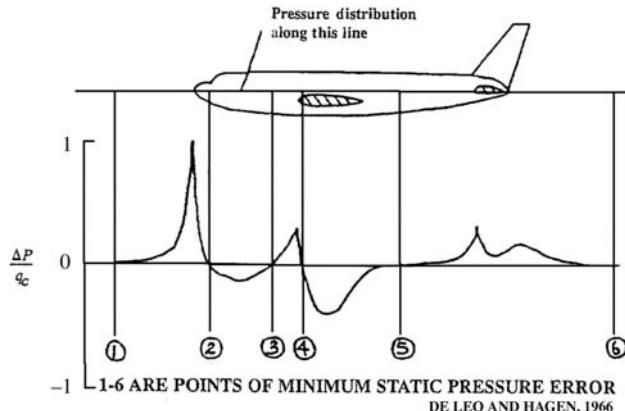


Fig. 10.14 Static pressure field along the length of aircraft (source: Ref. 36).

passenger transport aircraft. The overall procedure is somewhat difficult in practice, requires very accurate and steady flight, and is associated with some risk resulting from instabilities which are difficult to predict. On the other hand, the tower flyby method is a comparatively simpler procedure to accurately calibrate the static pressure measurements. It can be applied to any type of aircraft and somewhat less risky. The test procedure consists of flying the aircraft past an observation tower and recording the geometrical position of the aircraft, for example using a camera. The second approach, tower flyby, is adopted in this investigation.²⁸

A. Tower Flyby and Measurements

The tower flyby tests were performed with ATTAS aircraft at four flap positions covering the speed range from 120 to 240 KIAS (Mach number roughly



Fig. 10.15 C-160 flight test with trailing cone.

Table 10.1 Test flight conditions

Flaps KIAS	120	130	140	145	150	160	170	180	190	200	210	240
14 deg.	X		X			X			X			
5 deg.		X			X			X			X	
SP (1 deg.)			X				X			X		X
IN (-6 deg.)				X			X			X		X

0.14–0.45). Table 10.1 defines the flight test matrix of airspeed and flap setting. At each flight condition a tower flyby is performed, keeping height above the runway and speed as constant as possible. The nominal height above the runway is about 150 ft.

An onboard measurement system installed in the test aircraft ATTAS provides measurements of a large number of signals such as aircraft motion variables, atmospheric conditions, control surface positions, engine parameters, and so on. Pertinent to the current aspect of investigations, the onboard measurements of altitude, static pressure, total pressure, and temperature are obtained from a digital air data computer. The altitude AGL (above ground level, i.e., runway) is measured using the radio altimeter. Besides the above standard onboard measurements, the aircraft position above the runway during the flyby is filmed using photo-cameras as well as a digital-video camera from the top of the observation tower. The static pressure and temperature are recorded at the observation tower. Furthermore, the QNH, air pressure calculated to MSL (mean sea level), relayed by the control tower is recorded.

B. Data Analysis

A schematic of tower flyby maneuver analysis is shown in Fig. 10.16. Knowing the tower height, distance from tower to runway centerline, and to marking lights, it is a simple geometrical procedure of evaluating the photos to obtain the height above the runway. Having determined the AGL, it is straightforward to obtain the aircraft altitude above sea level from the known runway elevation. Figure 10.17 provides the altitude above sea level as obtained from the geometrically computed height above the runway from the two different cameras. Some small differences are observed in the altitudes obtained from the two different cameras. These minor deviations are attributed mostly to the variations in the flight path above the runway, different time points of exposure and the manual evaluation procedure. Nevertheless, these altitudes, obtained from two independent sources, match the radio altitude fairly well.

From the measured pressure altitude h (in meters) during the tower flyby, the static pressure p_s (in Pascal) is obtained as

$$p_s = p_{0\text{Akt}} \left(1 + \frac{T'}{T_{0\text{Akt}}} h \right)^{5.25588} \quad (10.40)$$

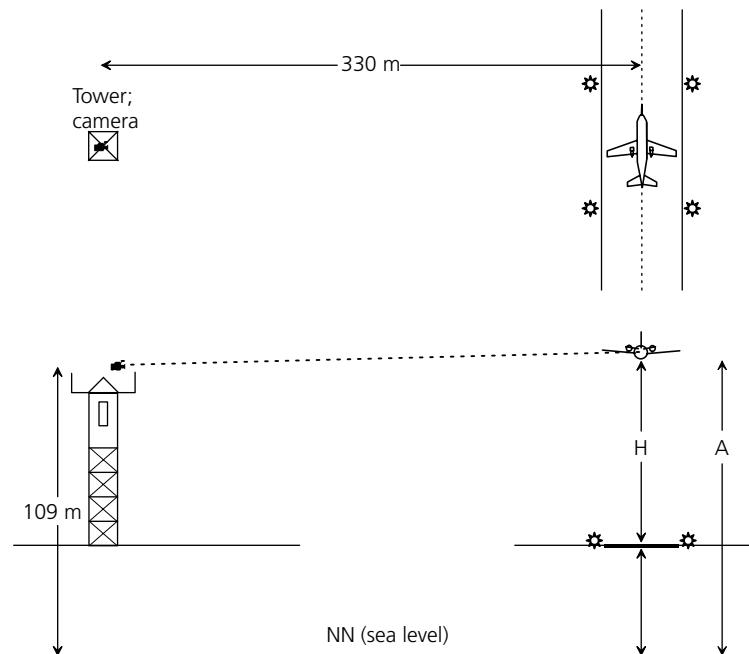


Fig. 10.16 Schematic of tower flyby maneuver analysis.

where $p_{0\text{Akt}}$ (in Pascal) and $T_{0\text{Akt}}$ (in Kelvin) are the current QNH and temperature at the time of the particular test, and T' is the temperature gradient, which is assumed to be -0.0065 K/m according to the standard atmosphere for altitudes up to 11000 m; $p_{0\text{Akt}}$ and $T_{0\text{Akt}}$ are noted during each tower flyby.

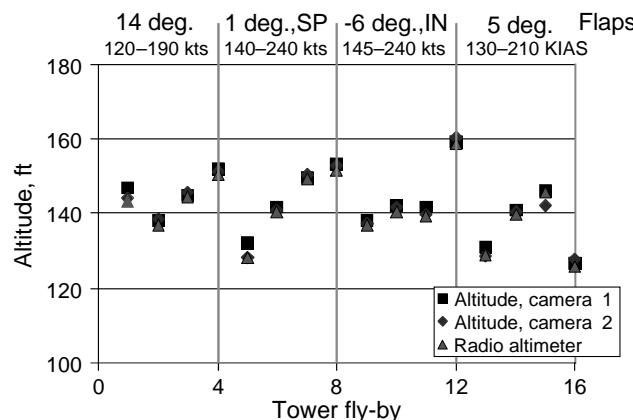


Fig. 10.17 Altitude above runway during tower flybys.

In the present investigations, it was noted that the variations in $T_{0\text{Akt}}$ during the complete flight test were less than 0.5°C and hence were neglected in the detailed computation of static pressure for each flyby. The average temperature was 18°C . The static pressure at sea level at takeoff was 1013 hPa and for the most part of the flight maneuvers up to completion of the test program after 1 h 40 min it was 1012 hPa.

Based on the above noted $T_{0\text{Akt}}$ and $p_{0\text{Akt}}$, the static pressure in hPa at the altitude h_{RA} during each flyby is obtained from the relation

$$p_{\text{sRA}} = 1012 \left(1 - 2.23253 * 10^{-5} * h_{\text{RA}}\right)^{5.25588} \quad (10.41)$$

These static pressure values p_{sRA} corresponding to the altitude above NN obtained from the radio altimeter measurements shown in Fig. 10.17 are plotted in Fig. 10.18.

Figure 10.18 also gives values of $p_{\text{s-m}}$ and $p_{\text{sNB-m}}$, the static pressure measured by the basis system (DADC) and that by the five-hole probe. These mean values were obtained from the time histories for the tower flyby phase during the 16 maneuvers. From Fig. 10.18, it is observed that, in general, the static pressure $p_{\text{s-m}}$ from the basis system matches well the p_{sRA} derived from the tower flyby tests. On the other hand, the static pressure values measured by the five-hole probe sensor assembly (points marked as triangles in Fig. 10.18) differ from the other two. After concluding from Fig. 10.18 that the DADC- and altimeter-derived values are correct, the position error in the static pressure at the noseboom is the difference between $p_{\text{sNB-m}}$ and p_{sRA} . Figure 10.19 shows this position error in static pressure measured at the noseboom as a function of speed. A simple

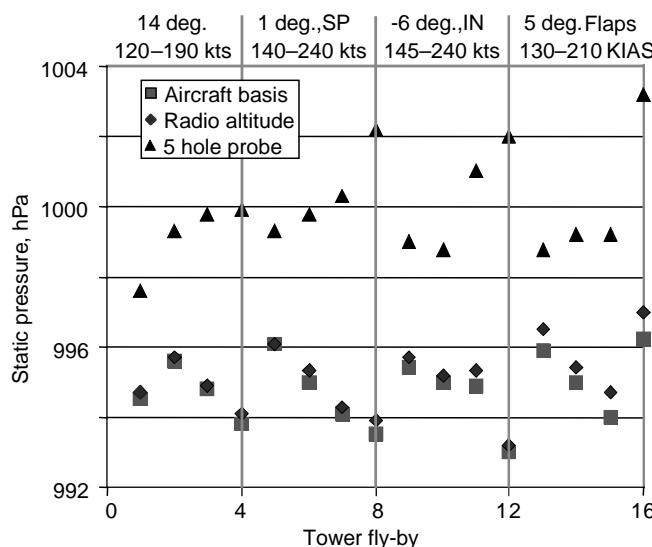


Fig. 10.18 Static pressures during tower flyby.

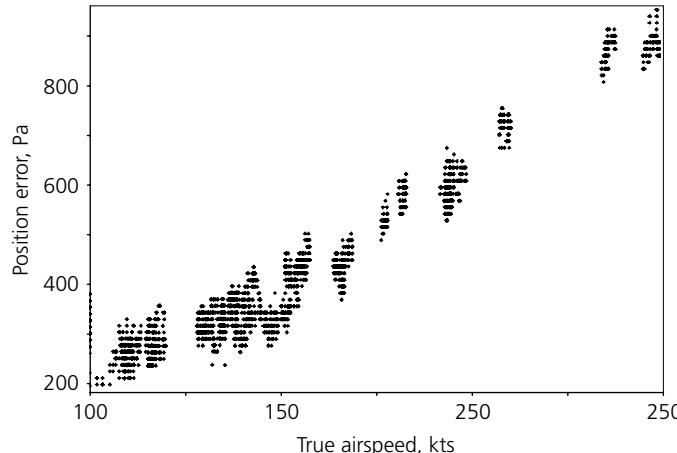


Fig. 10.19 Position error in static pressure measured at noseboom.

correction factor was identified from this evaluation. As already pointed out in Sec. VI, in the case of ATTAS the noseboom length was roughly 1.35 times the fuselage diameter, which is probably not sufficient to reach the free stream flow; see Fig. 10.10a.

VIII. Wind-box Maneuver Technique

A special flight test procedure, called wind-box maneuver, and a Kalman filter algorithm incorporating GPS information was proposed and used at NRC, Canada, to calibrate a flush air data system, consisting of five pressure ports.^{39,40} Calibration of such a flush air data system is, in general, configuration and flight condition-dependent, because the flow at the aircraft nose is strongly distorted. Although in the present investigation the flow angles are not measured using a flush air data system, but rather a five-hole probe mounted on a noseboom, we compare the standard technique described in Sec. VI and the approach presented in this section.

The wind-box maneuver leads to a typical horizontal flight track, shown in Fig. 10.20. It consists of acceleration and deceleration phases leading to angle of attack variations, build up of steady state sideslips in both directions, and 90 deg. turns to traverse the spatial field in x - y directions. This special type of maneuver helps to eliminate the wind effects. The typical duration of such a maneuver is about 700 s. Flight tests with ATTAS included two wind-box maneuvers, one at 8000 ft and another at 16,000 ft. The following signals are necessary to analyze this maneuver: 1) static pressure p_s , 2) dynamic pressure p_{dyn} , 3) total temperature, T_{tot} , 4) difference pressure for angle of attack p_{da} , and for angle of sideslip, p_{ds} , 5) Euler angles ϕ , θ , and ψ , 6) angular rates p , q , and r , and 7) three GPS inertial speed components.

Two approaches to analyzing such a maneuver are possible, one for off-line analysis and one for real-time onboard verification. We consider here the

DATA COMPATIBILITY CHECK

369

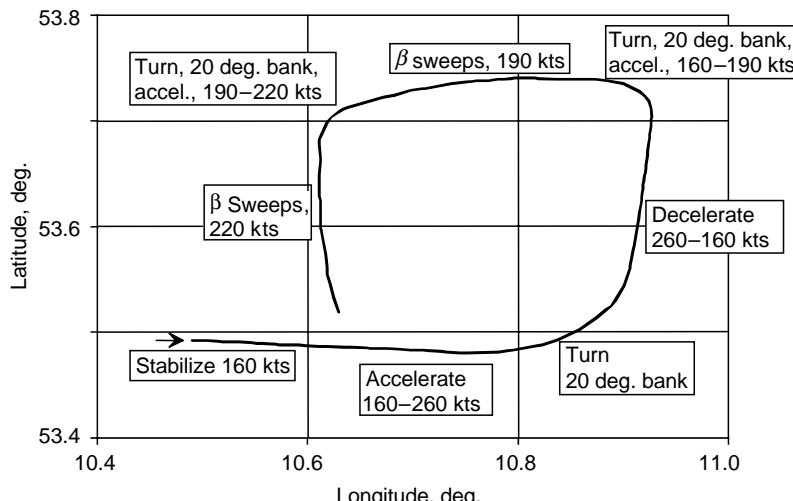


Fig. 10.20 Typical flight track during wind-box maneuver.

second method, which is based on a filter technique incorporating GPS information. As is common to all such techniques, it is necessary to postulate suitable models for propagation of states and for observations, in the present case the combined air-data and navigation equations. The propagation equations are in terms of the relationships between the errors in the air data navigation and those in the geographic velocity and position. The detailed equations for the Kalman filter algorithm and those for the aircraft velocity components are found in Ref. 40; here we just focus on the part related to air data. The dynamic pressure is modeled as

$$p_{\text{dyn}} = C_{p0} + (1 + C_{p1})p_{\text{dyn_m}} \quad (10.42)$$

where C_{p0} and C_{p1} are the unknown coefficients modeling the position error. The flow angles are modeled as

$$\begin{aligned} \alpha &= C_{A0} + C_{A1} \frac{p_{d\alpha}}{p_{\text{dyn}}} \\ \beta &= C_{B0} + C_{B1} \frac{p_{d\beta}}{p_{\text{dyn}}} \end{aligned} \quad (10.43)$$

where C_{A0} and C_{B0} account for the misalignment between noseboom and aircraft axes, and C_{A1} and C_{B1} are sensitivity factors. As described previously, $p_{d\alpha}$ and $p_{d\beta}$ are the difference pressures corresponding to angle of attack and angle of sideslip, and p_{dyn} is the dynamic pressure. Although Eq. (10.43) may appear to be formulated differently, it is basically the same as Eq. (10.38). It is obvious that C_{A1} and C_{B1} appearing in Eq. (10.43) are inverses

of the sensitivity coefficients K_α and K_β appearing in Eq. (10.38). We sacrifice here the uniformity of notation to facilitate further referencing the corresponding literature.

The Kalman filter algorithm consists of 18 error states (three air navigator position errors, three wind component errors, two calibration errors for angle of attack, two calibration errors for angle of sideslip, two calibration errors for position error correction, three aircraft Euler angle errors and three GPS position offset errors). All of the error states, other than the three air data navigational position errors, are modeled as first-order Gauss–Markov processes. A schematic of the error feedback Kalman filter is shown in Fig. 10.21.

For a typical wind-box maneuver, Fig. 10.22 shows time histories of the measured and estimated variables. The reconstructed variables match the measured data fairly well. It is observed that the sensitivity coefficients are estimated with very low standard deviation. The sensitivity coefficients C_{A1} and C_{B1} are estimated as 12.03 and 12.02, respectively.

The results obtained from Eq. (10.43) applying the Kalman filter technique to a wind-box maneuver are compared with those obtained from Eqs. (10.38) and (10.39) applying the output error method to dynamic maneuvers. The inverse of the C_{A1} estimate yields 0.0831 ($= 1/12.03$), which compares fairly well with the estimate of K_α ($= 0.0819$). The same is true for the calibration factors pertaining to angle of sideslip. The minor differences are attributed primarily to the fact that in the Kalman filter technique the time delays in the flow angles were not accounted for. Investigations in the past have clearly demonstrated that omission of time delays can affect estimation results significantly. Furthermore, the noise covariances were prespecified and kept fixed; tuning of noise covariances can somewhat influence the estimates. Nevertheless, the agreement between the two different analysis techniques applied to two

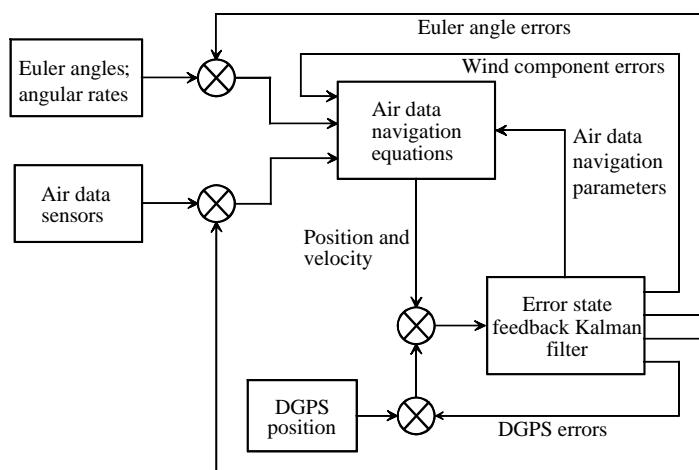


Fig. 10.21 Schematic of error feedback Kalman filter.

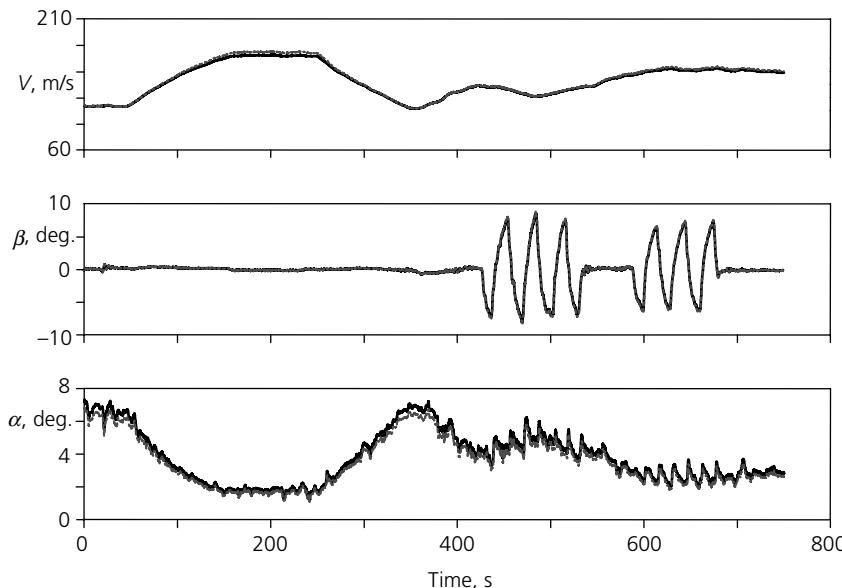


Fig. 10.22 Flight measured and reconstructed time histories for a wind-box maneuver (—, flight measured; - - -, reconstructed).

different types of flight maneuvers is good and increases the confidence in the results generated from flight data analysis.

The procedure described in this section is a relatively new approach. It can be implemented onboard in real time. As elaborated above, it allows simultaneous determination of flow angles and of position error; see Eq. (10.42). Elimination of position error leads to more reliable information on the dynamic pressure, and thereby on the true airspeed. The wind-box maneuver provides information necessary for this purpose. In the conventional approach discussed in Sec. II, we had mainly focused on calibration of flow angles from dynamic maneuvers only. As apparent from Fig. 10.22, the duration of the wind-box maneuver is quite long and time-consuming; if weather conditions are not stable during the test time, it will become necessary to improve upon the model to include varying wind conditions. Likewise, the Kalman filter performance may be affected by other measurements and may need proper tuning. Although an appealing approach, the maneuvering time and analysis methods appear somewhat cumbersome for repeated checking after minor modifications of the complete flight instrumentation system and for application of a routine basis. The conventional approach appears to be more direct and simpler.

IX. Concluding Remarks

The well-defined kinematic equations of aircraft motion provide a sound basis for a data compatibility check, which is often called the flight-path reconstruction

or aircraft state estimation. The separation of state estimation and estimation of aerodynamic model parameters is commonly termed the two-step method or estimation-before-modeling technique. The subtle differences between the two approaches have been clearly brought out. Strictly speaking, the so-called first step of state estimation is also a combined state and parameter estimation, since the procedure includes sensor models, and estimates parameters such as calibration factors, biases and time delays in the recorded signals. The problem may be of smaller size and easier than that of estimation of aerodynamic model parameters; nevertheless it is still the combined state and estimation problem that we had handled previously in Chapters 4 and 5.

A rigorous approach based on the extended Kalman filter and a simplified procedure using the classical output error method have been presented. The latter is applied to a simple test case first, and then to a detailed investigation pertaining to the calibration of a five-hole probe for flow angles from dynamic maneuvers. The importance of flight validation of the sensor laboratory calibrations has been highlighted. The corrections may be aircraft configuration-dependent. A related aspect of calibrating the static pressure port has been presented. To determine the position error in static pressure, the tower flyby technique was adopted. Finally, an alternative procedure based on a wind-box maneuver incorporating an error-feedback Kalman filter and GPS information has been elaborated, which allows a simultaneous calibration of flow angle, dynamic pressure and static pressure sensors. A data compatibility check to generate smoothed states and to eliminate instrument errors is an important part of aircraft modeling from flight data, because it helps to obtain more reliable and accurate estimates of aerodynamic derivatives.

References

- ¹Gerlach, O. H., "Determination of Performance, Stability and Control Characteristics from Measurements in Non-steady Manoeuvres," CP-17, AGARD, Sept. 1967, pp. 499–523.
- ²Jonkers, H. L., "Application of the Kalman Filter to Flight Path Reconstruction from Flight Test Data Including Instrumental Bias Error Corrections," Ph.D. Thesis, Delft University of Technology, Delft, 1976.
- ³Mulder, J. A., Sridhar, J. K., and Breeman, J. H., "Identification of Dynamic Systems, Applications to Aircraft, Part 2: Nonlinear Analysis and Manoeuvre Design," AGARD AG-300, Vol. 3, Part 2, May 1994.
- ⁴Stalford, H. L., "High-Alpha Aerodynamic Model Identification of T2C Aircraft Using the EBM Method," *Journal of Aircraft*, Vol. 18, No. 10, 1981, pp. 801–809.
- ⁵Sri-Jayantha, M. and Stengel, R. F., "Determination of Nonlinear Aerodynamic Coefficients Using the Estimation-Before-Modeling Method," *Journal of Aircraft*, Vol. 25, No. 9, 1988, pp. 796–804.
- ⁶Hoff, J. C. and Cook, M. V., "Aircraft Parameter Identification Using an Estimation-Before-Modeling Technique," *Aeronautical Journal*, Vol. 100, No. 997, Aug.–Sept. 1996, pp. 259–268.
- ⁷Etkin, B., *Dynamics of Atmospheric Flight*, John Wiley & Sons, New York, 1972.

DATA COMPATIBILITY CHECK

373

⁸Maine, R. E. and Iliff, K. W., "Identification of Dynamic Systems—Applications to Aircraft. Part 1: The Output Error Approach", AGARD-AG-300, Vol. 3, Part 1, Dec. 1986.

⁹Jategaonkar, R. V., "Maximum Likelihood Parameter Estimation Method for a Class of Nonlinear Systems and Its Applications to Problems in Flight Mechanics," Ph.D. Thesis, Indian Institute of Science, Bangalore, Jan. 1986.

¹⁰Jategaonkar, R. V., "Identification of the Aerodynamic Model of the DLR Research Aircraft ATTAS from Flight Test Data," DFVLR-FB 90-40, July 1990.

¹¹Weiss, S., Friehmelt, H., Plaetschke, E., and Rohlf, D., "X-31A System Identification Using Single-Surface Excitation at High Angles of Attack," *Journal of Aircraft*, Vol. 33, No. 3, 1996, pp. 485–490.

¹²Keskar, D. A. and Klein, V., "Determination of Instrumentation Errors from Measured Data Using Maximum Likelihood Method," AIAA Paper 80-1602, Aug. 1980.

¹³Feik, R. A., "A Maximum Likelihood Program for Non-Linear System Identification with Application to Aircraft Flight Data Compatibility Checking," ARL-AERO-Note 411, July 1982.

¹⁴Mulder, J. A., "Estimation of the Aircraft States in Non-Steady Flight," AGARD CP-172, Paper No. 19, May 1975.

¹⁵McBrinn, D. E. and Brassell, B. B., "Aerodynamic Parameter Identification for the A-7 Airplane at High Angles of Attack," *Proceedings of the 3rd Atmospheric Flight Mechanics Testing Conference*, 1976, pp. 108–117.

¹⁶Klein V. and Schiess J. R., "Compatibility Check of Measured Aircraft Responses Using Kinematic Equations and Extended Kalman Filter," NASA TN D-8514, Aug. 1977.

¹⁷Mulder, J. A., Jonkers, H. L., Horsten, J. J., Breeman, J. H., and Simons, J. L., "Analysis of Aircraft Performance, Stability and Control Measurements," AGARD LS-104, Paper No. 5, Nov. 1979.

¹⁸Martin, C. A. and Feik, R. A., "Estimation of Aircraft Dynamic States and Instrument Systematic Errors from Flight Test Measurements," *Proceedings of the 2nd Conference on Control Engineering*, Newcastle, 1982.

¹⁹Evans, R. J., Goodwin, G. C., Feik, R. A., Martin, C., and Lozano-Leal R., "Aircraft Flight Data Compatibility Checking Using Maximum Likelihood and Extended Kalman Filter Estimation," *Proceedings of the 7th IFAC Symposium on Identification and System Parameter Estimation*, Pergamon Press, Oxford, 1985, pp. 487–492.

²⁰Plaetschke, E., "Kalman-Filterung," Lecture Notes: "Praktische Einführung in die Systemidentifizierung," Carl-Cranz-Gesellschaft, Oberpfaffenhofen, Germany, 11–13 June 1985, Section 10 (in German).

²¹Bach, R. E., "State Estimation Applications in Aircraft Flight Data Analysis," NASA RP 1252, March 1991.

²²Gelb, A., *Applied Optimal Control*, The MIT Press, Cambridge, MA, 1974.

²³Ljung, L., "Asymptotic Behaviour of the EKF as a Parameter Estimator for Linear Systems," *IEEE Transactions on Automatic Control*, Vol. AC-24, No. 1, 1979, pp. 36–50.

²⁴Rauch, H. E., Tung, F., and Striebel, C. T., "Maximum Likelihood Estimation of Linear Dynamic Systems," *AIAA Journal*, Vol. 3, No. 8, 1965, pp. 1445–1450.

²⁵Haykin, S., "Kalman Filters," in *Kalman Filtering and Neural Networks*, edited by S. Haykin, John Wiley & Sons, New York, 2001, pp. 1–21.

²⁶Morelli, E. A., "Estimating Noise Characteristics from Flight Data Using Optimal Fourier Smoothing," *Journal of Aircraft*, Vol. 32, No. 4, 1995, pp. 689–695.

²⁷Jategaonkar, R. V., Fischenberg, D., and von Gruenhagen, W., "Aerodynamic Modeling and System Identification from Flight Data—Recent Applications at DLR," *Journal of Aircraft*, Vol. 41, No. 4, 2004, pp. 681–691.

²⁸Parameswaran, V., Jategaonkar, R. V., and Press, M., "Calibration of Five-Hole Probe for Flow Angles from Dynamic and Tower Flyby Maneuvers," *Journal of Aircraft*, Vol. 42, No. 1, 2005, pp. 80–86.

²⁹"Rosemount Model 858 Flow Angle Sensors," Bulletin 1014, Rosemount Inc., Burnsville, MN, 1988.

³⁰DeLeo, R. V. and Hagen, F. W., "Aerodynamic Performance of Rosemount Model 858AJ Air Data Sensor," Rosemount Report 8767, Minneapolis, MN, July 1976.

³¹Pruett, C. D., Wolf, H., Heck, M. L., and Siemers, P. M., "Innovative Air Data System for the Space Shuttle Orbiter," *Journal of Spacecraft and Rockets*, Vol. 20, No. 1, 1983, pp. 61–69.

³²Whitmore, S. A., Moes, T. R., and Larson, T. J., "High Angle-of-Attack Flush Airdata Sensing System," *Journal of Aircraft*, Vol. 29, No. 5, 1992, pp. 915–919.

³³Whitmore, S. A., Davis, R. J., and Fife, J. M., "In-Flight Demonstration of a Real-Time Flush Airdata Sensing (RT_FADS) System," AIAA Paper 95-3433, Aug. 1995.

³⁴Weiss, S., "Comparing Flush Air Data System Algorithms Using VECTOR-FADS Wind Tunnel Data," AIAA Paper 2002-0535, Jan. 2002.

³⁵Jost, M., Schwegmann, F., and Köhler, T., "Flush Air Data System—An Advanced Air Data System for the Aerospace Industry," AIAA Paper 2004-5028, Aug. 2004.

³⁶DeLeo, R. V. and Hagen, F. W., "Flight Calibration of Aircraft Static Pressure Systems," Federal Aviation Agency Contract FA 64WA-5025, Report No. RD-66-3, Rosemount Engineering Co., Minneapolis, MN, Jan. 1966.

³⁷Brown, E. N., "Position Error Calibration of a Pressure Survey Aircraft Using a Trailing Cone," National Center for Atmospheric Research, Technical Note NCAR/TN-313 + STR, Boulder, CO, July 1988.

³⁸Gallagher, G. L., Higgins, L. B., Khinoo, L. A., and Pierce, P. W., "Fixed Wing Performance," Naval Test Pilot School Flight Test Manual, Report USNTPS-FTM-No. 108, Sept. 1992.

³⁹Hui, K., Srinivasan, R., and Baillie, S., "Simultaneous Calibration of Aircraft Position Error and Airflow Angles Using Differential GPS," *Canadian Aeronautics and Space Journal*, Vol. 42, No. 4, Dec. 1996, pp. 185–193.

⁴⁰Leach, B. W. and Hui, K., "A DGPS-Based Technique for Real-Time Simultaneous Calibration of Air Data Systems Using Kalman Filtering," *Proceedings of the 13th Symposium on "Navigation," Canadian Aeronautics and Space Institute*, 2000, pp. 165–174.

Chapter 11

Model Validation

I. Introduction

ESTIMATES ARE not the same as the facts. Model validation is necessary to gain confidence in, or reject, a particular model. This basic fundamental principle applies to all engineering and other processes. In our particular case of flight vehicles, as discussed in Chapter 1 and depicted in Figs. 1.2 and 1.5, the parameter estimation and the model validation are integral parts of system identification. From the foregoing chapters, it is apparent to the readers that the parameter estimation methods provide an answer to the question: "Given the system inputs and responses, what is the model?" On the other hand, model validation tries to provide an answer to the question:¹ "How do you know that you got the right answer?" It is obvious that the answer in this case means the identified model. In this chapter we deal with the issues related to the process of determining the correctness, accuracy, adequacy, and applicability of the identified model. We also attempt to check the validity of the underlying theoretical assumptions which were made in the derivation of the parameter estimation methods applied. An apt definition of model validation is provided by Schlesinger et al.²

Validation refers to the process of confirming that the conceptual model is applicable or useful by demonstrating an adequate correspondence between the computational results of the model and the actual data (if it exists) or other theoretical data.

The various different aspects of model validation can be broadly classified into three subcategories: 1) statistical properties of the estimates, 2) residual analysis, and 3) model predictive quality. These three techniques provide clues to the effectiveness, or lack thereof, of model parameters. They provide suitable means of judging the adequacy of identified models and their parameters to duplicate the system sufficiently closely.

Careful analysis of each of these aspects is of paramount importance. Some of the criteria used in this methodological process are related to the basic statistical properties, whereas others have conceptually evolved keeping in mind the application domain of the identified model. Several criteria, to be used in conjunction with each other, that help to validate the identified model, can be

broadly classified as: 1) standard deviations of the parameter estimates, 2) correlation coefficients among the estimates, 3) goodness of fit (i.e., value of the cost function being minimized, e.g., determinant of the covariance matrix of the residuals), 4) statistical analysis of residuals (bias, variance, covariance, and power spectral density), 5) model deficiencies in terms of residual control inputs (inverse simulation), 6) plausibility of estimates, and 7) model predictive capability.

In most of the identification exercises, it is necessary to predefine the objectives of the model, that is, for what purpose the model is to be used. This dictates in most of the cases the amount of effort spent on arriving at an adequate model for the intended purpose, because if the validation process indicates inadequacy of the model, the parameter estimation and model development processes will have to be repeated. Additional experiments may become necessary to improve the estimates or to extend the range of model validity. This iterative procedure of model identification and model validation is to be performed again and again until the validation process proves adequate correspondence of the model.

The techniques that we discuss in this chapter can be applied to validate simple models derived for a particular trim point as well as for global models covering the entire operational flight regime. The basic philosophy in the two cases remains the same.

II. Statistical Accuracy of Parameter Estimates

Having applied statistical estimation methods, the first and the most natural question is about the statistical accuracy of these estimates. Common criteria are the standard deviations of the estimates and the correlation coefficients amongst them. As already pointed out in Chapter 4, Sec. III, for the maximum likelihood estimation, the Fisher information matrix, Eq. (4.30), provides a good approximation to the parameter error covariance matrix.^{3,4} Thus,

$$P \approx \left\{ \sum_{k=1}^N \left[\frac{\partial y(t_k)}{\partial \Theta} \right]^T R^{-1} \left[\frac{\partial y(t_k)}{\partial \Theta} \right] \right\}^{-1} \quad (11.1)$$

The estimation error covariance matrix P thus depends upon the model parameters Θ , the data points being analyzed N , and the covariance matrix of the residuals R (i.e., on the measurement noise). If a priori information, as discussed in Chapter 4, Sec. XVI, is included in the estimation, it is necessary to use the system matrix of Eq. (4.86) while computing the parameter error covariance matrix in Eq. (11.1).

The diagonal elements of P , which are the variances of the estimates, are indicators of the accuracy of the estimates. The standard deviations of the estimates, also called Cramér–Rao bounds, can be readily computed from Eq. (11.1) as

$$\sigma_{\Theta_i} = \sqrt{p_{ii}} \quad (11.2)$$

and the relative standard deviation as a percentage is then simply given by $100 * \sigma_{\Theta_i} / \Theta_i$, where i is the index for the parameter.

In addition to the standard deviations, the correlation coefficients, which are a measure of the statistical dependence between the parameters, can be obtained from the off-diagonal elements of P as

$$\rho_{\Theta_i \Theta_j} = \frac{p_{ij}}{\sqrt{p_{ii} p_{jj}}} \quad (11.3)$$

The estimates are statistically more accurate the smaller the standard deviations and the smaller the correlation coefficients.

In practice, the estimated error bounds (i.e., the standard deviations), obtained from Eq. (11.2), are found to be too optimistic. The cause for this discrepancy has been traced to colored residuals.^{5–7} The output error method lumps together both process and measurement noise and the deterministic modeling error, thus corrupting the noise statistics. In addition, the assumption of Gaussian measurement noise used in the derivation of the estimator may not be true. Some theoretical techniques have been suggested to account for the band-limited (colored) noise in the residuals.^{5–8} We do not, however, discuss such options, mainly because these problems are unavoidable. Instead, we use the simpler ad hoc approach of a fudge factor to obtain a realistic estimate of the errors. It is common practice to use a fudge factor of typically 5–10. This is an ad hoc approach, but based on experience. It attempts to bridge the gap between theory (the assumption of independent noise samples) and practice (white noise is a mathematical assumption and residual spectra are seldom white).⁹

Having corrected the error bounds through a fudge factor, one would expect the scatter in the estimates of a particular derivative from the same model and from repeated experiments to agree within a band resulting from their corrected standard deviations. Unfortunately, we do come across cases in which this is not the case, that is, the scatter in the estimates is larger than the standard deviations. There is no sound theoretical procedure to overcome this discrepancy, and we have to keep in mind this subtle issue while interpreting these statistical accuracies. This issue is nontrivial, because we cannot avoid scatter in the estimates, which could result from a multitude of different causes, like insufficient and improper excitation, inadequate model postulate, measurement errors, atmospheric turbulence, and the wrong choice of estimation method.

The foregoing brief argument is true in the case of correlation coefficients as well. Usually smaller correlation coefficients among the estimates are simply ignored, but what is meant by “small” is not specified and may vary from application to application. Correlation coefficients greater than 0.9 require some attention, and those greater than 0.95 can be considered to be almost linearly dependent. Unfortunately, this is not true in all the cases; in some of the exercises we have come across estimates that were plausible and in the expected range, but the correlation coefficients were larger than the threshold indicated above or close to 1. If parameters are linearly dependent, it implies that we can achieve the same match by compensating for the changes due to variations in one parameter through proportional variations in the other. We come across higher correlations mostly between scale factors and measurement biases, when identified from data with limited information content (for example, few

trim conditions or not enough variations in the magnitude). Ideally such correlations are to be avoided in the estimation. Thus, in a limited sense, subject to the concerns just mentioned, these two statistical criteria help indirectly to improve upon the model.

III. Residual Analysis

Residual analysis is one of the most commonly used statistical approaches to judge the quality of the fit. The criteria checked are goodness of the fit in terms of cost function value, Theil's inequality coefficient, and whiteness of the residuals.

A. Goodness of Fit

The value of the cost function is a measure of the model fit (in the case of maximum likelihood method, it is $|R|$, the determinant of the covariance matrix of residuals). Although it is the most direct way to evaluate the model quality, there are some practical difficulties associated with the evaluation of the numerical values. As in the case of the statistical properties of the estimates, requiring variances of the estimated parameters and that the covariances must be small, the cost function should also be small. The cost function value depends upon the number of system outputs and amount of noise in the data analyzed. It also depends upon the units (radians or degrees for angles, radians per second or degrees per second for angular rates, meters per second per second or g for linear accelerations, meters or feet for altitude) of the variables being weighed in the cost function. For these reasons the absolute value alone does not yield any clues to the quality of the model fit. As a consequence, one difficulty that we encounter here is that no consistent workable criteria have been so far put forward for this aspect that would allow direct comparison of different models. Nevertheless, for a chosen number of system outputs and using consistently the same units, the cost function values can be used in a limited way to judge the improvements that may result from introducing additional aerodynamic parameters.

In the same connection we study yet another peculiar characteristic of the cost function. Very low value of the cost function does not necessarily guarantee that all the variables are matched equally well. If a better matching of any one output variable is feasible, the optimization procedure may do so even when the match for other variables may deteriorate, provided the overall cost function value is reduced. In an ideal case, perfect matching of a single output would be sufficient to reach the absolute minimum, namely zero; and yet the model could be completely inappropriate. Luckily, in practice we do not come across such a theoretical possibility, mainly because of the presence of measurement noise and modeling errors, both of which are always present. In some cases, a relative scaling of the variables is incorporated to aid matching of various variables equally well. Whether or not any such scaling is considered in the optimization will depend upon the postulated model and data being analyzed. In all the cases reported in this book, including the advanced examples of large systems or highly nonlinear models, we have used the standard cost function without facing difficulties of the above nature.

The mean of the residuals is expected to be zero. Accordingly, the standard deviations σ of the output errors is given by

$$\sigma_i = \sqrt{\frac{1}{N} \sum_{k=1}^N [z_i(t_k) - y_i(t_k)]^2}, \quad i = 1, 2, \dots, n_y \quad (11.4)$$

Since we deal with multi-input, multi-output systems, we have to check the standard deviations for each of the n_y output variables. Furthermore, Eq. (11.4) provides the total standard deviation based on N , that is, for the complete data analyzed. As we have already discussed in Chapter 3, multiple experiment analysis is a common procedure in flight vehicle system identification. In such a case, we have to check the standard deviations for each of the n_E time segments analyzed. In general, the standard deviation for each time segment must be of roughly the same order of magnitude as that of other experiments or better, and also the same as that given by Eq. (11.4). A significant deviation in any one of them suggests a poor model quality for that particular experiment. Based on this clue, we may attempt to determine the reason for the bad fit, which could be due to inappropriate system excitation in the particular experiment. Another possibility is that the model postulate may not be adequate to duplicate the actual system response, for example linear model postulate for strongly nonlinear phenomenon. It could also result from an inappropriate choice of the estimation algorithm, for example output error method being applied to data with process noise.

According to an old adage, goodness of fit is no criterion; it is a necessary but not a sufficient condition. A perfect fit of one or more experiments is likely, we hope, to result in good predicting capabilities for other data, but it does not guarantee perfect prediction capability, that is, ability to duplicate system response. We will discuss in more detail prediction capabilities and ways to check them in one of the following sections.

B. Theil's Inequality Coefficient and Decomposition of Fit Error

Yet another form of output statistics for the overall fit is provided by Theil's inequality.¹⁰ Compared with the goodness of fit in terms of the residuals, it provides a little more insight into the correlation between the two time series, in our case the measured aircraft responses z and the model predicted responses y . Theil's inequality coefficient for each of the output variables is defined as

$$U_i = \frac{\sqrt{1/N \sum_{k=1}^N [z_i(t_k) - y_i(t_k)]^2}}{\sqrt{1/N \sum_{k=1}^N [z_i(t_k)]^2} + \sqrt{1/N \sum_{k=1}^N [y_i(t_k)]^2}}, \quad i = 1, 2, \dots, n_y \quad (11.5)$$

where N is the total number of data points, z the measurement vector, and y the model output vector. Equation (11.5) provides n_y separate Theil's inequality coefficients, $U_i, i = 1, \dots, n_y$, corresponding to n_y outputs. It is also possible to define a single measure for the overall fit, called multiple Theil's inequality

coefficient, based on the fit errors in the multiple pairs of time series.¹¹ However, we prefer to treat the inequality coefficients U separately, because it aids in locating outputs those are matched better and those which are not. It is also possible to apply Eq. (11.5) separately to each of the n_E time segments being analyzed simultaneously. It helps to determine maneuvers for which the model shows comparatively larger discrepancies compared with others, which may result from modeling errors or high noise content of that particular maneuver. Theil's inequality coefficient, U , is a measure of degree of conformance between two time series. In statistical terminology, it represents the ratio of the root mean square fit error and the root mean square values of the measured and estimated signals summed together.

From Eq. (11.5), it is obvious that U is a normalized index, bounded within the closed interval between zero and unity; this is the main advantage of this procedure over the goodness of fit discussed in the foregoing section. A perfect fit is implied by $U = 0$ (case of equality), and $U = 1$ corresponds to the case of maximum inequality. Values of U close to 1 arise when the two time series are significantly different. Thus, Theil's inequality provides insight into the validity of the predicted responses. Although the acceptable value for U depends on the application, in general, as a rule of thumb, a value in the range 0.25–0.3 indicates a good agreement.

Theil also decomposed the fit error between the two time series, that is, $[z_i - y_i]$, in terms of bias, variance and covariance proportions given by^{10,11}

$$U_i^M = \frac{(\bar{z}_i - \bar{y}_i)^2}{(1/N) \sum_{k=1}^N [z_i(t_k) - y_i(t_k)]^2} \quad (11.6)$$

$$U_i^S = \frac{(\sigma_{z_i} - \sigma_{y_i})^2}{(1/N) \sum_{k=1}^N [z_i(t_k) - y_i(t_k)]^2} \quad (11.7)$$

$$U_i^C = \frac{2(1 - \rho_i)\sigma_{z_i}\sigma_{y_i}}{(1/N) \sum_{k=1}^N [z_i(t_k) - y_i(t_k)]^2} \quad (11.8)$$

where as before the subscript “ i ” ($i = 1, \dots, n_y$) refers to the i th output variable, \bar{z}_i and \bar{y}_i denote corresponding the mean values, σ and ρ the standard deviations and correlation coefficient, respectively, of the two time series z and y . They are given by

$$\sigma_{z_i} = \sqrt{\frac{1}{N} \sum_{k=1}^N [z_i(k) - \bar{z}_i]^2}, \quad \sigma_{y_i} = \sqrt{\frac{1}{N} \sum_{k=1}^N [y_i(k) - \bar{y}_i]^2} \quad (11.9)$$

$$\rho = \frac{1}{\sigma_{z_i}\sigma_{y_i}} \frac{1}{N} \sum_{k=1}^N [z_i(k) - \bar{z}_i][y_i(k) - \bar{y}_i] \quad (11.10)$$

Once again, we treat the three proportions defined in Eqs. (11.6)–(11.8) separately for each output variable.

The bias proportion U_i^M is a measure of the systematic error in the identified model, whereas variance proportion U_i^S represents the model's ability to duplicate the variability in the true system. The covariance proportion U_i^C is also called a measure of nonsystematic error. The above decomposition provides an insight into the causes of fit error. It is obvious that the bias and variance proportions should be very small, in an ideal case, zero; whereas the covariance proportion should be close to 1. The sum of these three proportions turns out to be 1. For both U_i^M and U_i^S , a large value, typically above 0.1, would be of concern and indicate the need for model update. Theil's inequality and the breakdown fit error has been used regularly for economic forecasting for quite some time;^{10,12} since recently it has also been used in flight vehicle system identification.^{13–15} In combination with visual inspection of the match between the time histories, these criteria provide slightly better insight into the properties of residuals.

C. Test for Whiteness

In an ideal case, the residuals must be white, that is, independent at different time instants. The autocorrelation function and the power spectral density of the residuals are statistical measures to check for whiteness. For a given set of samples, the autocorrelation matrix is estimated as

$$C(\tau) = \frac{1}{N} \sum_{k=\tau}^N [v(k) - \bar{v}][v(k - \tau) - \bar{v}]^T \quad (11.11)$$

where v is the fit error (residual), \bar{v} the mean, and $\tau = 1, 2, \dots$ the lag. If residuals $[v(t_k), k = 1, 2, \dots, N]$ are white, the autocorrelation coefficients C_i are zero for all τ . However, since we deal in practice with a finite length of data points N , the estimates of mean will not be true, even if the residual is part of a white noise sequence. To overcome these practical limitations of finite N , and also to exclude outliers, a confidence level is introduced in statistical sense. In many applications, it is customary to use a 95% confidence level. In this case, for all τ greater than 1, the correlation coefficient should lie in the band $\pm 1.96/\sqrt{N}$ more than 95% of the time. In such a case the population (as N goes to ∞) and sample (finite N) means are not different enough to matter. The choice of 95% confidence is somewhat arbitrary; 90 or 99% intervals have also been used in specific applications, but we do not go into any further detail for reasons which will be discussed next. A more detailed treatment on various aspects of autocorrelation functions and its statistical significance can be found in any standard textbook on time series analysis, for example Ref. 16.

We also know that the white noise has infinite bandwidth and the corresponding power spectrum is flat over the complete range. Accordingly, the power spectral density of the residual gives some rough idea about the noise statistics and of the validity of the theoretical assumptions made. The reader may recall from Chapter 5, Sec. XI.B that we used this criterion to compare the filter error and output error methods. The test of whiteness is rarely used critically, except to verify whether the process noise option would be more appropriate and the improvements obtained therefrom. However, if the test

for whiteness fails, that is, indicates colored noise, we generally do not undertake any steps of model update. Noise statistics is of secondary importance, our primary goal being a model that is sufficiently close to reality and with good predictive capability. Both these criteria are being verified more closely by other diagnostic tests being discussed in this chapter. The theoretical assumption of noise being Gaussian distributed was anyway made to keep the mathematical procedure tractable. Hence, as already pointed out in Sec. II, we do not discuss the techniques that have been suggested to account for band limited (colored) noise.

IV. Inverse Simulation

Recalling from Chapter 1, Sec. II the definition of simulation (given inputs u and system model f , find system output y), the inverse simulation, as the name implies, is the process of calculating desired controls, for the given system model and response. Thus, it is similar to the control problem definition (given system model f and output y , find control input u). The subtle difference is that the classical control problem does not require measured system responses, whereas inverse simulation explicitly needs measured responses, and leads to controls based on residuals.

The principle of inverse simulation is depicted in Fig. 11.1. The measured control inputs are fed to the identified mathematical model. The computed model response is compared with the flight measured aircraft response; the residuals, which contain the model deficiencies, are fed back to the system through a feedback controller.¹⁷ Usually, a proportional and integral controller is quite adequate for the purpose at hand. The feedback controller ensures that the computed model responses match the measured system response exactly, whereby the modeling deficiencies now appear in the control inputs.

This approach provides another interpretation of the model deficiencies in terms of inputs instead of outputs. Furthermore, the model does not deviate from the actual flight (trim) condition. For a sufficiently accurate model, the incremental control inputs Δu will be small enough and centered about zero. As a rule of thumb, during aerodynamic model development, deviations in

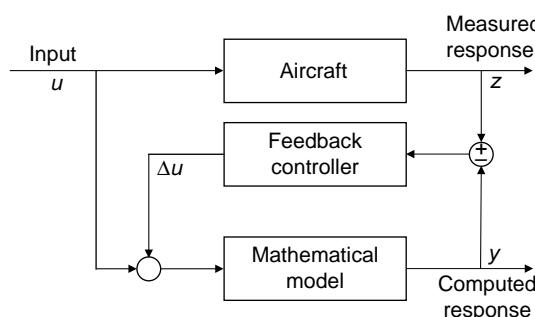


Fig. 11.1 Principle of inverse simulation.

control inputs smaller than half a degree are considered acceptable. Otherwise, the deviation in any particular control variable suggests a modeling error and provides clues to possible model extensions. The modeling task would then reduce to minimizing the controller activity to zero.

We can extend this approach to detect secondary aerodynamic effects such as those due to landing gear or ground effect. It calls for a flight validated model of the basic aircraft without the landing gear or ground effect.¹⁸ In such a case, the inverse simulation will show discernible controller activity. Thus, inverse simulation provides a suitable tool not only for the classical applications,^{19,20} for example validating the simulation software, but also for detecting modeling deficits, which can be reduced in the next step through system identification.

V. Model Plausibility

As pointed out in Sec. III.A, since the goodness of fit, that is, the response match, is a necessary but not sufficient condition, it is advisable to check estimates from yet another aspect, namely the plausibility of the estimates from physics of the dynamical system. An overparameterized model will give a good response match, but not necessarily a good system representation. The most direct way to check the plausibility of the estimated parameters is by comparison with estimates obtained from other sources. Since we are dealing mainly with flight vehicles, aerodynamic databases derived from wind-tunnel measurements on a scaled model are generally available. Hence, a comparison with such wind-tunnel-derived databases or with analytical estimates is a widely used approach to check the plausibility of the estimates.

Another commonly applied method is to check and interpret the estimates from physical understanding of the system under investigation. In such cases not only the polarity of the estimates, but also the relative magnitudes of the different derivatives contributing to a particular aerodynamic force or moment coefficient can provide clues to the plausibility. This is particularly true for the secondary derivatives which are difficult to estimate, for example, landing gear effects which are typically less than 5% of those for the clean configuration. Nevertheless, such effects are very important for critical phases of flight.²¹

Yet another example pertains to estimation of $C_{L\delta e}$, the lift curve slope of the elevator. The lift due to elevator being a very small portion of the total lift, the major portion coming from the wings, the estimates of $C_{L\delta e}$ are often associated with large standard deviations and scatter, both indicating lower confidence in the estimates. From the physics of the problem, by looking at the cause-effect relation, we know that an elevator deflection primarily leads to a pitching moment, resulting from small changes in lift due to elevator, but quite large lever arm (the tail length, that is, the distance from the neutral point of the wing to that of the tail). For this reason, the pitching moment due to elevator deflection $C_{m\delta e}$, the elevator effectiveness, is always well identifiable, which results mainly from matching the pitching motion variables. The two parameters $C_{L\delta e}$ and $C_{m\delta e}$ are physically coupled, and we can use this knowledge to improve the estimates by estimating just one parameter,²² namely $C_{L\delta e}$, and then

computing the contribution to the moment due to elevator through the relation

$$C_{m\delta e} = -C_{L\delta e} \frac{r_H}{\bar{c}} \frac{S_H}{S} \quad (11.12)$$

where r_H is the tail length, \bar{c} the mean aerodynamic chord, S the wing reference area, and S_H the horizontal tail area.

Yet another procedure to check the plausibility is to look at an equivalent linear system representation, which is helpful to gain insight into the system behavior and also to apply analysis techniques well established for linear systems, although the physical phenomenon being modeled may be nonlinear. The S-plane plot of the eigenvalues of the linearized system matrices derived from different identified models helps, for example, to highlight the influence of model extensions such as those due to the effect of rotor dynamics on the rigid-body motion of a helicopter.¹ The most commonly adapted numerical procedure to compute eigenvalues and eigenvectors is based on balancing the system matrix, reduction to upper Hessenberg form by orthogonal transformations and QR decomposition. Standard functions under Matlab® or from the LAPACK library in FORTRAN are available. A typical plot of eigenvalues is shown in Fig. 11.2, also showing the contours of the constant natural frequencies (ω_n) and constant damping ratios (ζ).

The eigenvectors are plotted in Fig. 11.3 as bar charts. We use here the normalized eigenvectors having Euclidean norm equal to 1. The vertical bars represent the magnitudes; the phase information is not considered in this form of representation. There are other possibilities to plot the eigenvectors, but the one depicted in Fig. 11.3 is easy to interpret and clearly shows the contribution of different variables to the particular mode. Depending upon the

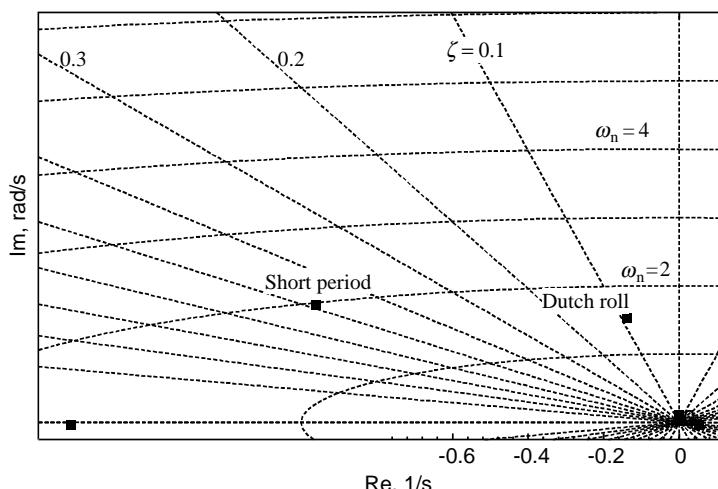


Fig. 11.2 Typical S-plane plot of eigenvalues.

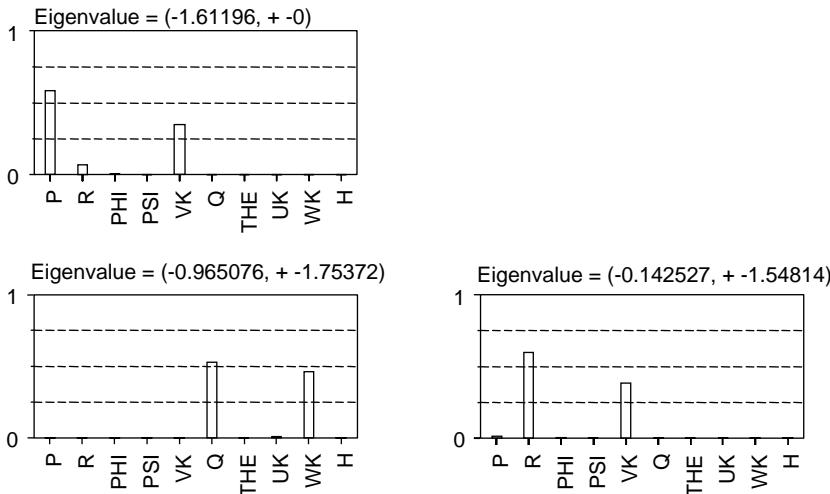


Fig. 11.3 Typical plot of eigenvectors.

largest contribution or the combination of individual components of the eigenvectors, the eigenvalues (modes of motion) can be determined. For example, the two plots at bottom of Fig. 11.3 with complex eigenvalues pertain to two oscillatory modes; on the left hand that of the short period mode with components due to pitch rate q and vertical velocity w , and on the right hand that of the Dutch roll with dominant components due to yaw rate r and lateral velocity v .

The eigenvalues and eigenvectors of the system state matrix completely characterize the system behavior; the eigenvalues determine the nature of the transient response whereas eigenvectors determine the contributions of each state to that response. The eigenvalues corresponding to the oscillatory modes of the system appear as complex conjugate pairs. The natural frequency ω_n and damping ζ of the oscillatory modes are given by

$$\omega_n = \sqrt{\text{Re}^2 + \text{Im}^2}, \quad \zeta = \frac{-\text{Re}}{\omega_n} \quad (11.13)$$

where Re and Im are the real and imaginary parts of the eigenvalues, respectively. Using the above formulas, the frequencies and damping of the short period and Dutch roll can be worked out as ($\omega_{SP} \approx 2 \text{ rad/s}$, $\zeta_{SP} \approx 0.48$) and ($\omega_{DR} \approx 1.55 \text{ rad/s}$, $\zeta_{DR} \approx 0.092$), respectively. These values can be traced to the two modes marked in Fig. 11.2. The eigensystem analysis is one way to investigate the system stability and to compute the frequency and damping ratio of oscillatory modes required for the proof-of-match cases that we will discuss in the next section.

The other approach is to approximate the oscillatory motion by a simple spring–mass–damper system and to apply one of the parameter estimation methods to estimate the damping and period. In such a case, we postulate a

second order system given by

$$\ddot{x} = -\omega_n^2 x - 2\zeta\omega_n \dot{x} + Ku \quad (11.14)$$

which is transformed into two first-order state equations for parameter estimation purposes as

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ K \end{bmatrix} u \quad (11.15)$$

where the two states are x and \dot{x} , the position and the velocity, respectively. We can now apply the output error method we discussed in Chapter 4 to estimate the parameters ω_n and ζ .

VI. Model Predictive Capability

The predictive capability of the identified model is determined by comparing the flight measured system responses with those predicted by the model for the same (“identical”) control inputs. In flight vehicle applications terminology, this process is often called “proof-of-match” (POM); it is an important part of flight simulator certification and acceptance. In this proof-of-match process, the identified (aerodynamic) model is kept fixed.

Besides the use of identical control inputs, three other issues that play a very important role are: 1) proper choice of the data set to compare against, 2) initial conditions on the state variables (trim), and 3) criteria to check adequate correspondence between model response and measured aircraft outputs. A fundamental principle of empirical sciences suggests, as a general rule, complementary flight data, that is, flight maneuvers not used in the estimation of the aerodynamic database, to check the model predictive capability. Validation on complementary data not used in the estimation is sometimes also termed loosely the “acid test.” In most of the model validation exercises, including those for the flight simulators, this approach of separating the data for model development and demonstration of model fidelity is adopted. However, there is another philosophy, although rarely practiced, which advocates incorporating the maximum amount of available flight data in the model development and not splitting the flight data in two sets. The logic behind this approach is that the identification process would automatically lead to a model with adequate fidelity, satisfying the validation tests that we will discuss in the following. It may also lead to fewer iterative steps in the overall model building. However, we recommend the first approach based on demonstrating the model fidelity on complementary data, because it provides increased confidence in the model predictive capability.

Conducting the flight tests for validation purposes generally calls for greater precision and caution than those for database development. It is also necessary to cover the complete operational envelope and to test the relevant modes of aircraft motion. Accordingly, for a full range of proof-match to demonstrate adequate model fidelity for simulator applications, the Federal Aviation Administration (FAA) has defined a set of a little more than 100 different cases, covering different modes of aircraft motion, and configurations.^{23,24} These tests are usually

carried out under calm atmospheric conditions to minimize the aircraft response due to stochastic inputs.

The second issue that plays a very important role is that of setting the proper initial conditions on the state variables to be used in the simulation.²⁵ Ideally, the simulation is to be started from the same initial conditions as in the flight. The initial conditions have to be suitably adjusted to match the flight conditions being verified. Unfortunately, measurements of the control inputs as well as those of the outputs (states) are subject to measurement errors and noise. Furthermore, even when the maneuvers for proof-of-match may be carried out under seemingly smooth atmospheric conditions, the aircraft is excited by some small amount of nonmeasurable excitation due to turbulence. Likewise, the identified models are not precise. Owing to these unavoidable practical difficulties, the proof-of-match turns out to be a nontrivial task. Starting with the initial conditions as measured in flight, in most of the cases it becomes difficult, if not impossible, to meet the tolerances. A pragmatic approach, therefore, is to allow small biases on the selected initial conditions and on the measured control deflections which are the inputs to the simulation.

To eliminate subjective evaluation of the match between measured system responses and model predicted outputs, FAA has specified guidelines in terms of tolerances for each variable, depending upon the nature of the validation test. As an example, Table 11.1 provides the definition of four tests, giving tolerances, flight conditions to be tested and comments for each. The reader is referred to Refs. 23 and 24 for the complete list of validation tests. The corresponding flight measurements with these tolerances define a band within which the model predicted response must lie to meet the necessary fidelity requirements.

The majority of the flight vehicle system identification applications deal with estimation of aerodynamic databases. In such cases, the proof-of-match is performed using the measured control surface deflections as the input to the model. For aircraft with an artificial feel system, this is quite adequate and an accepted procedure. In the case of aircraft with reversible controls (that is, with a mechanical control system), the task is a little more complex. Besides the aerodynamic database, the hinge-moment database is also required. For such cases, to improve the accuracy of each database, and also to reduce the computational burden during the database development, the two databases are estimated as so-called stand-alone models, that is, separately, as shown in Fig. 11.4a. On the other hand, it is necessary to demonstrate the end-to-end match based on an integrated model shown in Fig. 11.4b.^{26,27}

If the two stand-alone models are identified sufficiently accurately, the integrated model used in the end-to-end match (i.e., using six degrees-of-freedom equations of aircraft motion driven through the dynamic models of the flight controls incorporating the identified databases for rigid-body and hinge moments, respectively), is most likely to yield a match which will be acceptable within the tolerances. Owing to highly complex models, some deterioration in the end-to-end match compared with the stand-alone quality is unavoidable, although each may have fidelity far in excess of the required tolerances. In such a highly complex and demanding task of end-to-end validation, we are able to appreciate the need to define the purpose of model identification to minimize the overall effort, an issue that we discussed in Sec. I and Chapter 1, Sec. I. In this task

Table 11.1 Sample validation tests (source: Ref. 23)

Tests	Tolerances	Flight condition	Comments
2c10 phugoid dynamics	$\pm 10\%$ of period $\pm 10\%$ of time to half or double amplitude or 0.02 of damping ratio	Cruise	Test should include three full cycles (six overshoots after input completed) or that sufficient to determine time to half amplitude, whichever is less
2c11 short period dynamics	± 1.5 deg. pitch or ± 2 deg./s pitch rate ± 0.1 g normal acceleration	Cruise	Test in normal and non-normal control state
2d2 roll response (rate)	$\pm 10\%$ or ± 2 deg./s roll rate	Cruise and approach or landing	Test with normal wheel deflection (about 30%). Airplanes with reversible flight control systems must also plot wheel force ($\pm 10\%$ or ± 3 lb/1.3 daN)
2e1 normal landing	± 3 kts airspeed ± 1.5 deg. pitch ± 1.5 deg. angle of attack $\pm 10\%$ altitude or ± 10 feet (3 m)	Landing	Test from a minimum of 200 ft (61 m) AGL to nosewheel touch-down Derotation may be shown as a separate segment from the time of main gear touch-down. Medium, light, and near—maximum landing weights must be shown. Airplanes with reversible flight control systems must also plot stick/column force [$\pm 10\%$ or 2.2 daN (5 lb)]

the criteria to judge the adequate correspondence between real process and identified model play an important role, an issue that we have covered in this section.

The overall POM computational procedure is as follows. In the first attempt, we set the initial conditions of the states to those measured in flight; as already pointed out in Chapter 3, Sec. VII, better practice is to use an average of the first few data points to reduce the effects of noise and other measurement errors. Keeping the aerodynamic database fixed, we allow small biases Δu in the measured control inputs. These biases may be adjusted manually or an estimation algorithm can even be applied to determine them. If the resulting match between the measured and simulated responses is within the specified tolerances (Table 11.1), then the model fidelity is established, provided that the biases Δu are also within limits. In some cases, small adjustments of the initial conditions x_0 may also become necessary. Depending upon the particular test,

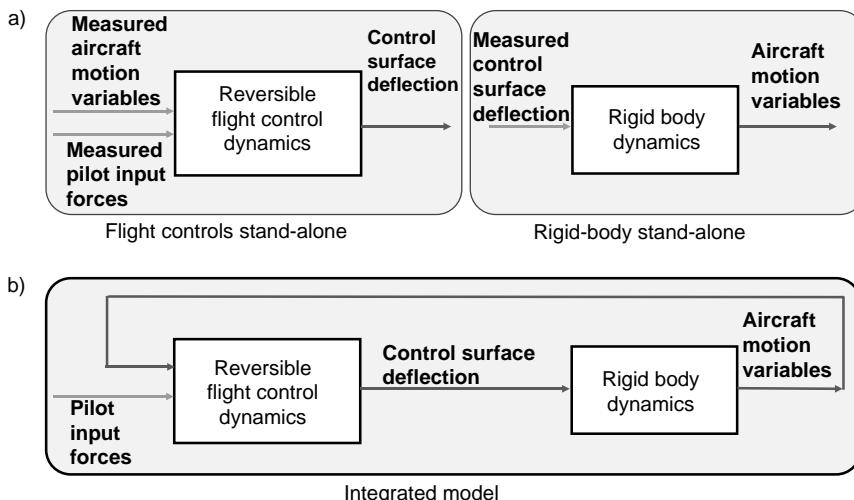


Fig. 11.4 Stand-alone vs integrated models. a) Stand-alone model identification, b) integrated model for validation.

pertaining to either longitudinal or lateral-directional motion, the tolerances are checked stringently for the variables pertaining to the major axis. A list of variables to be matched is provided by certifying authorities^{23,24} (see for example Table 11.1), or can be defined in the acceptance guide (ATG). For the other variables, particularly those of the cross axis, a qualitative match, showing correct trends, is usually considered adequate.

The above general procedure has been applied to validate the flight derived databases that we will discuss in Chapter 12, Sec. VIII. Here we demonstrate just the application of above basic procedure on a typical example. Figure 11.5 shows a proof-of-match for the validation test 2c11, short-period dynamics. The elevator input, and the match for the pitch rate and vertical acceleration are plotted. The model adequacy is quite apparent from the figure. The dashed lines are obtained from the measured data plus/minus the tolerances specified in Table 11.1, and the continuous lines show the model predicted output which is well within the allowed band for level-D model fidelity. Roughly 100–120 test cases, covering various configurations and maneuvers, including special test cases for military aircraft, are usually needed to demonstrate the fidelity of aerodynamic databases for flight simulators.

VII. Range of Model Applicability in Frequency Domain

Although the validation tests are performed in the time domain either through time histories or in terms of period and damping ratios of the oscillatory modes such as phugoid or Dutch roll, it is also possible to extend the verification to the frequency domain, which brings out more clearly the range of applicability of the

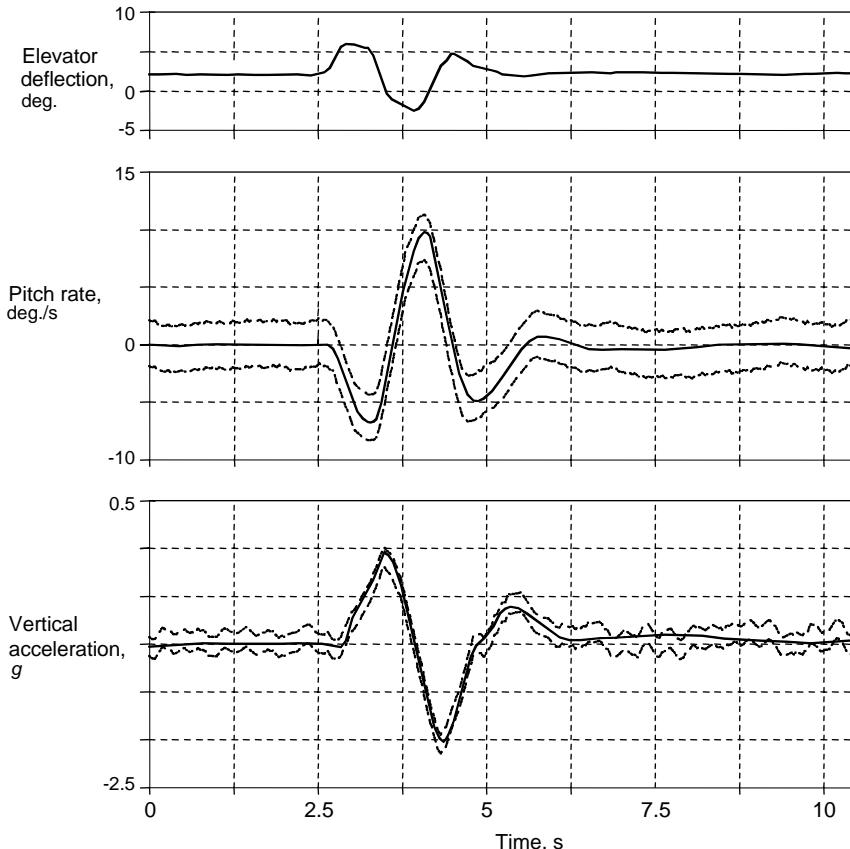


Fig. 11.5 Validation tests 2c11, short period dynamics (-----, flight measured \pm tolerances; ——, model estimated).

identified model.^{28,29} This is particularly important for high-authority flight control systems or in cases where aeroservoelastic effects may be dominant.

The procedure is fairly simple. It is illustrated for a single variable, pitch rate, on the example of short period motion considered in Sec. VI and the time-domain results shown in Fig. 11.5. In this approach, it is necessary to compute the frequency responses of the error function q_m/q , that is, of the ratio of measured to computed pitch rate. The Bode plot of magnitude and phase angle provides information on the model fidelity and also on the range of validity in the frequency domain. For a perfect match, we will have 0 dB magnitude and 0 deg. phase angle throughout the frequency range, which is possible for hypothetical cases and exercises dealing with simulated data only, but not in any practical application with real data. As in the case of time-domain verification in terms of tolerances, the mismatch boundaries in terms of magnitude and phase are defined. These boundaries are based on the LOES (low-order equivalent system) mismatch

criteria and provide boundaries of unnoticeable dynamics, that is, any discrepancies within the band are not perceptible to the pilot.³⁰ Accordingly, for sufficient fidelity, magnitude and phase of the error function should be within these boundaries.

For the case at hand, namely short period motion, the frequency domain verification is shown in Fig. 11.6, which shows the global model to be sufficiently accurate. The figure shows a more restrictive tolerance band in the range 1–5 rad/s pilot cross-over frequency, allows more error outside of it (LOES phase-lead leniency), and is asymmetric at low frequencies (allowing more phase lead error than phase lag error) to have better fidelity with respect to lag error, which might cause pilot-induced oscillations.³¹ The procedure to compute frequency responses is not covered in this book, but can be found in any standard book on frequency domain techniques. If the recorded data is too noisy, we may have to pay more attention while applying these techniques.

Besides looking at the error analysis as just discussed, a comparison of the Bode plot of measured system response and identified model provides useful information on the system dynamics, and possibly on model deficiencies. From the linearized system matrices A , B , C , and D , the frequency response matrix can be easily computed as given by:³²

$$G(j\omega) = C(j\omega I - A)^{-1}B + D \quad (11.16)$$

From the measured flight responses, it is possible to approximate frequency responses through classical fast Fourier transform (FFT) techniques. Since this is not our primary goal, we do not cover this subject here anymore. However,

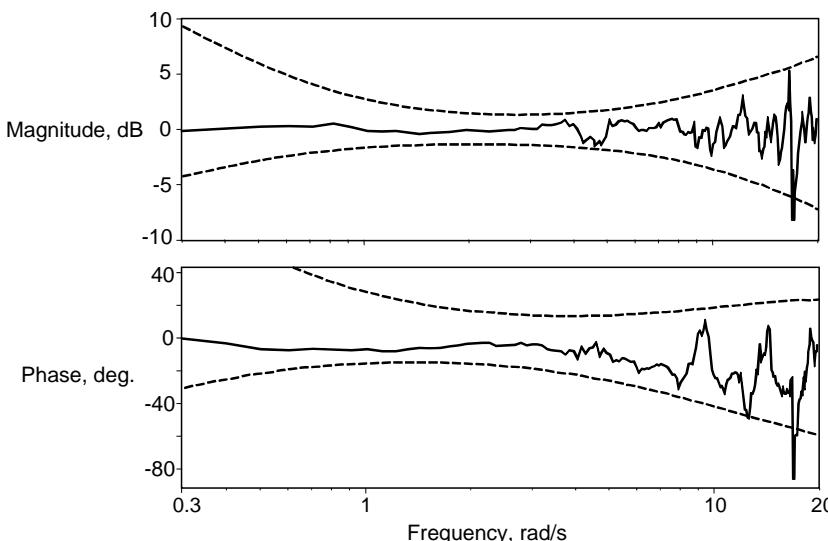


Fig. 11.6 Model validation in frequency domain.

as already mentioned, a caution is required when applying techniques to noisy data. Smoothing techniques may falsify the results and interpretations.

VIII. Concluding Remarks

Besides model estimation, model validation is the other central issue in system identification. In this chapter we discussed some diagnostic tests and approaches to assess the quality of estimated individual parameters and of the model as a whole, both qualitatively and quantitatively. We discussed in some depth procedures to determine the model quality in terms of statistical properties of the estimates and of residuals. Plausibility of the parameter estimates needs to be verified, because a good response match is just a necessary condition and not a sufficient one. Such verification is possible through a comparison of flight estimates with wind-tunnel or analytical predictions, or from physical understanding of the dynamic system.

It is pointed out that the demonstration of adequate model predictive capability, particularly for aerodynamic databases for flight simulators, is crucial. The difficulties encountered in the end-to-end match, the treatment of initial conditions on the state variables (trim), biases in the control inputs, and the importance of using complementary data, that is, data not used in the model development, are highlighted. It has been argued that, to minimize the effort, the application domain of the model must be defined in advance and the methodological procedures of this chapter applied accordingly to judge the model adequacy. Both the time domain and the frequency domain verification have been elucidated using a simple example to demonstrate the basic principle.

Through the discussion of practical issues, it is shown that there is no single criterion that would give us a clear cut "yes" or "no" answer about the model adequacy. Nevertheless, some guidelines on the methodological procedure are presented in this chapter. Those can be judiciously applied to gain confidence in, or reject, a particular model.

References

- ¹Hamel, P. G. and Jategaonkar, R. V., "Evolution of Flight Vehicle System Identification," *Journal of Aircraft*, Vol. 33, No. 1, 1996, pp. 9–28.
- ²Schlesinger, S., Buyan, J. R., Callender, E. D., Clarkson, W. K., and Perkins, F. M., "Developing Standard Procedures for Simulation Validation and Verification," *Proceedings of the Summer Computer Simulation Conference*, Houston, TX, Vol. 1, 9–11 July 1974, pp. 927–933.
- ³Plaetschke, E. and Mackie, D. B., "Maximum-Likelihood-Schätzung von Parametern Linearer Systeme aus Flugversuchsdaten—Ein FORTRAN Programm", DFVLR-Mitt. 84–10, Jan. 1984 (in German).
- ⁴Maine, R. E. and Iliff, K. W., "Identification of Dynamic Systems," AGARD AG-300 Vol. 2, Jan. 1985.
- ⁵Balakrishnan, A. V. and Maine, R. E., "Improvements in Aircraft Estimation Programs," NASA CR-145090, 1975.

⁶Iliff, K.W. and Maine, R. E., "Further Observations on Maximum Likelihood Estimates of Stability and Control Characteristics Obtained from Flight Data," AIAA Paper 77-1133, Aug. 1977.

⁷Maine, R. E. and Iliff, K. W., "Use of Cramér–Rao Bounds on Flight Data with Colored Residuals," *Journal of Guidance and Control*, Vol. 4, No. 2, 1981, pp. 207–213.

⁸Morelli, E. A. and Klein, V., "Accuracy of Aerodynamic Model Parameters Estimated from Flight Test Data," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 74–80.

⁹Maine, R. E. and Iliff, K. W., "Identification of Dynamic Systems—Applications to Aircraft Part 1: The Output Error Approach," AGARD AG-300 Vol. 3, Part 1, Dec. 1986.

¹⁰Theil, H., *Economic Forecasts and Policy*, 4th ed., North-Holland, Amsterdam, 1975.

¹¹Kheir, N. A. and Holmes, W. M., "On Validating Simulation Models of Missile Systems," *Simulation*, April 1978, pp. 117–128.

¹²Pindyck, R. S. and Rubinfeld, D. L., *Econometric Models and Economic Forecasts*, 3rd ed., McGraw-Hill, New York, 1991, pp. 336–342.

¹³Murray-Smith, D. J., "Methods for the External Validation of Continuous System Simulation Models: A Review," *Journal of Mathematical and Computer Modelling of Dynamical Systems*, Vol. 4, No. 1, 1998, pp. 5–31.

¹⁴Jategaonkar, R. V., Fischenberg, D., and von Grünhagen, W., "Aerodynamic Modeling and System Identification from Flight Data—Recent Applications at DLR," *Journal of Aircraft*, Vol. 41, No. 4, 2004, pp. 681–691.

¹⁵Paris, A. C. and Bonner, M., "Nonlinear Model Development from Flight Test Data for F/A-18E Super Hornet," *Journal of Aircraft*, Vol. 41, No. 4, 2004, pp. 692–702.

¹⁶Box G. E. P. and Jenkins, G. M., *Time Series Analysis—Forecasting and Control*, Holden Day, San Francisco, CA, 1970.

¹⁷von Grünhagen, W., "Inverse Simulation: A Tool for the Validation of Simulation Programs—First Results," *ZFW*, Vol. 17, No. 3, 1993, pp. 211–219.

¹⁸Fischenberg, D., "Ground Effect Modeling Using a Hybrid Approach of Inverse Simulation and System Identification," AIAA Paper 99-4324, Aug. 1999.

¹⁹Hess, R. A., Gao, C., and Wang, S. H., "Generalized Technique for Inverse Simulation Applied to Aircraft Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 920–926.

²⁰Rutherford, S. and Thomson, D. G., "Improved Methodology for Inverse Simulation," *Aeronautical Journal*, Vol. 100, No. 993, 1996, pp. 79–86.

²¹Hamel, P. G. and Jategaonkar, R. V., "The Role of System Identification for Flight Vehicle Applications: Revisited," RTO-MP-11, Paper 2, March 1999.

²²Jategaonkar, R. V., "Determination of Aerodynamic Characteristics from ATTAS Flight Data Gathering for Ground-Based Simulator," DLR-FB 91-15, May 1991.

²³"Airplane Simulator Qualification," FAA Advisory Circular, AC 120-40C, Draft Version, July 1995.

²⁴"Joint Aviation Requirements—Aeroplane Flight Simulators," JAR-STD 1A, Westward Digital, Cheltenham, April 1997.

²⁵Neville, K. W., "Simulator-to-Flight Validation—Methods, Problems and Dilemmas," *Proceedings of the Symposium on "Data Issues for Flight Simulators—An Ongoing Problem?"*, 10–11 Nov. 1993, Royal Aeronautical Society, London, pp. 14.1–14.7.

²⁶Jategaonkar, R. V. and Mönnich, W., "Identification of DO-328 Aerodynamic Database for a Level D Flight Simulator," AIAA Paper 97-3729, Aug. 1997.

²⁷Weiss, S., Gockel, W., Mönnich, W., and Rohlf, D., "Identification of Dornier-328 Reversible Flight Control System," AIAA Paper 98-3729, Aug. 1998.

²⁸Hamel, P. G. and Kaletka, J., "Rotorcraft System Identification—An Overview of AGARD FVP Working Group 18," AGARD CP-552, Paper 18, Oct. 1994.

²⁹Tischler, M. B., "System Identification Methods for Aircraft Flight Control Development and Validation," NASA TM 110369, Oct. 1995.

³⁰Hoh, R. H., Mitchell, D. G., Ashkenas, I. L., Klein, R. H., Heffley, R. K., and Hodgkinson, J., "Proposed MIL Standard and Handbook—Flying Qualities of Air Vehicles," Report AFWAL-TR-82-3081, Vol. 2, 1982.

³¹Hodgkinson, J. and Mitchell, D., "Handling Qualities," in *Flight Control Systems*, edited by R. W. Pratt, AIAA Progress in Astronautics and Aeronautics Series, Vol. 184, Chapter 4, 2000.

³²Laub, A. J., "Efficient Multivariable Frequency Response Computations," *IEEE Transactions on Automatic Control*, Vol. AC-26, No. 2, 1981, pp. 407–408.

Chapter 12

Selected Advanced Examples

I. Introduction

APPLICATIONS OF system identification methodology to flight vehicles have been too numerous to count. This is evident from the plethora of applications reported in the literature, both in terms of sheer number of aircraft configurations being analyzed and also in terms of modeling complexity. A representative summary of such applications is found in the special focus issue on “Flight Vehicle System Identification—Engineering Utility” of the *AIAA Journal of Aircraft*.¹ References 2–6 provide a consolidated account of activities related to aircraft parameter estimation from flight data. Common to most of these advanced applications is the time-domain methodology of system identification, which we discussed in Chapters 4–8. In those chapters, we also applied them to typical problems to bring out the underlying mathematics, computational details, practical issues, and scope of applicability. Simulated data as well as flight data were analyzed using the software described in the respective chapters. Both the data and software are supplemented to the book. These estimation techniques, either directly or in a suitably modified form, were applied to unstable aircraft in Chapter 9. In general, aerodynamic modeling aspects as well as those pertaining to the data compatibility check using flight path reconstruction techniques were discussed.

In the present chapter we focus on a few more advanced and comprehensive applications than those reported in the preceding chapters. Specifically, we address the following aspects: 1) modeling of transit time lag effect resulting from angle of attack or direct lift flap, or speed brakes and thrust variations, 2) estimation of aerodynamic effects of secondary order like those due to landing gear, 3) determination of effects due to control surface malfunction, namely roll-spoiler hardover, 4) modeling and identification of unsteady aerodynamics and quasi-steady stall covering effects in both longitudinal and lateral-directional modes, 5) extraction of ground effects for low-wing and high-wing configurations, 6) generation of high fidelity aerodynamic databases for level-D flight simulators of Transall C-160 military transport aircraft, Dornier 328 propeller aircraft and VFW-614 ATTAS in-flight simulator, 7) database validation and update of X-31A unstable aircraft, 8) wake vortex encounter model, 9) verification and update of predicted longitudinal aerodynamics of the reusable launch vehicle demonstrator Phoenix, and 10) rotorcraft modeling including rotor

degrees of freedom and wake distortion dynamics. In each case we analyze flight recorded data. We provide the flight data gathered during stall maneuvers with the ATTAS aircraft. In all these cases, we discuss in detail the modeling issues and results which were obtained therefrom. The goal of this chapter is to provide an appreciation of a wide variety of aerodynamic modeling and parameter estimation problems.

II. Modeling of Transit Time Lag Effects

In the theory of aircraft flight mechanics, it is well known that, for a conventional aircraft with wing and horizontal tail, the most significant effect of the wing on the tail is the downward deflection of the flow due to wing circulation, which can be aerodynamically modeled as a downwash angle ε .^{7–10} Such effects are present even under the assumption of quasi-steady flow and need to be properly accounted for. As the downwash is caused by circulation, it changes anytime the circulation changes; for example 1) changes in the wing lift due to changes in the angle of attack, 2) deflection of direct-lift-control flaps which are a modified part of the landing flaps on the wings, and 3) operation of wing-mounted speed brakes in flight. The modified flow at the wing, however, reaches the empennage after a time interval $\tau = r_H/V$, where V is the airspeed and r_H the tail length. Thus, the effective angle of attack at the tail is changed only after a time delay of τ s. The same transit time effect is also applicable to cases considering vertical wind based on Taylor's hypothesis of treating turbulence as a frozen pattern. Similarly, slip-stream transit time effects result from dynamic thrust variations from wing-mounted engines, particularly for propeller aircraft. In this section we investigate the estimation of the transit time lag effects originating from the aforementioned different cases.

A. Separation of Pitch Damping Derivatives

Conventionally, as a first-order approximation, the downwash angle is assumed to be proportional to the wing angle of attack and is represented as a constant derivative $\partial\varepsilon/\partial\alpha$. The so-called downwash lag effect can be mathematically expressed as $(\partial\varepsilon/\partial\alpha)\alpha(t - \tau)$. In the context of estimation of aerodynamic derivatives from flight data, two approaches are possible to account for the downwash lag effect.¹¹

The first approach is based on a linearized aerodynamic model referred to a single point, in which the downwash lag is approximated through derivatives with respect to the translational acceleration in the vertical direction.¹⁰ The derivatives due to the body-fixed translational acceleration in the vertical direction are equivalent in the stability axis to those with respect to $\dot{\alpha}$, the rate of change of angle of attack. The Taylor series expansions of the lift force and pitching moment result in an aerodynamic model that is linear in parameters. Typically,

$$C_L = C_{L0} + C_{L\alpha}\alpha + C_{L\delta_e}\delta_e + C_{Lq}\frac{q\bar{c}}{V} + C_{L\dot{\alpha}}\frac{\dot{\alpha}\bar{c}}{V} \quad (12.1)$$

$$C_m = C_{m0} + C_{m\alpha}\alpha + C_{m\delta_e}\delta_e + C_{mq}\frac{q\bar{c}}{V} + C_{m\dot{\alpha}}\frac{\dot{\alpha}\bar{c}}{V} \quad (12.2)$$

Focusing on the pitching moment, the derivatives C_{mq} and $C_{m\dot{\alpha}}$ contribute to the pitch damping. In several applications, particularly those related to flight data analysis, $C_{m\dot{\alpha}}$ is usually treated as a part of the total equivalent pitch damping $C_{mqEqv} = C_{mq} + C_{m\dot{\alpha}}$. However, in a few specific cases the separation of these two components is desirable, because although in some cases they are closely related, they result from two uncorrelated aerodynamic phenomena. The downwash lag $C_{m\dot{\alpha}}$ is related to variations in the flow pattern generated by the wing, whereas C_{mq} comes dominantly from change in tail angle of attack due to pitching.

Many of the investigations pertaining to aircraft parameter estimation were necessarily restricted to models linear in parameters, mainly because the available estimation programs were capable of handling only the linear models. Although algorithmically it is feasible to estimate the derivatives $C_{L\dot{\alpha}}$ and $C_{m\dot{\alpha}}$, the linearization results in a correlation and affects the independent estimation of these derivatives. In a typical flight maneuver, for example elevator doublet or multistep input, employed in estimating longitudinal derivatives, the pitch rate q is highly correlated to $\dot{\alpha}$ (see Fig. 12.1). This is clear from the $\dot{\alpha}$ equation¹⁰

$$\ddot{\alpha} = -\frac{\bar{q}S}{mV} C_L + q + \frac{g}{V} (\cos \theta \cos \phi \cos \alpha + \sin \theta \sin \alpha) - \tan \beta (p \cos \alpha + r \sin \alpha) \quad (12.3)$$

In a conventional longitudinal maneuver, variations in the roll angle ϕ are minimal. Since the lift coefficient C_L is a linear function of α , q , and δ_e , the first two terms in Eq. (12.3) yield linearly dependent $\dot{\alpha}$ components. To overcome this linear dependence, special flight maneuvers are necessary. An independent component of $\dot{\alpha}$ can be generated through the gravity term, that is, the third term on the right-hand side of Eq. (12.3). The magnitude of the contribution of this term depends on the excursion in the pitch angle θ or the roll angle ϕ . In addition, simultaneous variation in the airspeed V can be used to augment the contribution. Figure 12.2 clearly indicates that a roll maneuver removes the correlation between the variables q and $\dot{\alpha}$. The variable $\dot{\alpha}$ is not directly measured in flight but is obtained by numerical differentiation only for the qualitative comparison.

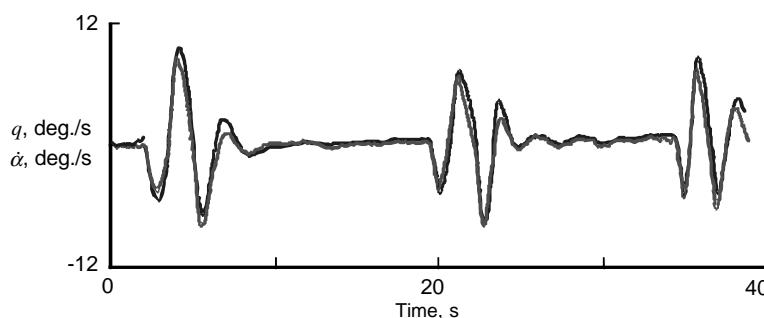


Fig. 12.1 Correlation between pitch rate q and rate of change of angle of attack $\dot{\alpha}$ (source: Ref. 11).

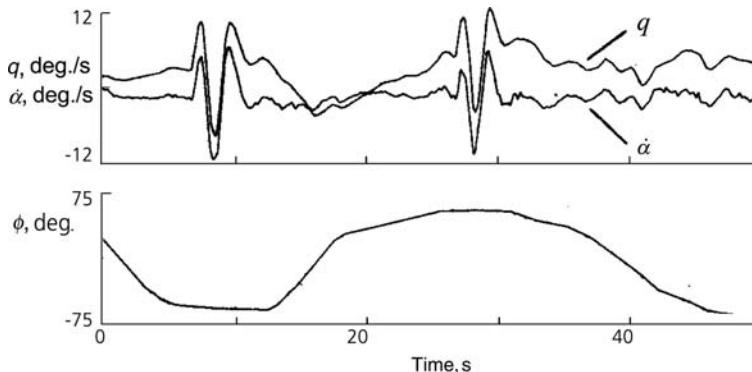


Fig. 12.2 Uncorrelated pitch rate and rate of change of angle of attack from roll maneuver (source: Ref. 11).

For large aircraft, large pitch angle excursions may pose practical difficulties. On the other hand, a roll maneuver is easy to carry out and also provides the necessary separation of q and $\dot{\alpha}$. Reference 10 demonstrates successful estimation of the two pitch damping derivatives from a 180 deg. (inverted position) roll maneuver with a highly maneuverable aircraft, indicating limitations of performing such extreme maneuvers with large aircraft. Yet another restriction of the above linearized model pertains to implementation. Equation (12.1) may lead to an algebraic loop, because computation of C_L appearing in the \dot{w} equation requires $\dot{\alpha}$ (or equivalently \dot{w}).

The second approach is based on the multipoint modeling of the aerodynamic effects. For a conventional aircraft with wing and horizontal tail, the lift and drag forces can be assumed to be working at two points, namely the neutral points of the wing and of the tail. Knowing the forces, the moments can be computed readily, since the lever arms are generally known. In a general case, the lift and pitching moment acting on an aircraft can be represented as¹²

$$C_L = C_{L0} + C_{L\alpha W} \alpha(t) + \frac{S_H}{S} \times \left\{ C_{LaH} \left[\alpha(t) - \frac{\partial \epsilon}{\partial \alpha} \alpha(t - \tau) + i_H + \alpha_{dyn} \right] + C_{L\delta e} \delta_e \right\} \quad (12.4)$$

$$C_m = C_{m0} - \frac{r_H^* S_H}{\bar{c}} \times \left\{ C_{LaH} \left[\alpha(t) - \frac{\partial \epsilon}{\partial \alpha} \alpha(t - \tau) + i_H + \alpha_{dyn} \right] + C_{L\delta e} \delta_e \right\} + C_{mqW} \frac{qc}{V} \quad (12.5)$$

where the subscript W refers to the wing/body combination and H to the horizontal tail; S and S_H denote the wing and tail areas, \bar{c} is the mean aerodynamic chord,

δ_e the elevator deflection, i_H the tail incidence angle. The dynamic angle of attack due to pitch rate is given by $\alpha_{dyn} = \tan^{-1}(qr_H/V)$. The lever arms r_H and r_H^* are, respectively, the distances from the center of gravity and from the neutral point of the wing to the neutral point of the tail. For the sake of clarity, only α is shown to be a function of time, although the other variables are also time-dependent. If vertical wind is to be accounted for, then the expression within the square brackets on the right-hand sides of these two equations will have to be extended appropriately to include the angle of attack due to gust at wing and at horizontal tail, accounting for the transit time effect.

The two-point aerodynamic model necessarily leads to a more complex nonlinear-in-parameters model. The use of such a model to estimate the lift derivatives separately for the wing and tail requires an estimation program capable of handling nonlinear system models as well as a provision to generate delay in a specified variable, in the present case α . Both of these options were available in the estimation program used in these investigations.

Equivalent linear derivatives, if required, say, for comparison purposes, may be obtained from the estimated lift curve slope of the tailplane $C_{L\alpha H}$ and the downwash parameter $\partial\varepsilon/\partial\alpha$ by simple recomputations. For example,

$$C_{mq} = - \left(C_{L\alpha H} \frac{S_H}{S} \frac{r_H^*}{\bar{c}} \right) \frac{r_H}{\bar{c}} + C_{mqW} \quad (12.6)$$

$$C_{m\dot{\alpha}} = - \left(C_{L\alpha H} \frac{S_H}{S} \frac{r_H^*}{\bar{c}} \frac{\partial\varepsilon}{\partial\alpha} \right) \frac{r_H}{\bar{c}} \quad (12.7)$$

The test aircraft VFW-614 ATTAS (Advanced Technologies Testing Aircraft System) is a medium-size twin-jet short-haul 44-passenger aircraft (Fig. 12.3), modified and instrumented for the research purposes at DLR. It also serves as an in-flight simulator.^{13,14} The aircraft with a span of 21.5 m, wing area of 64 m², and full-fuel takeoff weight in the range of 20 tons, has maximum roll



Fig. 12.3 VFW-614 ATTAS.

angle capability up to roughly 70 deg., and the maximum safe load factor is 2.8 g for retracted landing flaps. To investigate estimation of C_{mq} and $C_{m\dot{\alpha}}$, a flight test program was carried out at an altitude of 5000 m and indicated airspeed of 200 kts. Apart from the conventional maneuvers to excite the longitudinal and lateral-directional motion with rapidly changing 3–2–1–1 multistep inputs, five roll maneuvers were carried out. The first four roll maneuvers were with bank angles of 20, 40, 50, and 60 deg., respectively. In these maneuvers, the variations in airspeed were minimal. In the fifth maneuver, the 60 deg. roll is combined with acceleration and deceleration, that is, airspeed variation. Elevator doublets were also applied in the banked position to maximize the information. During the complete maneuver, one must ensure that the load factor is within the allowable safe limits. Although the conventional multistep inputs were generated from the onboard computer and applied automatically, roll maneuvers requiring more coordination and continuous monitoring were carried out manually by the pilot.

The onboard recorded flight data was analyzed off-line using a maximum-likelihood estimation program for nonlinear systems. Although a detailed analysis of all of the test maneuvers has been carried out, only typical results sufficient to demonstrate the estimation of downwash effects are presented here. Five roll maneuvers were analyzed separately to find the effect of the roll angle on the separation of C_{mq} and $C_{m\dot{\alpha}}$. In each case, the roll maneuver is combined with conventional multistep input maneuvers, exciting separately the longitudinal and lateral-directional motion.

The results based on the linearized model of Eqs. (12.1) and (12.2) indicate that it was not possible to estimate separately the derivatives C_{mq} and $C_{m\dot{\alpha}}$ from the roll maneuver limited to 20 deg. bank angle. The other roll maneuvers with more than 40 deg. bank angle provided uncorrelated estimates of these two derivatives, which appeared physically meaningful and also compared well with wind-tunnel predictions. The estimates obtained from the accelerated and decelerated 60 deg. roll maneuver are provided in Table 12.1. In Fig. 12.4 showing two maneuvers demarcated by the vertical lines, the agreement between the measured and estimated time responses of typical motion variables is seen to be good. The first segment corresponds to the multistep elevator input and the second to the roll maneuver with superimposed elevator doublet input.

The analysis has been repeated using the two-point aerodynamic model, Eqs. (12.4) and (12.5). Although this approach does not necessarily require special

Table 12.1 Pitching moment derivatives

Parameter	Wind-tunnel prediction	Estimates from	
		Linearized model	Two-point model ^a
$C_{m\dot{\alpha}}$	-1.07	-1.18	-1.12
$C_{m\dot{\delta}\dot{e}}$	-1.40	-1.66	-1.68
C_{mq}	-8.67	-7.86	-9.03
$C_{m\ddot{\alpha}}$	-4.65	-5.23	-4.59

^aRecomputed estimates.

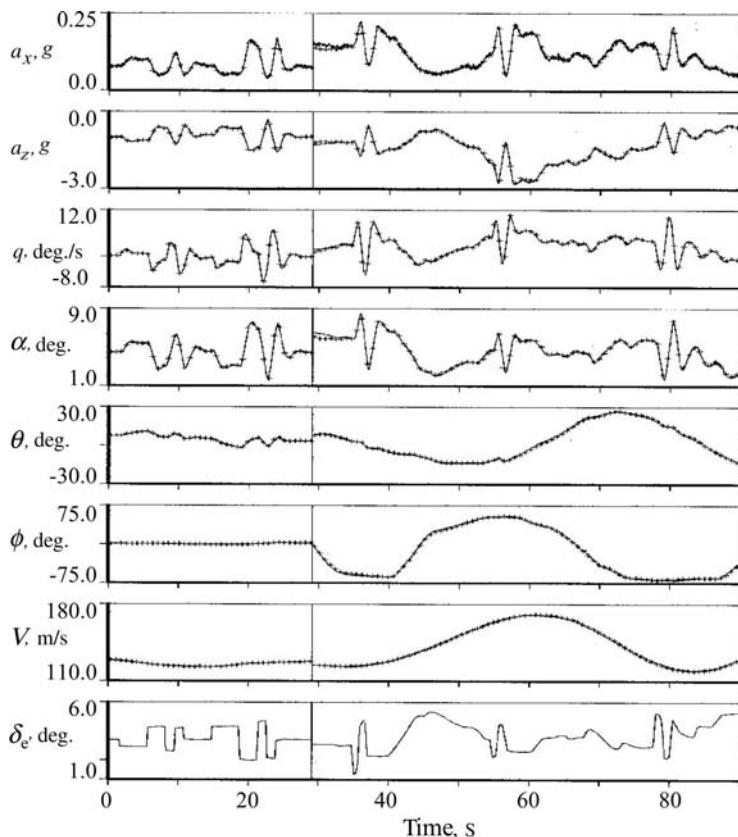


Fig. 12.4 Elevator input and roll maneuvers (—, flight measured; +++, estimated).

flight tests, in order to compare the results with the linearized model, the same set of time records including a roll maneuver is analyzed. Since the downwash effect is implicitly modeled, the roll maneuver with only a 20 deg. bank angle provided estimates of the downwash parameter $\partial e / \partial \alpha$, which agreed well with the estimates from the other roll maneuvers with larger bank angle. From the estimated $C_{L\alpha H}$ and $\partial e / \partial \alpha$, the equivalent linear pitch damping derivatives C_{mq} and $C_{m\dot{\alpha}}$ are computed, using Eqs. (12.6) and (12.7). These values compare well with the estimates from the linearized model and wind-tunnel predictions (see Table 12.1). The match between the flight-measured and estimated responses was essentially the same as in Fig. 12.4. Table 12.1 also provides other pitching moment derivatives. The elevator effectiveness, $C_{m\delta e}$, as estimated by both of the models, is found to be a little higher than the wind-tunnel prediction.

To summarize, the investigations clearly show that the pitching moment derivatives estimated from flight data by the two complementary methods and wind-tunnel predictions agreed well.¹¹ The linearized approach based on the

Taylor series requires special flight maneuvers to remove the correlation between the motion variables. The roll maneuvers with more than 40 deg. bank angle yielded uncorrelated estimates of the two pitch damping derivatives. The two-point aerodynamic model, based on the forces acting on the wing and tail, accounts explicitly for the downwash lag effect through a constant derivative and time delay. It requires, however, a more advanced parameter estimation program capable of handling general nonlinear systems, but eliminates the need for special flight maneuvers otherwise necessary to remove correlation between the motion variables. The second approach is more sophisticated for two reasons:

- (1) It provides a physically more realistic representation of the aerodynamics and accounts for the downwash effect explicitly through the derivatives $\partial \varepsilon / \partial \alpha$ and a time delay τ .
- (2) It provides an alternative to account for the downwash effects for large aircraft with limited roll angle capabilities, which we had briefly indicated earlier in this section.

In an analogous treatment of wing/body undergoing lateral perturbations, the resulting forces on the fin lead to a derivative with respect to the rate of change of angle-of-sideslip $\dot{\beta}$. This derivative, modeling the sidewash effect, contributes to yaw damping. As in the case of longitudinal motion, many applications consider the total equivalent yaw damping $C_{nrEqv} = C_{nr} + C_{n\dot{\beta}}$. Estimation of $C_{n\dot{\beta}}$ from flight test maneuvers is slightly easier than that of $C_{m\dot{\alpha}}$. The classical yawing and rolling maneuvers, which are performed anyway to allow determination of lateral-directional derivatives, contain a small $\dot{\beta}$ component independent of other variables. This is evident from the following equation:

$$\dot{\beta} = \frac{\bar{q}S}{mV} C_Y + p \sin \alpha - r \cos \alpha + \frac{g}{V} \sin \phi \cos \theta \quad (12.8)$$

The flight investigations with ATTAS and the following parameter estimation confirmed that the classical rapid roll and Dutch roll maneuvers did lead to consistent and reliable estimates of the sidewash derivative. A comparison with the wind-tunnel predictions (see the far right side plot of Fig. 12.5) clearly show that the prediction had neglected the contribution due to rate of change of angle of sideslip. This was further confirmed by estimating an equivalent yaw damping derivative C_{nrEqv} from flight maneuvers, yielding a value between -0.33 and -0.28 , and not close to the predicted -0.6 . On the other hand, separate estimation of the two provided values of -0.568 and -0.281 for C_{nr} and $C_{n\dot{\beta}}$ respectively; the difference between the two roughly matching with the equivalent derivative obtained from flight (note that the r and $\dot{\beta}$ are opposite in phase). The flight estimation of the two components of yaw damping resolved the discrepancies observed in the performance of the model-following control that was optimized on the ground using predicted database and tested in flight.

Figure 12.5 shows a comparison of the other damping derivatives as well. It is observed that the roll damping $C_{\ell p}$ including its dependence on the angle of

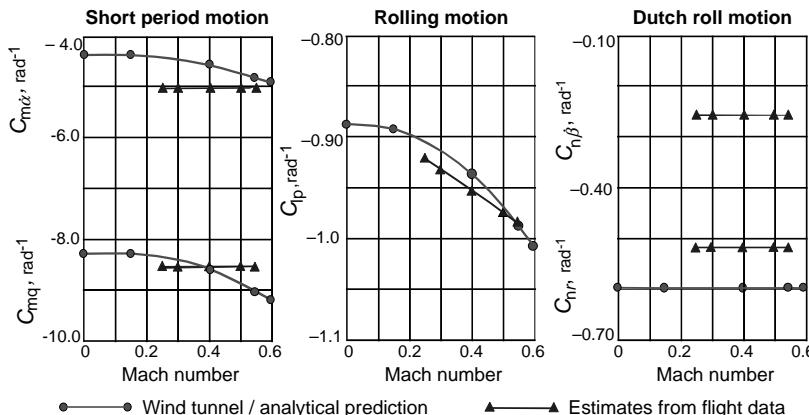


Fig. 12.5 ATTAS damping derivatives.

attack was estimated best, matching very well with the predictions. The estimates of the pitch damping derivatives C_{mq} and $C_{m\dot{\alpha}}$ from two sources were also comparable. The estimates of the dependence of these two derivatives on the Mach number were found to be poor and correlated. Furthermore, their inclusion was also not justified from the quality of the match for all the output variables. Hence, they are shown as constant values in Fig. 12.5. However, we know from theory that such dependencies do exist in practice, and, hence, some fairing of results between the predictions and flight estimates would be justifiable.

B. Aerodynamic Effects of Direct Lift Control Flaps

As already pointed out, the VFW-614 ATTAS, Fig. 12.3, is a fly-by-wire aircraft used for in-flight simulation (i.e., simulation of other aircraft in flight). One of the major features of ATTAS is that, in addition to the conventional controls, it is equipped with direct lift control (DLC) flaps for high-frequency direct-lift modulation.^{13–15} They provide an additional independent longitudinal control, which is particularly suitable and necessary for realistic in-flight simulations. In such applications, the feed-forward control laws based on the inversion of the mathematical model of the “host” flight vehicle play an extremely important role. This is because high-bandwidth control systems need more accurate regulation of initial response than of final output error. The fidelity of the in-flight simulator depends to a large extent on the accuracy of the aerodynamic database representing the host aircraft, and as such the demands on the modeling accuracy are very high. Accordingly, system identification of ATTAS has been carried out, including the rigid-body model based on wing and tail separately, modeling DLC, control surface interference, ground effects, and modeling of actuation systems of primary controls.^{14,15}

The DLC flaps are derived by modifying a trailing edge portion (45%) of the landing flaps; see Fig. 12.6. They are divided into six fast moving flaps, three on each wing. Each of the six flaps is driven by an individual electrohydraulic

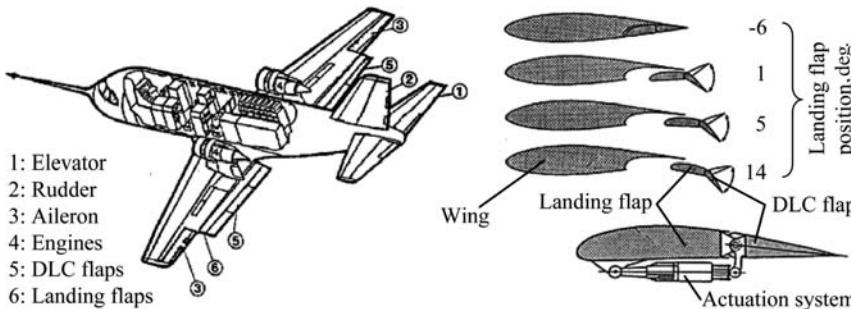


Fig. 12.6 Research aircraft ATTAS and direct lift control flaps.

actuation system which is both rate- and force-limited for structural safety. The six flaps are designed for a maximum deflection of ± 35 deg. and for an actuation rate of 75 deg./s under aerodynamic loads. The DLC flaps can be operated for landing-flap deflections up to 14 deg. Simultaneous variations of the DLC and landing flaps are possible.

For the purpose of aerodynamic modeling, the six DLC flaps are considered in two units, each consisting of three flaps on each wing. The equivalent left and right DLC-flap deflections, denoted by $\delta_{\text{DLC}}^{\text{L}}$ and $\delta_{\text{DLC}}^{\text{R}}$, are computed from the individually measured flap deflections proportional to the surface areas. The lift, drag, and pitching moment generated by the DLC flaps are modeled as¹⁵

$$\begin{aligned} C_{\text{LDLC}} &= \Delta C_{\text{LDLC}}^{\text{L}} + \Delta C_{\text{LDLC}}^{\text{R}} \\ C_{\text{DDLC}} &= \Delta C_{\text{DDLC}}^{\text{L}} + \Delta C_{\text{DDLC}}^{\text{R}} \\ C_{\text{mDLC}} &= \Delta C_{\text{mDLC}}^{\text{L}} + \Delta C_{\text{mDLC}}^{\text{R}} + \Delta C_{\text{mDLC}}^{\varepsilon} \end{aligned} \quad (12.9)$$

where the contributions due to the left and right DLC flaps are denoted by the superscripts L and R, respectively. In the above equation we have considered contributions due to the left and right DLC flaps separately, because it was found that the lift and drag differentials resulting from unequal DLC flap deflections on the two sides lead to non-negligible rolling and yawing motion.^{14,15} We restrict ourselves here to discussion of the longitudinal motion only. The following general model for these contributions is postulated:

$$\begin{aligned} \Delta C_{\text{LDLC}} &= A_1 \delta_{\text{DLC}} + A_2 \delta_{\text{DLC}}^2 + A_3 \delta_{\text{DLC}}^3 \\ \Delta C_{\text{DDLC}} &= (W_1 + W_{1\alpha}\alpha) \delta_{\text{DLC}} + W_2 \delta_{\text{DLC}}^2 \\ \Delta C_{\text{mDLC}} &= M_1 \delta_{\text{DLC}} + M_2 \delta_{\text{DLC}}^2 + M_3 \delta_{\text{DLC}}^3 \end{aligned} \quad (12.10)$$

where A_i , W_i , and M_i are the unknown derivatives that are to be estimated from flight data. Furthermore, in Eq. (12.9), $\Delta C_{\text{mDLC}}^{\varepsilon}$ denotes the downwash lag effect

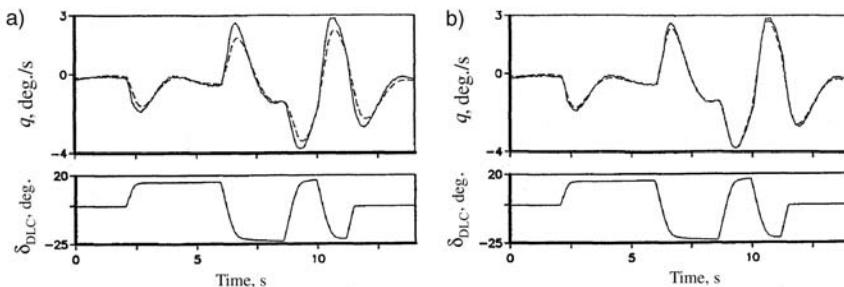


Fig. 12.7 Pitch rate in response to DLC flap input (—, measured; - - -, estimated). a) Neglecting the downwash lag, b) accounting for downwash lag.

due to deflection of the DLC flaps. Neglecting this term for the downwash, estimation of the derivatives appearing in Eq. (12.10) yields the pitch rate response due to a multistep 3–2–1–1 input to the DLC flaps shown in Fig. 12.7a. It is observed that the match between the flight measured and estimated response for steady-state level is acceptable, but that for dynamic changes from one steady-state level to another shows larger deviations.

By analogy to the translational acceleration derivatives derived from an approximation of the unsteady aerodynamic effect (in particular, the derivative $C_{m\dot{\alpha}}$ modeling the downwash lag due to the variations in the angle of attack), in the present case the downwash lag due to a DLC flap deflection can be approximated through a first-order linear pitching moment derivative with respect to rate of change of DLC flap deflection $\dot{\delta}_{DLC}$. This would call for measurement of $\dot{\delta}_{DLC}$ or computation of the same from measured δ_{DLC} by numerical differentiation. However, as discussed in Sec. II.A, we adopt here the more elegant approach based on a two-point aerodynamic model incorporating a time-delayed signal in parameter estimation. Accordingly, the downwash lag effect is modeled as

$$\Delta C_{mDLC}^e = C_{LaH} \frac{S_H r_H^*}{S} M_\tau \delta_{DLC}(t - \tau) \quad (12.11)$$

where C_{LaH} is the tail lift curve slope, M_τ the unknown downwash parameter and $\delta_{DLC}(t - \tau)$ the time delayed signal, where $\tau = r_H/V$ denotes the transit time required for flow modifications generated at the wing to reach the tail; other variables have been defined previously. Estimation of parameters appearing in Eqs. (12.10) and (12.11) yields the match for the pitch rate shown in Fig. 12.7b. The improvements in the match in Fig. 12.7b compared with 12.7a are, without any further explanation, self-illustrative of the downwash lag effect and efficacy of the modeling approach. Thus, the cubic model for the pitching moment due to DLC flaps in Eq. (12.10) needs to be augmented with Eq. (12.11) for the downwash lag effect.

The estimates of the DLC flap effectiveness are shown in Fig. 12.8. It is observed that the estimated lift coefficient C_{LDLC} is smaller than predicted, the

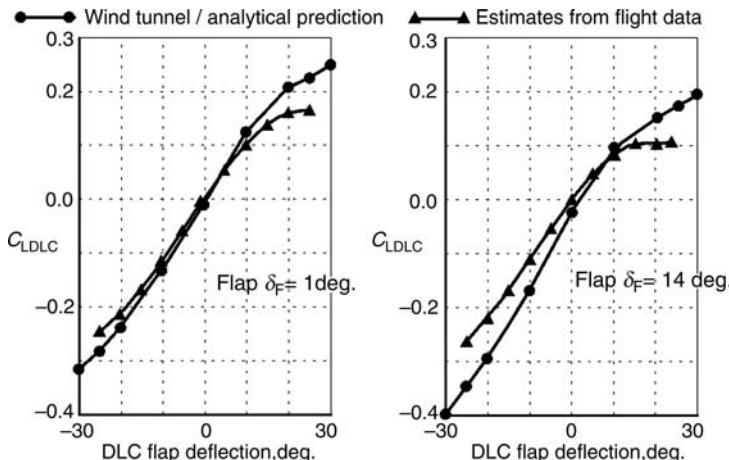


Fig. 12.8 DLC flaps effectiveness: static characteristics.

difference being pronounced for larger positive deflections. The influence of flow separation for deflections above 10 deg. is to be observed. It is evident that the DLC flaps are somewhat less effective in generating direct-lift than designed for and predicted from wind-tunnel tests. The flight validated static aerodynamic characteristics together with the downwash transient effects provided a high fidelity model for the in-flight simulation.

C. Modeling of Speed Brake Effect

The “Transall” C-160 is a military transport aircraft capable of carrying troops, casualties, freight, supplies, and vehicles, and serves the needs of the German Air Force; see Fig. 12.9. It is powered by twin turboprop Rolls Royce Tyne engines with four-bladed propellers. Each engine is capable of generating 70,000 N thrust. The maximum takeoff and landing weight is 51,000 kg, and the maximum cruising speed at an altitude of 16,000 ft is 277 kts (513 km/h). The cantilever high-wing has an area of 160.1 m² with a span of 40 m. The overall length is 32.4 m. The horizontal tail area is 43.8 m². Apart from the primary control surfaces, that is, elevator, rudder, and ailerons, the C-160 is equipped with hydraulically operated air-brakes mounted inboard on the wings and spoilers mounted outboard forward of the flaps on each wing; see Fig. 12.10. The spoilers are designed to improve the aileron effectiveness and are deflected proportionally for aileron deflections greater than 3 deg. and less than 10 deg. For aileron deflections greater than 10 deg. the spoilers are deflected to the maximum limit of 45 deg. The underside of the upswept rear fuselage lowers to form a loading ramp. The tricycle landing gear is retractable. Each main unit comprises two pairs of wheels in tandem and the twin-wheel nose unit is steerable.

As a part of a comprehensive data gathering and aerodynamic database development program for a training simulator,^{16–18} which will be addressed in



Fig. 12.9 Transall C-160.

Sec. VIII, static and dynamic effects for all operational configurations including all primary and secondary control surfaces were modeled and estimated from flight data. In the present section, we concentrate on a single aspect, namely speed brakes. The aerodynamic effects due to speed brakes are modeled as incremental changes in the lift, drag, and pitching moment coefficients, that is, as ΔC_{LSB} , ΔC_{DSB} , and ΔC_{mSB} the subscript SB denoting speed brakes. Flight tests during which, starting from trimmed level flight, 40% speed brakes were applied are taken as a typical case for evaluation. Based on the aforementioned comprehensive aerodynamic model, the above incremental effects are identified separately for three landing flap positions, namely $\delta_F = 0, 30$, and 60 deg. This identification yields the estimated responses shown by dashed lines in Fig. 12.11, which are compared with the flight measurements shown by solid lines, where a_x

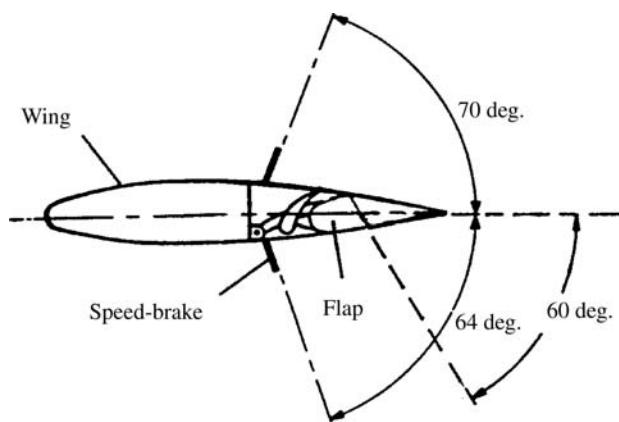


Fig. 12.10 Speed brake and flap configuration.

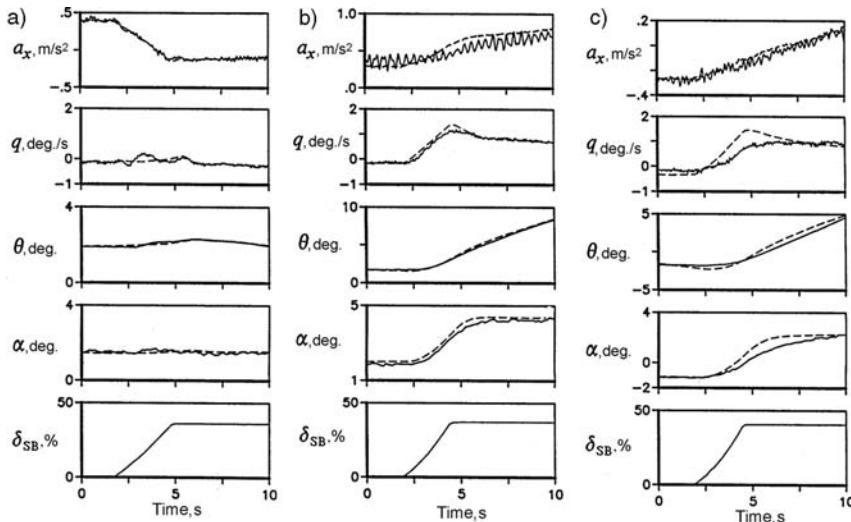


Fig. 12.11 Identification of aerodynamic effects caused by speed brakes (—, flight measured; - - - estimated). Landing flap = a) 0, b) 30, and c) 60 deg.

is the longitudinal acceleration, q the pitch rate, θ the pitch angle, α the angle of attack, and δ_{SB} the speed brake deflection.¹⁸ The braking effect is clearly observed from the reduction in the longitudinal acceleration a_x for $\delta_F = 0$ deg.; see Fig. 12.11a. Variations in the pitching motion as seen from pitch rate q are negligible. In the other two cases of 30 and 60 deg. flaps (Fig. 12.11b and c, respectively), the braking effect is, of course, present, but not directly visible in a_x due to pitch up. The speed brakes, as seen in Fig. 12.10, consist of lower and upper flaps on each wing and are deflected symmetrically. As such, they primarily work in a classical sense as a drag inducing device.

A closer look at the response match for $\delta_F = 30$ and 60 deg. in Fig. 12.11, however, shows some discrepancies. A time lag between the flight measurements and the estimated responses is clearly to be seen, this time lag increasing with the increasing flap deflection. As in the previous two cases presented in Secs. XII.A and XII.B, these discrepancies are modeled as a downwash generated by the speed brakes. Accounting for the transit time effect, the angle of attack α_H at the tail is now given by¹⁸

$$\begin{aligned} \alpha_H = \alpha + i_H - \frac{\partial \varepsilon_H}{\partial \alpha} \alpha(t - \tau_\alpha) + \frac{\partial \varepsilon_H}{\partial C_S} C_S(t - \tau_{CS}) \\ + \frac{\partial \varepsilon_H}{\partial \delta_{SB}} \delta_{SB}(t - \tau_\alpha) + \alpha_{dyn} \end{aligned} \quad (12.12)$$

where α is the angle of attack at the wing, C_S the thrust coefficient, ε_H the downwash angle, and i_H and δ_{SB} have been defined earlier. τ_α and τ_{CS} represent the transit time for flow changes generated at the wing and at the propeller,

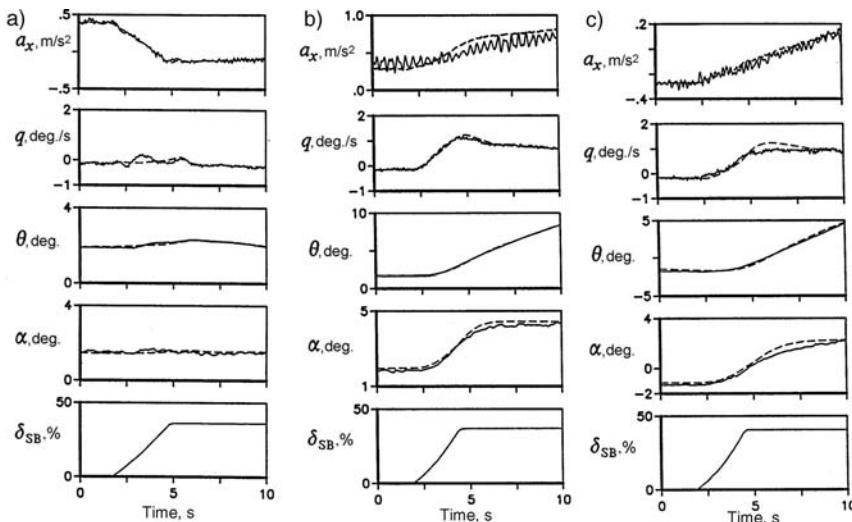


Fig. 12.12 Influence of downwash lag caused by speed brakes (—, flight measured; - - - estimated). Landing flap = a) 0, b) 30, and c) 60 deg.

respectively, to reach the tail. The downwash parameters $\partial\epsilon_H/\partial\alpha$ and $\partial\epsilon_H/\partial C_S$ are known from the foregoing step of basic aerodynamic model identification without speed brakes. The downwash parameter $\partial\epsilon_H/\partial\delta_{SB}$ has to be estimated from flight tests with speed brakes.

The model postulated in Eq. (12.12) yields the responses shown in Fig. 12.12. Compared with Fig. 12.11, the significantly improved match for the pitch rate q , and pitch attitude θ , particularly for the 30 and 60 deg. landing flap, clearly demonstrates the appropriateness of the formulated model. The parameter $\partial\epsilon_H/\partial\delta_{SB}$ was zero for $\delta_F = 0$ deg., as expected from Fig. 12.11. The estimates for the other two flap positions were found to be nearly linearly dependent on the flap deflection.

To summarize, in this section we addressed a specific aerodynamic effect of downwash lag. The phenomenon of downwash is typical for conventional aircraft configurations with wing/body and horizontal tail. For high fidelity databases, it is important to account for these effects induced through dynamic motion as well as through control deflections. The commonality of the chosen modeling technique to account for downwash resulting from different operating conditions and configurations has been presented.

III. Aerodynamic Effects of Landing Gear

In general, determination of primary derivatives from flight test data does not pose any serious difficulties. For high-fidelity databases for flight simulators, however, it is necessary to properly model the aerodynamic effects of secondary order as well. This is particularly true when the model fidelity has to meet the level D requirements.^{19,20} The ramp door and landing gear effects and the

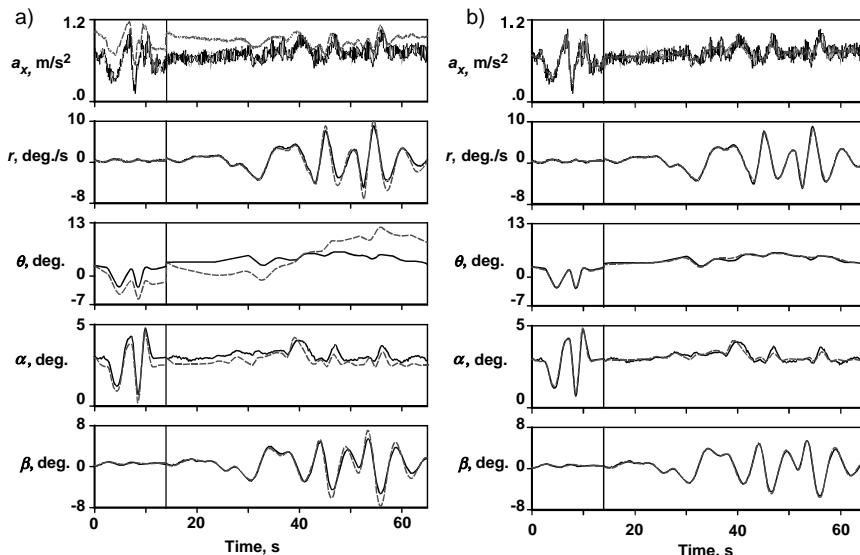


Fig. 12.13 Identification of landing gear effects (—, flight measured; ---, estimated); a) neglecting and b) accounting for landing gear effects.

downwash due to speed brakes are some typical examples of this class; we have already addressed the last one in the preceding section.

To enable estimation of aerodynamic effects due to landing gear, flight maneuvers exciting the longitudinal and lateral-directional motion were carried out with extended landing gear.¹⁸ The investigations covered the take-off and landing configurations. As a typical example, Fig. 12.13 shows two flight maneuvers flown with extended landing gear; the first maneuver is a short period motion excited by a multistep 3–2–1–1 input and the second is an aileron and rudder input maneuver resulting in rolling and yawing motion. Once again, following the general approach adopted in the preceding sections, modeling and identification of aerodynamic effects due to the landing gear are carried out stepwise. Proceeding from the basic aerodynamic model identified for the clean configuration, that is, neglecting the landing gear effects, the flight maneuvers with extended landing gear are analyzed. This yields the response match shown in Fig. 12.13a, where a_x is the longitudinal acceleration, r is the yaw rate, θ is the pitch attitude, α is the angle of attack, and β is the angle of sideslip. The discrepancies in the longitudinal acceleration are demonstrative of additional drag being introduced. Note that, for $t = 0$, the initial pitch attitude matches the measurement, but not the longitudinal acceleration, which is an indicator that the mismatch in a_x is not due to deviations in pitch attitude, but result directly from the increased drag. Additional drag not yet accounted for in the model leads to deviation in the match for other variables, including that for the pitch attitude. The lateral-directional variables, angle of sideslip β and yawing rate r also show some discernible differences.

Through a systematic approach, and keeping in mind the principle of parsimony, additional parameters were identified to model the incremental effects due to the landing gear. The basic aerodynamic model is augmented with additional parameters, namely incremental changes in the lift, drag, and pitching moment coefficients, that is, ΔC_{LLG} , ΔC_{DLG} , and ΔC_{mLG} , the subscript "LG" denoting the landing gear. Furthermore, the derivatives $C_{n\beta LG}$ and $C_{Y\beta LG}$, were introduced to model the influence on the lateral-directional motion. As expected, the identified value of ΔC_{DLG} suggests increased drag. The positive value $C_{n\beta LG}$ indicates increased weathercock stability. The two derivatives $C_{n\beta LG}$ and $C_{Y\beta LG}$ are adequate to model the influence of the landing gear on the lateral-directional motion. Attempts to estimate increased yaw damping due to landing gear did not lead to any improvements in the response match, whereas a destabilizing effect of the landing gear on the pitching motion, as indicated by a positive value of the pitch damping derivative C_{mqLG} , is identified. Estimation of the aforementioned aerodynamic derivatives from the maneuvers with extended landing gear posed no difficulties. They were estimated with very low standard deviations. The resulting model yields the response match shown in Fig. 12.13b, which is significantly better than that in Fig. 12.13a. The landing gear effects are typically in the range of 5–10% of the total, but still important for simulator fidelity. Proper test techniques and a systematic modeling approach enable accurate identification of such aerodynamic effects of secondary order.²¹

IV. Control Surface Malfunction Effects

Pilot training under abnormal conditions is an important application area of flight simulators. Extreme flight conditions can arise due to several reasons, for example flow separation at high angles of attack leading to stall (which we will address in the next section), due to asymmetric flight with large sideslip angles or due to control surface malfunctions leading to partial loss of control power. Modeling of aircraft behavior under such extreme conditions is complex and validation from flight tests particularly difficult because such configurations are difficult to fly and are safety critical involving some risk. Here, we consider the specific case of roll-spoiler hardover, tested on a Dornier 328 for the landing configuration of 32 deg. flaps. Starting from a horizontal level flight at an altitude of 8700 ft and 115 kts true airspeed (roughly 100 kts CAS), trim angle of attack of about 2.5 deg. with extended landing gear, the roll spoilers are drawn out on the left wing to the hardover limit of 75 deg. and mechanically held fixed. Under normal operating conditions the roll spoilers are coupled to the ailerons; they are deflected from 0 to 45 deg. linearly for aileron deflections between 3 and 25 deg.; for aileron deflections less than 3 deg. the roll spoilers are not active.

As common to the general procedure followed in the aerodynamic model development, first we simulate the aircraft response to flight maneuver inputs using a database developed from flight maneuvers under normal operating conditions, that is, in which aileron and roll spoilers are coupled and deflected proportionately as just described. The comprehensive aerodynamic model included angle of attack and thrust dependencies, as well as nonlinearities in the aileron and roll spoiler effectiveness for deflections up to 25 and 45 deg., respectively.

For example, the contributions to the rolling and yawing moment coefficients due to roll spoiler are given by

$$\begin{aligned}\Delta C_{\ell_RS} &= (C_{\ell RS} + C_{\ell RS3} \delta_{RS}^2 + C_{\ell RS\alpha} \alpha + C_{\ell RSCS} C_S) \delta_{RS} \\ \Delta C_{n_RS} &= (C_{n RS} + C_{n RS\alpha} \alpha + C_{n RSCS} C_S) \delta_{RS}\end{aligned}\quad (12.13)$$

where δ_{RS} is the roll spoiler deflection, C_S the thrust coefficient, and $C_{\ell(\cdot)}$ and $C_{n(\cdot)}$ the respective derivatives. The left and right spoilers are treated separately.

Applying such a model to the roll-spoiler hardover maneuver leads to the simulated responses shown by dashed lines in Fig. 12.14, showing the translational and rotational accelerations on the left-hand side, and the flow angles, bank angle, aileron, spoiler, and rudder deflections on the right-hand side. From the second plot from bottom on the right-hand side showing the roll spoiler deflection, we notice that after 8 s the left roll spoiler is jammed at the maximum limit of 75 deg. We observe that both the left and right roll spoilers are deflected, which is in contrast to the normal conditions under which, depending upon the aileron deflection, only one of the roll spoilers either on the left or right is deflected. The simulated aircraft response in Fig. 12.14 shows drastic deviations compared with the flight measured data shown by continuous lines. The trim flight prior to the control surface jam is well simulated without showing any tendency to drift or misbehavior otherwise, but after the malfunction starts at 8 s the simulated responses are unable to predict the aircraft behavior.

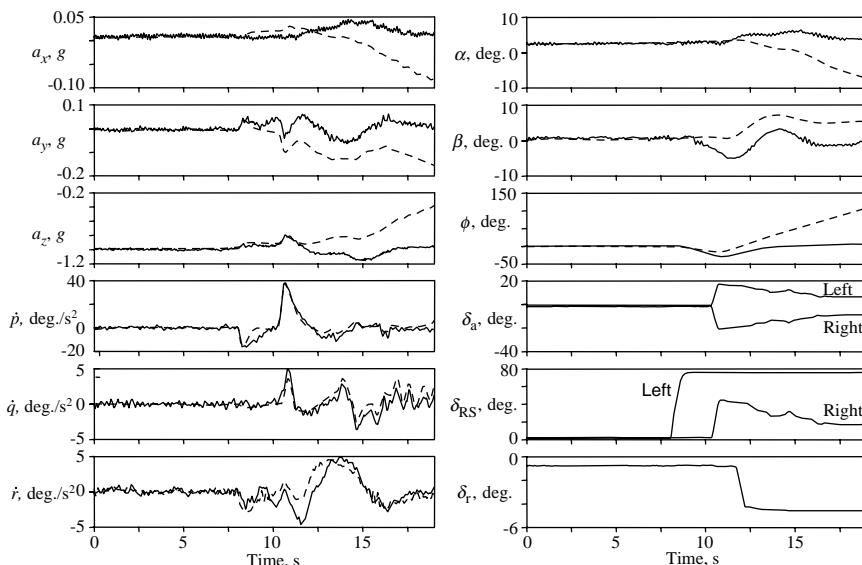


Fig. 12.14 Simulation of roll spoiler hardover using basic model (—, flight measured, -----, model predicted).

The roll spoiler hardover affects primarily the rolling and yawing motion, and leads to increased drag. Through a systematic approach, the discrepancies observed in Fig. 12.14 are traced to strong nonlinearities in the roll spoiler effectiveness. The comprehensive model is now updated through simple correction terms for roll spoiler deflections greater than 45 deg. The contributions to the rolling and yawing moment coefficients due to roll spoiler in Eq. (12.13) are now augmented with corrections given by

$$\begin{aligned}\Delta C_{\ell_RSNL} &= \begin{cases} C_{\ell RSNL}(\delta_{RS} - \delta_{RSBP}), & \text{for } \delta_{RS} > \delta_{RSBP} \\ 0, & \text{for } \delta_{RS} \leq \delta_{RSBP} \end{cases} \\ \Delta C_{n_RSNL} &= \begin{cases} C_{n RSNL}(\delta_{RS} - \delta_{RSBP}), & \text{for } \delta_{RS} > \delta_{RSBP} \\ 0, & \text{for } \delta_{RS} \leq \delta_{RSBP} \end{cases}\end{aligned}\quad (12.14)$$

In addition, it was found that the drag and pitching moment coefficients had to be modified as follows to include the effect due to roll-spoiler hardover.

$$\begin{aligned}\Delta C_{D_RSNL} &= \begin{cases} C_{DRSNL}(\delta_{RS} - \delta_{RSBP}), & \text{for } \delta_{RS} > \delta_{RSBP} \\ 0, & \text{for } \delta_{RS} \leq \delta_{RSBP} \end{cases} \\ \Delta C_{m_RSNL} &= \begin{cases} C_{mRSNL}(\delta_{RS} - \delta_{RSBP}), & \text{for } \delta_{RS} > \delta_{RSBP} \\ 0, & \text{for } \delta_{RS} \leq \delta_{RSBP} \end{cases}\end{aligned}\quad (12.15)$$

where the equivalent roll spoiler deflection $\delta_{RS} = \delta_{RS}^R + \delta_{RS}^L$ is the sum of the right and left roll spoiler deflections (they come out on one side only, deflected upwards and measured positive), δ_{RSBP} is the so-called breakpoint and $C_{\ell RSNL}$, $C_{n RSNL}$, C_{DRSNL} , and C_{mRSNL} are the linear derivatives representing corrections for roll spoiler deflections greater than δ_{RSBP} , taken to be 45 deg. The choice of 45 deg. as breakpoint is based on the fact that the basic model was estimated and validated from flight maneuvers covering deflections up to this range.

Estimation of the four additional parameters without affecting the performance of the basic model for deflections less than δ_{RSBP} (i.e., under normal operating conditions) leads to the results shown in Fig. 12.15. Compared with Fig. 12.14, it is now observed that the model-computed responses for all the variables match the flight-measured data very well. Some minor deviations in the match for angle of sideslip β and yaw acceleration r are observed, which are attributed to the simplified linear corrections for the highly complex flow phenomenon under such extreme flight conditions. A careful look at the match for the yaw acceleration r suggests that other effects, possibly sidewash leading to a delayed response, need to be accounted for. Nevertheless, simple model extensions in the rolling, yawing, and drag coefficients and parameter estimation from flight data were adequate to capture the basic aerodynamic phenomenon of roll-spoiler hardover. In the majority of the cases the correct trends in motion in all directions and a reasonable match between model prediction and flight measurements is usually sufficient for simulator pilot training under such abnormal conditions.

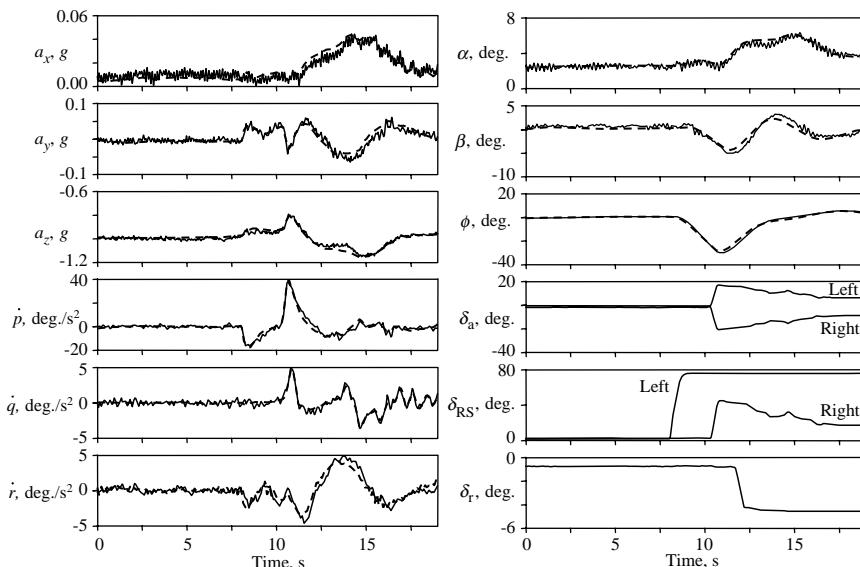


Fig. 12.15 Identification of roll-spoiler hardover (—, flight measured; ----, model predicted).

V. Unsteady Aerodynamics Modeling

Under stationary attached flow conditions, aerodynamic effects can be adequately described using time-invariant parameters and linear models. At higher angles of attack and aircraft undergoing stall, models are highly nonlinear, because flow separation and unsteady effects are dominant. Flow separation results primarily in a reduction of the lift curve slope before the maximum lift point, whereas a significant loss of lift occurs in the post-stall region. At such flight conditions the generated lift is influenced strongly by unsteady effects. Different types of flow separation occur depending upon the airfoil shape or wing configuration. The most common type is the trailing edge separation which is typical for most conventional aircraft having a turbulent boundary layer.

Unsteady aerodynamics has been a subject of extensive investigations using computational fluid dynamic methods, wind-tunnel tests, and semi-empirical models. The common approach has been based on indicial functions.^{22–26} Although such models provide a basis for analytical investigations of the complex flow phenomena, postulating them in an analytical form suitable for parameter estimation is difficult. Therefore, we follow the alternative approach of Refs. 27 and 28 to describe analytically the flow separation including stall hysteresis as a function of an internal state variable. Since the approach retains the state-space formulation, it is directly amenable to identification and validation from flight data, as has been validated on flight data in Ref. 29. For a review of unsteady aerodynamic modeling the reader is referred to Ref. 30.

In this section we draw heavily on the mathematical developments presented in Ref. 28 and on the applications to flight data in Refs. 29 and 31. Trailing edge

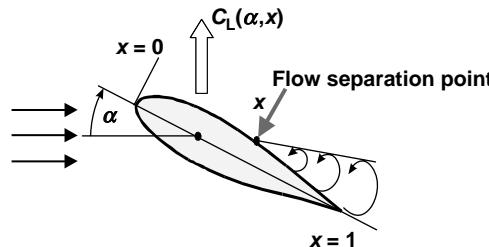


Fig. 12.16 Trailing edge flow separation phenomenon.

stall is a dominant phenomenon on wings having sufficient thickness ($t/\bar{c} > 0.15$; where t represents wing thickness and \bar{c} the wing mean aerodynamic chord), and undergoing a relatively slow variation in the angle of attack ($\dot{\alpha} < 0.02V/\bar{c}$; where $\dot{\alpha}$ is the rate of change of angle of attack and V is the freestream velocity). For such a case, a nondimensional state X is introduced, describing the instantaneous location of an idealized flow separation point along the chord on the upper surface of the wing ($0 \leq X \leq 1$), see Fig. 12.16; $X = 1$ and $X = 0$ correspond to attached and fully separated flow, respectively. Based on Kirchhoff's theory of flow separation, for a symmetrical profile the wing lift can be modeled as a function of angle of attack α and flow separation point X :

$$C_L(\alpha, X) = C_{L\alpha} \left\{ \frac{1 + \sqrt{X}}{2} \right\}^2 \alpha \quad (12.16)$$

where $C_{L\alpha}$ is the lift curve slope.

The position of the flow separation point can be conveniently described using a single ordinary differential equation:

$$\tau_1 \frac{dX}{dt} + X = X_0(\alpha, \tau_2 \dot{\alpha}) \quad (12.17)$$

where τ_1 and τ_2 are the time constants representing the transient and the quasi-steady aerodynamic effects, respectively, and X_0 the separation point under steady conditions which is, as shown on the right-hand side, a nonlinear function of α , $\dot{\alpha}$, and τ_2 . To bring out the hysteresis effect resulting from τ_2 , for typical values, we plot in Fig. (12.17) the flow separation point $X(\alpha, \tau_2 \dot{\alpha})$ and the corresponding lift coefficient $C_L[\alpha, X(\alpha, \tau_2 \dot{\alpha})]$. The solid lines correspond to steady conditions of $\dot{\alpha} = 0$ and the dashed lines show X and C_L for different but constant rates $\dot{\alpha}$. For a positive $\dot{\alpha}$, flow separation occurs at higher α , whereas for negative $\dot{\alpha}$ the flow reattachment is delayed to α smaller than that of the steady case. The resulting delay (hysteresis width) is proportional to $\dot{\alpha}$ and $\tau_2 \bar{c}/V$. We obtain an appreciation of the influence of these parameters by varying them and running the program /FVSysID/chapter12/HystTau2.m.

The steady flow-separation point X_0 depends upon the airfoil and wing configuration. Using Eq. (12.16) with $X = X_0$, the function can be determined

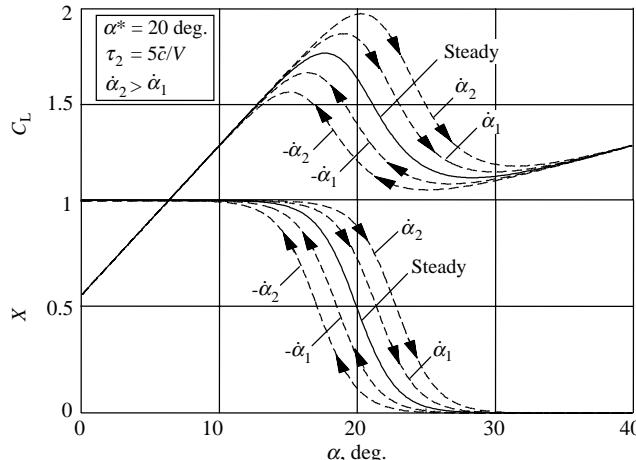


Fig. 12.17 Quasi-steady lift coefficient and flow separation point.

statically in a wind-tunnel. An approximation based on two exponential functions, which can be adapted with several parameters, is used in Ref. 32. We use here an alternative approximation based on hyperbolic-tangent investigated in Refs. 29 and 31.

$$X_0 = \frac{1}{2} \left\{ 1 - \tanh[a_1(\alpha - \alpha^*)] \right\} \quad (12.18)$$

where a_1 defines the static stall characteristics of the airfoil, and α^* the breakpoint corresponding to $X_0 = 0.5$. This approximation is better suited to parameter estimation, because it is a continuous function in its entire range and has just two unknown parameters, namely a_1 and α^* . The ease of adapting Eq. (12.18) to different airfoils and configurations is apparent from Fig. 12.18; showing X for various values of a_1 for a fixed value of $\alpha^* = 15$ deg. in Fig. 12.18a and for various values of α^* for a fixed value of $a_1 = 20$ in Fig. 12.18b. As seen from this figure, it is obvious that the function represented in Eq. (12.18) has its turning point at α^* . Varying α^* would move the function, that is, X_0 , horizontally. The lift computed from Eq. (12.16) using $X = X_0$ is also shown in the same figure.

The applicability of the proposed hyperbolic-tangent function to approximate the steady flow separation is demonstrated on wind-tunnel measurements on an OA213 rotor airfoil. Since the flow separation point was not measured in the wind-tunnel tests, it was computed point wise using the reformulated Kirchoff's formulation of flow separated lift, Eq. (12.16), extended by C_{L0} for a nonsymmetrical profile, and solving for X , that is, $X_0 = \{2\sqrt{[(C_L - C_{L0})/(C_{L0}\alpha)]} - 1\}^2$. Figure 12.19 shows the wind-tunnel-derived separation points and the lift curve. It also shows the approximation of the wind-tunnel data using Eqs. (12.16) and (12.18). The two unknown parameters in Eq. (12.18) turn out to be $\alpha^* = 15.2$ deg. and $a_1 = 17.8$. From the plot of X_0 , it is observed that

SELECTED ADVANCED EXAMPLES

417

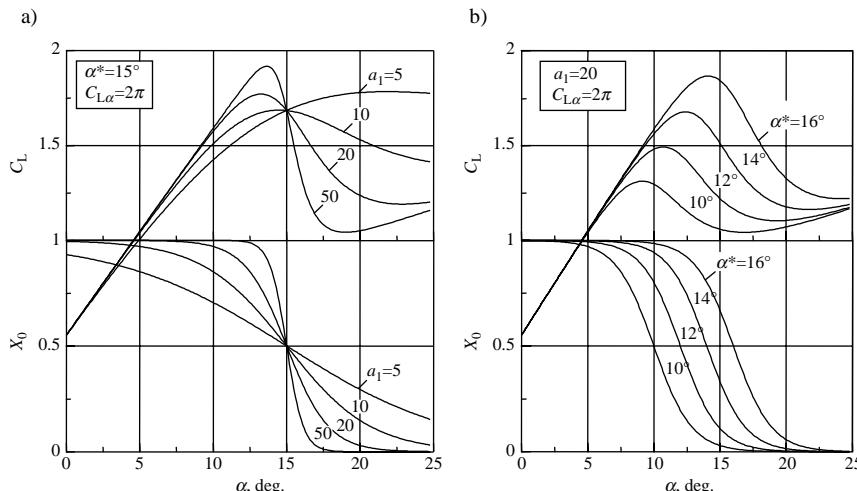


Fig. 12.18 Behavior of approximation function. a) Variation of a_1 , b) variation of α^* .

flow separation starts at about 10 deg. angle of attack, and it is fully developed at about 20 deg. The stall characteristic represented by the hyperbolic-tangent function in Eq. (12.18) is quite symmetrical with $X_0 = 0.5$ to the turning point α^* .

Now, combining Eqs. (12.17) and (12.18), we arrive at the general representation of unsteady flow, characterizing the transient and quasi-steady effects:

$$\tau_1 \frac{dX}{dt} + X = \frac{1}{2} \{1 - \tanh[a_1(\alpha - \tau_2 \dot{\alpha} - \alpha^*)]\} \quad (12.19)$$

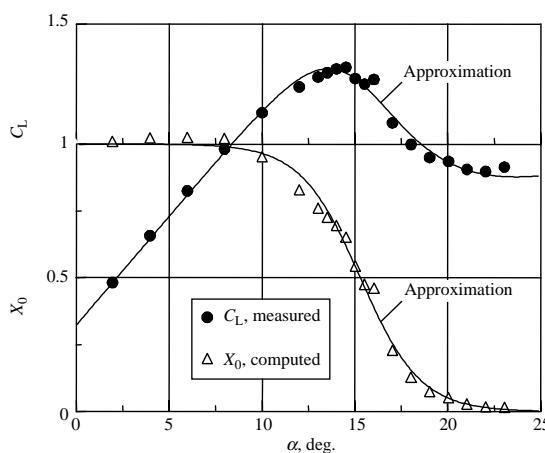


Fig. 12.19 Approximation of steady flow separation point.

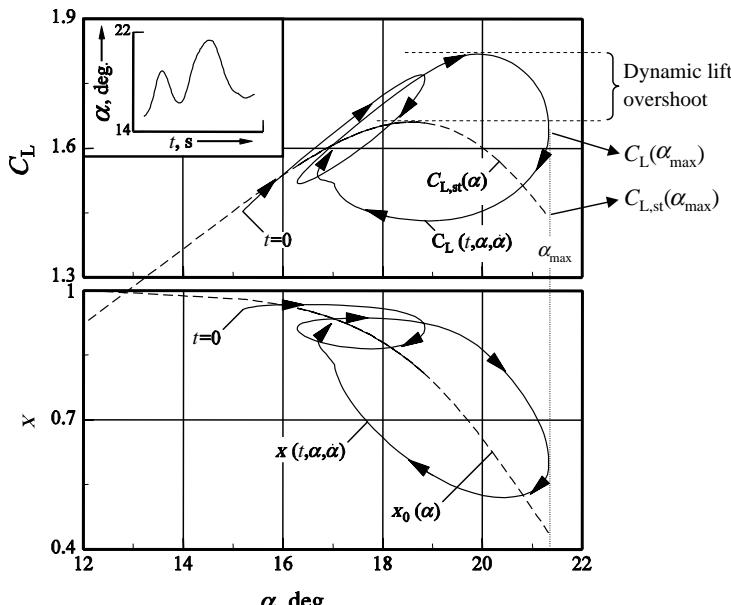


Fig. 12.20 Estimated lift coefficient and flow separation point during C-160 dynamic stall.

Successful estimation of the four parameters a_1 , α^* , τ_1 , and τ_2 appearing in Eq. (12.19) from dynamic stall maneuvers with a Transall C-160 has been reported in Ref. 29. A sample result in Fig. 12.20, taken from the same reference, shows a cross plot of C_L and X vs wing angle of attack during the dynamic stall portion. The plot shows the identified, steady $C_{L,st}(\alpha)$ and $X_0(\alpha)$, as well as the unsteady model outputs $C_L(t, \alpha, \dot{\alpha})$ and $X(t, \alpha, \dot{\alpha})$. We reach the following two conclusions from this figure: 1) there is a dynamic lift overshoot of about 10% above $C_{L,\max}$; 2) the angle of attack reaches into the post-stall region. The delaying effect of τ_1 can be inferred from the fact that at the maximum angle of attack of about 21.4 deg., the rate of change of angle of attack is zero, but $C_L(\alpha_{\max})$ is not the same as $C_{L,st}(\alpha_{\max})$ under static conditions.

VI. Quasi-steady Stall Modeling

Although Eq. (12.19) provides a model characterizing both the transient and quasi-steady stall characteristics in terms of four parameters, namely a_1 , α^* , τ_1 , and τ_2 , as common to any exercise, estimation of these parameters requires appropriate flight maneuvers containing adequate information necessary to estimate each parameter separately. Determination of both the time constants τ_1 and τ_2 requires highly dynamic stall maneuvers.

In many cases, however, a simplified approach accounting for quasi-steady stall characteristics is adequate and recommended. For example, in the case of

real-time flight simulators, the primary goal is usually to train pilots for correct recovery procedures from quasi-steady stall conditions. Elimination of an additional state equation also leads to minor reduction in overall computational burden.³¹ Furthermore, flight data with dynamic stall is more difficult and risky to gather, whereas quasi-steady stalls can be performed more readily. Flight data with quasi-steady stall would enable estimation of the hysteresis time constant τ_2 only. Accordingly, in the following investigation we neglect the transient effects by setting τ_1 to zero, which eliminates the need for differential equation and simplifies Eq. (12.19) to:

$$X = \frac{1}{2} \{ 1 - \tanh[a_1(\alpha - \tau_2 \dot{\alpha} - \alpha^*)] \} \quad (12.20)$$

The three parameters a_1 (airfoil static stall characteristics), τ_2 (time constant), and α^* (break point) are completely adequate to model the stall hysteresis.

Thus, the total lift given by Eq. (12.16), which is repeated here, and considering simplified expressions for the purpose of demonstration, the total drag and pitching moment coefficients are modeled as

$$C_L(\alpha, X) = C_{L0} + C_{L\alpha} \left\{ \frac{1 + \sqrt{X}}{2} \right\}^2 \quad (12.21)$$

$$C_D = C_{D0} + \frac{1}{e\pi\Lambda} C_L^2(\alpha, X) + \frac{\partial C_D}{\partial X}(1 - X) \quad (12.22)$$

$$C_m = C_{m0} + C_{m\alpha} \alpha + C_{mq} \frac{q\bar{c}}{V} + C_{m\delta_e} \delta_e + \frac{\partial C_m}{\partial X}(1 - X) \quad (12.23)$$

where Λ is the wing aspect ratio, e the Oswald factor and δ_e the elevator deflection. The major contribution to the unsteady drag comes from the drag polar using $C_L(\alpha, X)$, that is, from the lift modified due to flow separation. Any additional effects are accounted for through an empirical correction term $\partial C_D / \partial X$. The parameter $\partial C_m / \partial X$ models the hysteresis effect in the pitching moment. It is once again pointed out that Eqs. (12.21)–(12.23) are simplified expressions, which neglect additional terms accounting for contributions due to other variables and control deflections. They have been omitted in these expressions for the sake of brevity, but included in the model used for actual parameter estimation from flight data.

The parameters of the unsteady aerodynamic model for flow separation and stall were identified from flight data for three different aircraft, 1) ATTAS, 2) Dornier 328, and 3) Transall C-160. We present here the application of the software developed in Chapter 4 to the first case of ATTAS aircraft. This test case can be run and the results verified using the models and recorded data during a quasi-stall maneuver supplemented to the book. For the second example we provide typical results. In the third case of Transall C-160, the basic quasi-steady longitudinal stall model has been extended to asymmetric stall behavior observed in flight.

A. ATTAS Stall Modeling

In this section we return to the example considered in Chapter 8, Sec. VII.C, namely modeling of quasi-steady stall. Here, we apply the advanced modeling concepts based on the physics of the unsteady aerodynamics as elaborated in the foregoing section. The case considered pertains to flight data gathered with the test aircraft VFW-614 ATTAS (see Fig. 12.3) at an altitude of 16,000 ft and for clean configuration undergoing quasi-steady stall. Both output error method and regression analysis are applied to estimate parameters characterizing the stall characteristics. The results obtained applying the output error method are found in Ref. 29, in which the six DOF equations of motion combined with the aerodynamic model including stall model as postulated in Eqs. (12.20)–(12.23) has been used to match the recorded motion variables such as acceleration, angular rates, and attitude angles. We present here the application of the regression analysis, that is, least squares method, to this case and restrict ourselves to longitudinal motion only.

Since the postulated model, Eqs. (12.20)–(12.23), is nonlinear in parameters, it is imperative that we can obtain the LS estimates only iteratively, that is, by applying suitably the OEM software/FVSysID/chapter04/ml_oem. We designate this case as *test_case* = 27. As in the case of Chapter 6, Sec. IX.B and Chapter 8, Sec. VII.C, the first step consists of data pre-processing to compute the aerodynamic force and moment coefficients. These pre-computations are performed in the function “umr_reg_attas.m.” Since it is the same function that was used in the other two cases, we do not repeat the details here, but they can be easily traced from the said function. The recorded flight data from two stall maneuvers (data files \FVSysID\flt_data\fAttas_qst01.asc and fAttas_qst02.asc) are analyzed. The time segment related information *Nzi* and *izhf* is defined by following a procedure similar to that described in Chapter 8, Sec. VII.C.

Having derived the aerodynamic force and moment coefficients from the relevant measured data, we now define the input–output variables for parameter estimation based on the postulated model of Eqs. (12.20)–(12.23) as follows:

	No. of variables	Function name
States	0	—
Outputs	3	<i>C_D, C_L, C_m</i>
Inputs	14	<i>δ_e, δ_a, δ_r, p, q, r, α, β, V, ρ, F_{eL}, F_{eR}, ḡ, Ma</i>

Not all signals from the input array *Uimp* are required in the present case, but it is made oversized for the sake of similarity with other cases.

The 14 unknown derivatives *C_{D0}*, *e*, *C_{L0}*, *C_{Lα}*, *C_{LαMa}*, *C_{m0}*, *C_{ma}*, *C_{mq}*, *C_{mδe}*, *a₁*, *α^{*}*, *τ₂*, $\partial C_D / \partial X$, and $\partial C_m / \partial X$ appearing in Eqs. (12.20)–(12.23) are to be estimated. The contribution of the elevator deflection to the lift is indirectly accounted for through the term $(-C_{mδe}^* \bar{C}/r_H)$. Since we apply the LS method, there are no state equations. For the reasons elaborated in Chapter 6, Sec. IX.B, we either use the same dummy function “xdot_attas_reg” or define the function name for state equations as an empty string. The right-hand sides of the model, Eqs. (12.20)–(12.23), in terms of the above mentioned 14 parameters

are programmed in the function “obs_TC27_attas_regStall.” Thus, the model definition provided by the function “mDefCase27.m” for the software program “/FVSysID/chapter04/ml_oem” is as follows:

```
test_case = 27; % test case number
state_eq = [ ]; % empty string for state equations
obser_eq = 'obs_TC27_attas_regStall'; % function for observation equations
Nx = 0; % number of states
Ny = 3; % number of observation variables
Nu = 14; % number of input variables
Nparam = 14; % total number of parameters
dt = 0.04; % sampling time
```

For the reasons mentioned in Chapter 6, Sec. IX.C dealing with the estimation of a nonlinear lift and drag model, we specify some reasonable starting values for the parameters, which are found in the model definition “mDefCase27.m” for this example. The estimation run is started by calling “/FVSysID/chapter04/ml_oem.” The final estimation results are provided in Figs. 12.21 and 12.22. From the crossplot of lift coefficient vs angle of attack in Fig. 12.21, it is observed that the estimated hysteresis loop (shown in dashed lines) matches the flight data (shown in continuous lines) fairly well. The data contains a considerable amount of noise resulting from buffet; but the estimates are still reliable in the presence of noise. In Fig. 12.22, the three plots from the top show the match between the flight measured and model estimated aerodynamic coefficients. The match for the pitching moment coefficient shows some discrepancies in Fig. 12.22, which are attributed to the simplified linear model used in the regression. A more

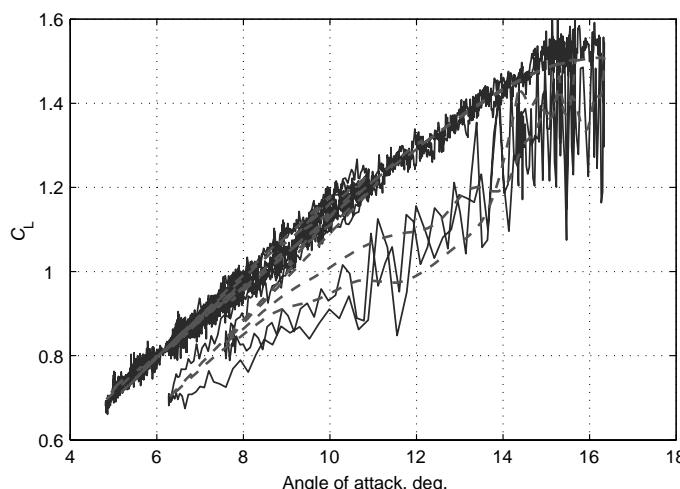


Fig. 12.21 ATTAS stall hysteresis (—, flight derived; -----, model estimated).

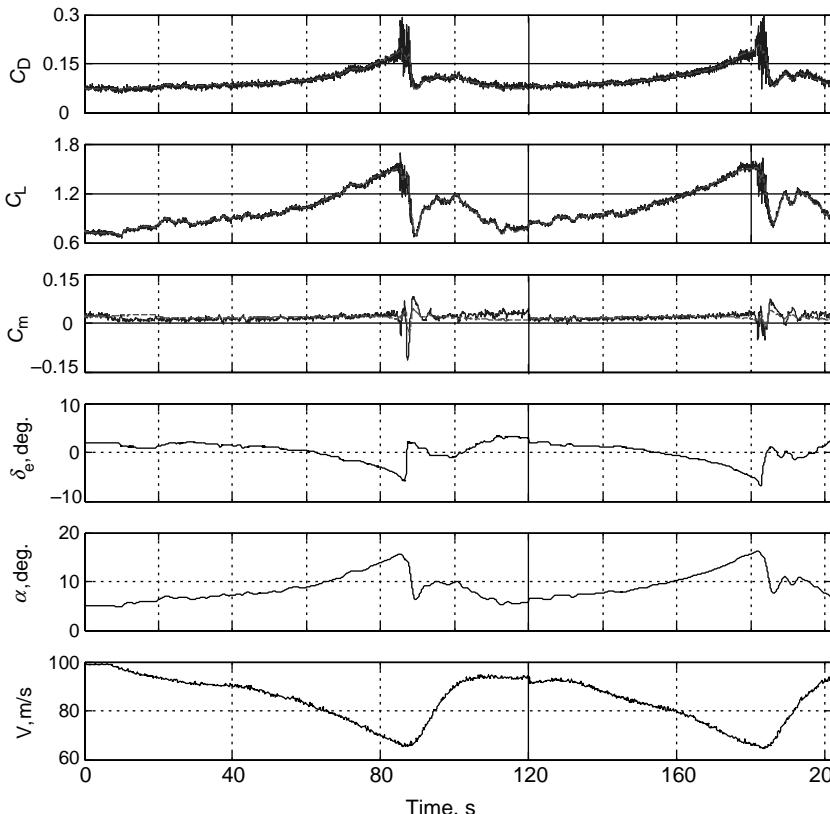


Fig. 12.22 Time histories of quasi-steady stall (—, flight measured; - - -, model estimated).

complex model incorporating angle of attack dependent elevator effectiveness provided a significantly improved match than that observed here. The three parameters modeling the stall characteristics are estimated to be $a_1 = 23.72$, $\alpha^* = 0.3087$ rad (≈ 17.7 deg.), and $\tau_2 = 24.02 \bar{c}/V$.

B. Dornier 328 Stall Modeling

The Dornier 328 is a high-wing, high-tail, twin turboprop regional transport aircraft designed to carry up to 33 passengers; see Fig. 12.23. It is powered by two Pratt & Whitney Canada PW 119A turboprop engines with six-bladed constant-speed propellers, each capable of generating up to 2180 SHP. The aircraft has an overall length of 69 ft 10 in, a wingspan of 68 ft 10 in, a mean aerodynamic chord of 6 ft 8 in, a wing area of 430.4 ft², and a horizontal tail area of 97.2 ft². The maximum takeoff and landing weight is 27558 lb, maximum rate of climb is 2420 feet/min and the maximum cruising speed is 335 KTAS. It has a maximum range of 700 nm with 30 passengers on board. In addition to



Fig. 12.23 Dornier 328 regional transport aircraft.

high-speed capability at all altitudes, the Dornier 328 has a low final approach and landing speed, providing short takeoff and landing capability. As will be discussed in Sec. VIII, a comprehensive database for a training simulator was developed from flight test data applying system identification. Modeling of quasi-steady stall characteristics from flight data was a part of this comprehensive database development program.³³

A model similar to that in Eqs. (12.20)–(12.23) has been applied to identify the stall characteristics of the Dornier 328 aircraft flying a stall maneuver with quasi-steady approach.^{4,33} All parameters of the basic aerodynamic model estimated in a foregoing step were kept fixed, estimating only the three parameters of the stall model. These three parameters are determined to be $\alpha_1 = 15.0$, $\alpha^* = 21.8$ deg., and $\tau_2 = 18.5\bar{c}/V$. Figure 12.24 shows a comparison of the total lift and pitching moment coefficients derived from flight measured data (shown by solid lines) with those estimated using system identification (shown by dashed lines).⁴ The increment in pitching moment due to unsteady

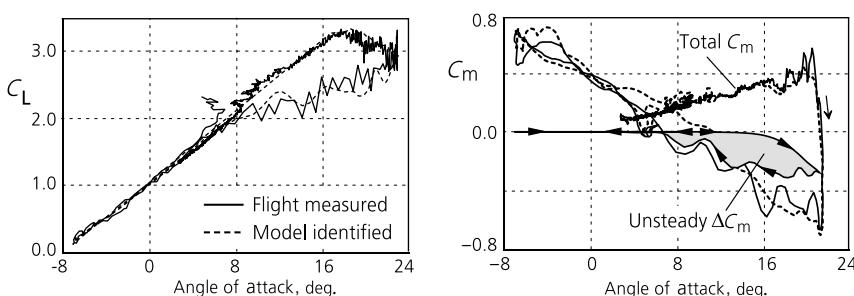


Fig. 12.24 Modeling of Dornier 328 stall hysteresis.

aerodynamics, leading to hysteresis, is plotted separately in Fig. 12.24 (shaded area). The time histories of important parameters are compared in Fig. 12.25.

The efficacy of the analytical model postulate is evident from the hysteresis in the lift and pitching moment coefficients. It is obvious from these figures that there are strong hysteresis effects during the pitch down part of the stall. The estimated model reproduces the hysteresis quite satisfactorily; only during the recovery phase are some small discrepancies observed.

In this section we have demonstrated the applicability of a fairly simple analytical model for quasi-steady stall characteristics. In general, the three parameters in Eq. (12.20) were estimated accurately and reliably, as suggested by the associated very low standard deviations, uncorrelated estimates, and the good match between the flight measured and model estimated responses. This was the case for the two examples presented here and for the case that will be discussed in the next section as well. The investigations with three different aircraft indicate that the quasi-steady stall model is quite adequate for simulation of

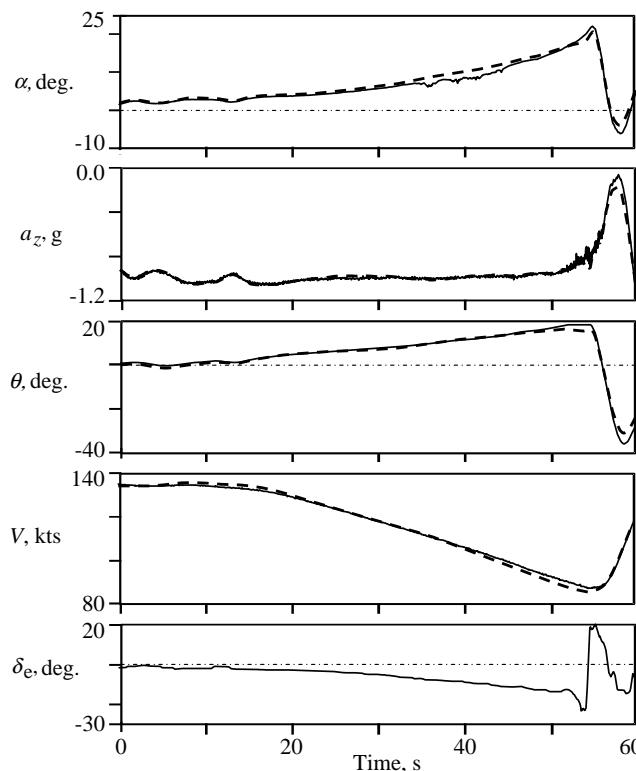


Fig. 12.25 Dornier 328 stall hysteresis time histories (—, flight measured; - - -, model estimated).

flow separated maneuvers with low dynamic effects. Such flight validated stall models are incorporated in the aerodynamic databases for the aforementioned level D training simulators Transall C-160 and Dornier 328, and the in-flight simulator ATTAS. The same approach was also applied to the X-31A post-stall maneuvering program covering up to 70 deg. angle of attack. Accurate stall modeling leads to an effective flight crew training on a simulator under extreme flight conditions with flow separation, and meets the recommendations made by the NTSB (National Transportation Safety Board) after a Jetstream 4101 aircraft crash following a stall due to improper pilot reaction to stall warning.³⁴

C. Asymmetric Stall Modeling

Analysis in the stall and post-stall region is more or less confined only to the longitudinal dynamics. As demonstrated in the preceding section, a state-space model of unsteady aerodynamics has been applied to longitudinal flight data. Flight analysts have, however, encountered cases in the past where an aircraft in stall is known to have exhibited large excursions in lateral-directional motion. This asymmetric behavior is attributed mainly to the significant sideforce and yawing moments generated because of formation and shedding of the forebody vortices.^{35,36} An asymmetric behavior was observed during the quasi-steady stall with the military transport aircraft Transall C-160, exhibiting more than 20 deg. bank variations. One possible cause for this phenomenon observed in flight could possibly be that the two engines equipped with four-bladed propellers rotating in the same direction influence the flow separation asymmetrically.

Applying the quasi-steady stall model discussed in Sec. VI to Transall C-160 data, it was found that the longitudinal behavior in stall could be satisfactorily reproduced. However, the same cannot be stated for the lateral case where poor simulation quality, specifically in quasi-steady stall approach, for the lateral-directional time histories was observed. It was observed that the quasi-steady stall is accompanied by an uncommanded left roll-off. This is attributed to unequal lift caused by different flow separation behavior on the left and the right wing. To account for the asymmetric flow separation, as depicted schematically in Fig. 12.26, the two wing panels are treated separately.³⁷ The total lift and drag appearing in Eqs. (12.21) and (12.22) are given by the sum of the components due to the left and right wings. Now, a simplified representation of the rolling and yawing motion resulting due to lift and drag differentials can be formulated as

$$\begin{aligned}\Delta C_\ell &= (C_{L_L} - C_{L_R}) \frac{\Delta y}{2b} \\ \Delta C_n &= (C_{D_L} - C_{D_R}) \frac{\Delta y}{2b}\end{aligned}\tag{12.24}$$

where the suffixes L and R represent the entities for the left and right wing, respectively, Δy the effective lever arm and ΔC_ℓ and ΔC_n the incremental changes to the rolling and yawing moments resulting from asymmetric flow

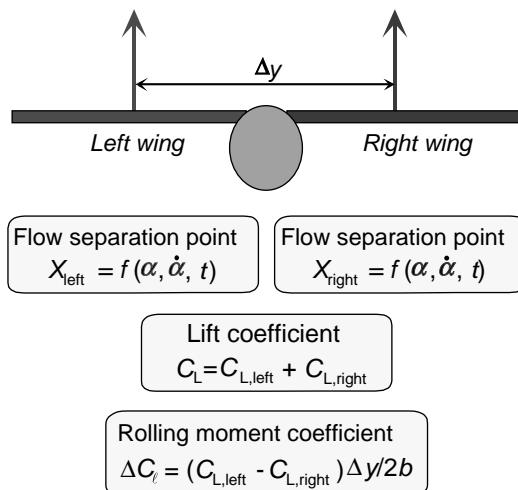


Fig. 12.26 Schematic of asymmetric stall modeling.

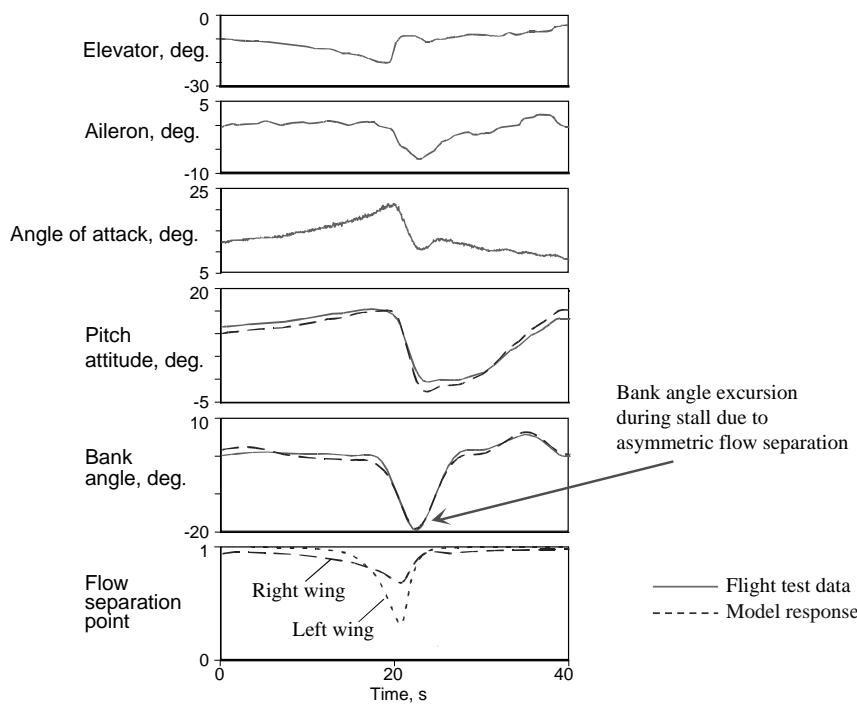


Fig. 12.27 Time histories of asymmetric stall behavior.

separation. The effective lever arm which can be assumed to be half the wing span or estimated from flight test data.³⁷

Equation (12.24), incorporating the lift and drag differentials, leads not only to a parsimonious model, but is also based on the physical interpretation of the force–moment relationship. This approach appears to be preferable over the other possible approach of modeling the changes in rolling and yawing moments due to asymmetric unsteady flow through additional aerodynamic derivatives in the respective moment coefficients.

The stall model of Eqs. (12.20)–(12.23) extended by treating the left and right wings separately and through Eq. (12.24) is applied to analyze the quasi-steady stall maneuver. The stall parameters a_1 , α^* , and time constant τ_2 for the left and right wing are estimated separately. The efficacy of such a fairly simple formulation to satisfactorily capture the lateral-directional motion during stall is evident from the comparison of the estimated and flight measured responses shown in Fig. 12.27. The asymmetry in flow separation is clearly observed from the plots of the separation points X_L and X_R for the left and right wing.

VII. Ground Effect Modeling

The ground effect is a physical phenomenon that results from the proximity of the moving flight vehicle to the ground, mainly because close to the ground the downward flow is interrupted by the ground. Loosely speaking, for example, during the landing phase, the ground effect leads to increasing pressure below the wings as the aircraft approaches the ground and thereby reduces the sink rate. Such effects are encountered, in general, for altitudes above ground level lower than the wing span. They have a dominant influence on the aircraft landing and takeoff performance, which are safety-critical flight phases. In general, the three main effects resulting from ground effect are⁸ 1) reduced downwash angle at the tail, 2) increase in the wing-body and the tail lift curve slopes, and 3) reduction in the induced drag. The ground effect mainly influences longitudinal motion, although, in general, almost all other aerodynamic derivatives might be slightly altered. Ground effect updates from flight data are usually necessary to accurately reproduce the landings and takeoffs, as well as for better assessment of touchdown sink rate.

The basic theory on ground effect was developed by Wieselsberger, analyzing it through a simple model based on Prandtl's lifting-line theory.^{38,39} To date extensive research has been devoted to this important topic. There is a vast literature, including dedicated text books, on ground effect modeling based on theory, numerical investigations, wind-tunnel tests, and determination from flight tests.^{40–44} A detailed discussion on the theoretical background and on the experimental investigations applying conventional methods is far beyond the scope of this book. Consistent with the overall philosophy followed in this book, we take a cursory look at the physics of the phenomenon and then address the present problem by considering analytical models amenable to parameter estimation applying the system identification techniques developed in this book. In the following we draw heavily on the investigations reported in Refs. 45 and 46.

Loosely speaking, the ground effect leads to reduction of the wing-tip vortices, which is equivalent to a reduction of the induced angle of attack. This implies that, for a constant geometrical angle of attack, the lift coefficient will increase as the aircraft approaches the ground. The reduction in the induced angle of attack results consequently in a reduction of the induced drag. Under ideal conditions (wing with elliptical lift distribution) Wieselsberger formulates the induced drag in terms of a ground effect influence function σ as³⁸

$$C_{Di} = (1 - \sigma) \frac{C_L^2}{e\pi\Lambda} \quad (12.25)$$

The various terms in Eq. (12.25) have already been defined in Chapter 6, Sec. IX.C. Equation (12.25) is the same as Eq. (6.78), but for the modification factor $(1 - \sigma)$ due to the ground effect. The ground effect influence function σ is a nonlinear function of non-dimensional height h/b , and varies between 0 and 1. Based on the lifting-line theory,³⁹ Prandtl provided values for σ , which are plotted in Fig. 12.28. Assuming a quadratic polar, it turns out that the L/D (lift to drag) ratio under the ground effect to that for the free stream conditions is given by $1/\sqrt{1 - \sigma}$.

The contributions to pitching moment due to wing as well as horizontal tail are also affected by the ground effect. The change in downwash at the horizontal tail is more pronounced than the pitch-up generated by the ground effect on the wing alone. Owing to the reduced downwash, the tail angle of attack increases, resulting in increased tail lift contribution and pitch-down moment of the overall aircraft.

In general, the incremental aerodynamic coefficients ΔC_L , ΔC_m , and Δe (lift, pitching moment, and downwash increments, respectively) induced by

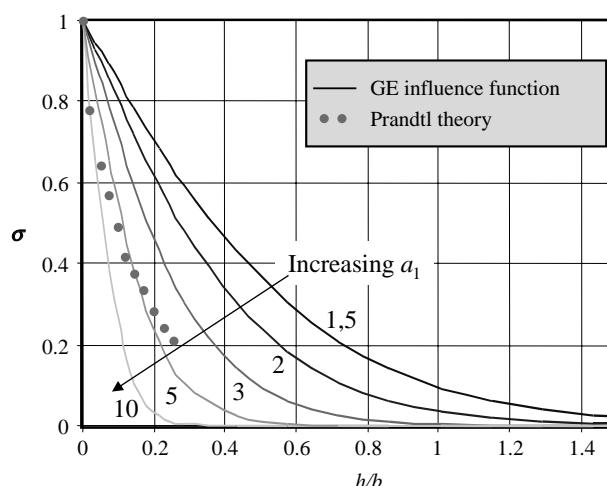


Fig. 12.28 Ground effect influence function.

ground effect are functions of 1) flap configuration, 2) Mach number, 3) angle of attack, 4) height above ground, 5) flight path angle, and 6) thrust level through jet interaction. For a given flap configuration, the ground effect varies nonlinearly as the aircraft approaches the ground. To enable parameter estimation, as in the case of stall hysteresis discussed in the previous section, we once again use the hyperbolic-tangent function to analytically model the ground effect as follows:

$$\sigma = 1 - \tanh\left(a_1 \frac{h}{b}\right) \quad (12.26)$$

where h is the height above ground level and b the wing span. The functional behavior of Eq. (12.26) for different values of a_1 is shown in Fig. 12.28. It is apparent that the functional value is limited from 0 to 1, and the parameter a_1 , which is a function of the configuration (like flap setting), characterizes the bending of the curves. The analytical model approximates the theoretical values provided by Prandtl, shown as solid circles; for such a case, in the limit the function has infinite slope for $h/b = 0$, and thus conforms to the theory.

Equation (12.26) is a general representation of the ground effect influence function, whereby the altitude of the wing above the ground is the critical parameter that determines the intensity of the ground effect. It approximates adequately the nonlinear behavior of aerodynamics in the proximity to ground for $h < b$.^{42,43} and hence is particularly suitable for low-wing configurations, for example the VFW-614 ATTAS aircraft (see Fig. 12.3). For many high-wing configurations, for example Transall C-160 (see Fig. 12.9), such low altitudes are more or less not encountered by the wing. In such cases, as a first-order approximation, a linear ground effect influence function is used instead of Eq. (12.26). Here we investigate both these cases from flight test maneuvers.

Irrespective of the type of model (linear or nonlinear) being identified to account for the ground effect, there are three important aspects critical to determination of ground effect from flight data, 1) flight test technique, 2) accurate height above ground level signal, and 3) high fidelity in-air aerodynamic model including landing gear effects.

The flight test techniques that allow estimation of ground effect consist of 1) landings, 2) takeoffs, 3) touch and go, 4) tower flyby over runway at different height AGL (typically 100, 40, 20, and 10 ft), and 5) tower flyby at different height AGL with small dynamic inputs in pitch, roll, and yaw. Extreme caution is called for during such flight testing. Furthermore, such maneuvers should be performed under very calm atmospheric conditions, with no wind and no turbulence. Such ideal conditions exist rarely, hence some minimum amount of horizontal wind must be tolerated, but the turbulence should be negligible. If the horizontal wind is non-negligible, a more advanced approach incorporating GPS/DGPS measurements will be necessary. A wind gradient, if encountered during such maneuvers, is more critical than constant horizontal wind.

Besides gathering flight data with sufficient information content, reliable extraction of a ground effect model needs accurate height AGL information, because the ground effect is a strong function of the nondimensional height h/b . Some aspects of height AGL calibration using radar altimeter and independent measurements using cameras have already been discussed in Chapter 10, Sec. VII. Furthermore, the ground effect is usually extracted from flight maneuvers as incremental deltas to the in-air aerodynamic model. Thus, invariably an accurate model for in-air aerodynamic, particularly in the low landing speed regime, including the landing gear effects is a prerequisite. Aspects of high-fidelity in-air aerodynamic modeling will be covered in the next section, whereas those of determining the landing gear effects have been presented in Sec. III.

The first case pertains to the Transall C-160 aircraft with a high-wing configuration (see Fig. 12.9), for which the wing is about 4 m above the ground and the mean aerodynamic chord \bar{c} is 4.18 m. Thus, for the reasons already elaborated (altitudes for which the ground effect changes nonlinearly ($h < \bar{c}$) are not reached by the high-wing configuration), we consider here a linear model for the ground effect. Accordingly, the model postulates for the incremental lift, drag, and pitching moment coefficients are formulated as

$$\left. \begin{aligned} \Delta C_{L, GE} &= \frac{\partial C_L}{\partial h}(h - h_{GE}) \\ \Delta C_{D, GE} &= \frac{\partial C_D}{\partial h}(h - h_{GE}) \\ \Delta C_{m, GE} &= \frac{\partial C_m}{\partial h}(h - h_{GE}) \end{aligned} \right\}, \quad h < h_{GE} \quad (12.27)$$

where for modeling purposes h is the altitude of the CG above ground, and h_{GE} the maximum ground-effect altitude (i.e., for altitudes h greater than h_{GE} the ground effect vanishes).

The three unknown parameters $\partial C_L/\partial h$, $\partial C_D/\partial h$, and $\partial C_m/\partial h$ are to be estimated from flight data. Figure 12.29 shows the identified coefficients plotted as a function of h/b . The maximum altitude h_{GE} was determined to be 20 m, which corresponds to $h/b = 0.5$. Estimation of ground-effect parameters appearing in Eq. (12.27) was mostly based on the landings and takeoffs, mainly because turbulence was encountered during the low-altitude flyby flights. In general, variations of ± 0.5 kts indicate very low turbulence, $\pm 1-1.5$ kts light turbulence, and $\pm 2.0-2.5$ kts light-to-moderate turbulence. The ground effect was identified separately for each flap position. The above simplified ground-effect model provided good landing performance, whereas the model fidelity for takeoff was just adequate. To further improve model characteristics during takeoffs, particularly after rotation, increased elevator effectiveness as a function of tail altitude above ground was identified. This is a possible empirical correction, because, during this phase, based on the ratio of the altitude above ground to the tail span, the tail comes into regions where the ground effects are not negligible. The linear model identified in Eq. (12.27) primarily accounts for the contributions due to the wing.

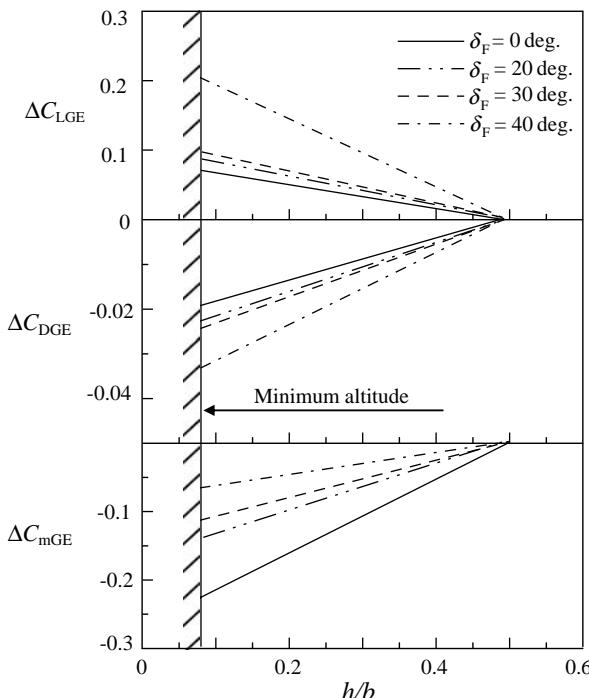


Fig. 12.29 Estimated ground effect (GE) for top-wing C-160 aircraft.

The second case pertains to the VFW-614 ATTAS aircraft with low-wing configuration; see Fig. 12.3. As already described in Secs. II.B and II.C the ATTAS aerodynamic model is a two-point model considering wing–body and horizontal tail separately. The overall model incorporates various dependencies and nonlinearities; the basis (i.e., those out of ground effect) contributions are denoted appending the respective subscripts through “basic.” Accordingly, the lift coefficient for the wing–body combination, including the ground effect, is given by

$$\begin{aligned} C_{L,WB} &= C_{L,WB_basic} + C_{L\alpha GE} \alpha \sigma_1 \\ C_{L,H} &= C_{L,H_basic} + C_{L\alpha HGE} \alpha_H \sigma_2 \\ C_{m25} &= C_{m25,basic} + C_{m0,GE} \sigma_1 \end{aligned} \quad (12.28)$$

where the subscript GE indicates contributions due to ground effect. The downwash at the horizontal tail is given by

$$\varepsilon_H = \varepsilon_{H,basic} + (\partial \varepsilon / \partial \alpha)_{GE} \alpha (t - \tau) \sigma_2 \quad (12.29)$$

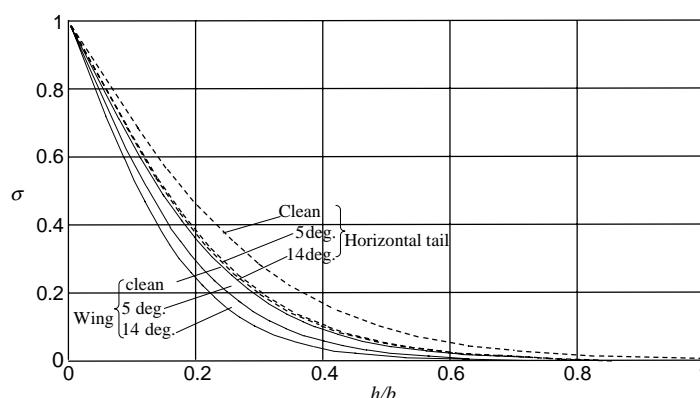
The major portion of the influence of the ground effect at the horizontal tail on the pitching moment is automatically accounted for in the two-point aerodynamic

Table 12.2 Estimates of GE parameters (source: Ref. 46)

Parameter	Flap		
	Clean	5 deg.	14 deg.
a_{1W}	6.9	4.4	5.0
a_{1H}	3.7	3.8	3.7
C_{LoGE}	1.3	2.3	1.1
$(\partial e/\partial \alpha)_{GE}$	-0.43	-0.69	-1.03
C_{m0GE}	0.11	0.085	0.111

model through the term tail-lift multiplied by the lever arm; see Eq. (12.5). The additional correction in the pitching moment is introduced to account for the ground effect on the wing–body through a zero pitching moment term.

We note that two influence functions σ_1 and σ_2 have been introduced to model the ground effect at the wing–body and horizontal tail separately. These functions are evaluated from Eq. (12.26) using two parameters a_{1W} and a_{1H} , respectively. Thus, a total of five parameters [a_{1W} , a_{1H} , C_{LoGE} , $(\partial e/\partial \alpha)_{GE}$, C_{m0GE}] characterize the ground effect model for the VFW-614 ATTAS. They are estimated from the following flight tests: 1) flybys at 100, 50, 20, and 10 feet AGL, each at 1.3 and 1.4 V_s , 2) similar flybys with small amplitude elevator, aileron, and rudder doublets, and 3) touch and go. All maneuvers were performed with extended landing gear. The horizontal wind was moderate, and turbulence level light to moderate. The estimates of the above five parameters for the three flap settings are provided in Table 12.2, and the ground-effect influence functions for the wing–body and horizontal tail are plotted in Fig. 12.30. As stated at the beginning of this section, the reduction in downwash at the tail has a more significant effect than that due to changes in lift and drag. From Table 12.2 it is observed that the downwash parameter due to ground effect, $(\partial e/\partial \alpha)_{GE}$, increases with flap position, leading to

**Fig. 12.30 Estimated GE influence function for low-wing VFW-614 ATTAS.**

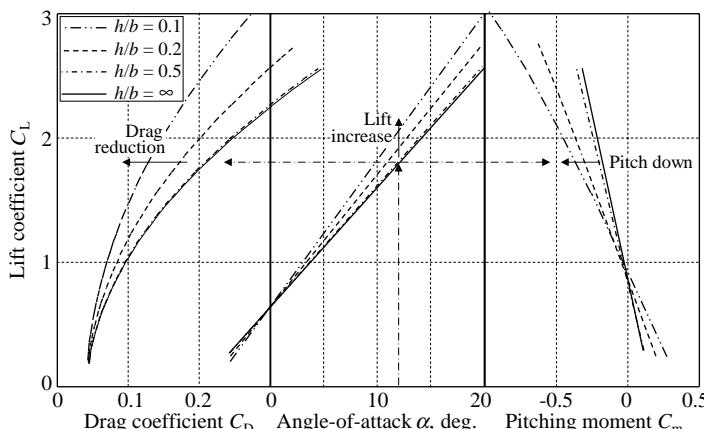


Fig. 12.31 VFW-614 ATTAS flight-identified total aerodynamic coefficients for 14 deg. flap (source Ref. 45).

stronger pitch-down motion. The influence function characterizing parameter for the tail, a_{1H} , is almost independent of the flap setting.

To better appreciate the influence of ground effect, the total longitudinal aerodynamic coefficients computed for different height AGL are plotted in Fig. 12.31, which confirms the three observations made at the beginning of this section. This can be inferred starting from the middle plot showing an increase in C_L as h/b decreases. Now, as shown schematically in the figure, for a constant angle of attack (or lift), moving along the dotted lines with arrow to the left and right, respectively, we notice reduction in the drag and the pitch down moment. The change in the total C_m results primarily from the reduction in the downwash.

To summarize, in this section we demonstrated estimation of ground effect parameters for high-wing and low-wing configurations from flight data using linear and nonlinear model postulates. Besides the two cases presented here, the same approach was also successfully applied in the case of X-31 VECTOR to identify ground effect from ESTOL (extremely short takeoff and landing) touch and go, and landing maneuvers.⁴⁷ An analytical model in terms of influence functions and three parameters to quantify the variation in downwash, lift, and drag proved very effective. The model not only conforms to the theoretical predictions, but also leads to a smooth transition from the in-air model to one accounting for ground effect. It leads to a homogeneous overall representation.

VIII. High-fidelity Databases for Training Simulators

Prior to the advent of digital computers, the accuracy of aircraft simulation was marginal due to the inherent limitations of analog computers. The digital computers opened up new horizons not only in the field of aircraft parameter estimation but also in the field of flight simulators.⁴⁸ With the evolution of

high performance modern aircraft and with the spiraling developmental and experimental costs, the importance of ground-based flight simulators and that of in-flight simulators increased continuously over the years. Simulators are increasingly used not only for pilot training but also for other applications such as flight planning, envelope expansion, design and analysis of control laws, handling qualities investigations, and pilot-in-the-loop studies. Many of these applications demand a high-fidelity flight simulator. The fidelity of a simulation depends to a large extent on the accuracy of the mathematical model and of the aerodynamic database representing the flight vehicle.

To arrive at flight-validated high-fidelity global databases for flight simulators, there are two viable approaches, 1) to verify, and if necessary, update the wind-tunnel predictions through incremental coefficients from flight data analysis, and 2) to generate a new database from flight data through a systematic system identification procedure starting from a basic six degrees-of-freedom model and extending it to include special effects like landing gear, high angle of attack regime, unsteady aerodynamics, ground effects, and engine-out effects. In both approaches the task is formidable and the process iterative, requiring model structure determination, and can be equally complex and tedious.

The first approach, often called incremental model update, makes maximum use of the a priori information available on an aerodynamic database derived from wind-tunnel measurements, theoretical predictions, or even intermediate updates based on preliminary flight data. It starts with computation of aerodynamic force and moment coefficients using flight-measured linear accelerations and angular rates, and comparing them with those computed from the so-called pre-flight database fed with measured aircraft motion variables and control surface deflections. The discrepancies between the two sets, called Δ , have to be eliminated, or minimized, to arrive at a flight validated and updated database. Figure 12.32 depicts schematically the approach just described. In this figure we

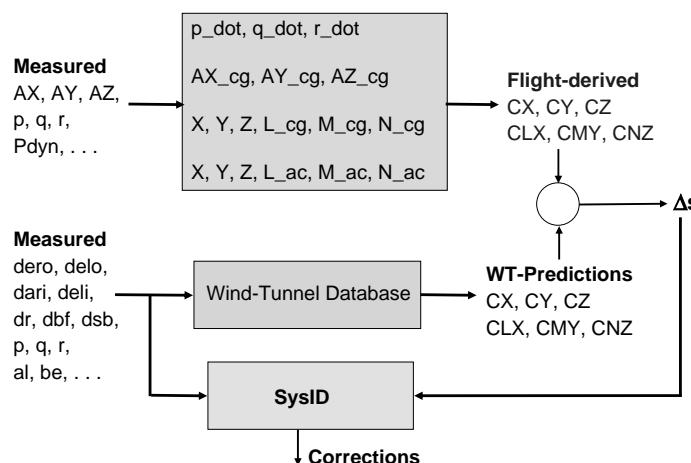


Fig. 12.32 Schematic of incremental model update procedure.

have given the notational names of the variables as they were used in a particular project, which we will address in Sec. XI. It leads to a two-step procedure, first, finding the discrepancies, and then identifying suitable models for these discrepancies. The first step of finding the deltas is fairly straightforward. The procedure to compute the aerodynamic forces and moments from the measurements of linear accelerations and angular rates has already been dealt with in Chapter 6, Sec. IX.A.

The second step of the approach is to fit appropriate models to the increments. In most of the cases, however, it turns out that this step is not a trivial one. Such updates are possible only through a repetitive procedure, because they are invariably functions of the angle of attack, Mach number, angle of sideslip, and other variables. They are also configuration-dependent. For estimation purposes, mostly the regression analysis based on the least-squares technique is applied, which allows direct handling of the aerodynamic coefficients. Although some model structure determination procedures can be applied to arrive at an acceptable model, considerable engineering judgments are usually necessary. The approach has found successful application in practice.^{49–52} It will be applied in Sec. XI to verify and update the longitudinal database of a reusable launch vehicle from free flights. A highly complex procedure following this general philosophy has also been applied to identify increments to the X-31A aerodynamic database to arrive at a global model. The increments were estimated in a table-lookup form, along with the break points as well, which automatically allowed more precision in matching the nonlinear regions. Discussion of this advanced X-31A global model development is far beyond the scope of this book. An excellent account is found in Refs. 47 and 53, analyzing in a single run up to 10 h of flight test time, separated into up to 99 time slices with a sampling rate of 12.5 Hz, covering representative flight maneuvers in cruise, high lift and power approach configurations, including their transitions and the acceleration/deceleration segments during takeoff and landing.

The second approach of generating a comprehensive homogeneous database, valid over the entire operational flight envelope, is equally if not more elaborate. It is also an iterative process, which starts with point-identification at all the flight tested points. Point-identification results in models related to specific trim conditions. Based on this bulk of the estimation results in the normal flight regime, the aerodynamic model postulates can be extended to include angle of attack, Mach number, and angle of sideslip dependencies, coupling derivatives, and nonlinearities. Through multipoint identification several flight conditions are analyzed simultaneously to arrive at an extended model. After freezing the validated model for the normal flight regime, the submodels of other aerodynamic effects resulting from landing gear, ground effect, stall approach and quasi-steady stall, takeoff and landing, single engine, and so on, are estimated and appended to the basic model. Depending upon the aircraft, special configurations may have to be modeled, for example ramp door or load drops for military transport aircraft. Figure 12.33 depicts schematically the approach just described, which was also followed in several cases to generate aerodynamic databases meeting the level D fidelity criteria specified by the FAA for training simulators.^{17,33} Both types of parameter estimation methods, namely aerodynamic coefficient matching by regression method and aircraft motion responses matching by output error method, have found successful applications.

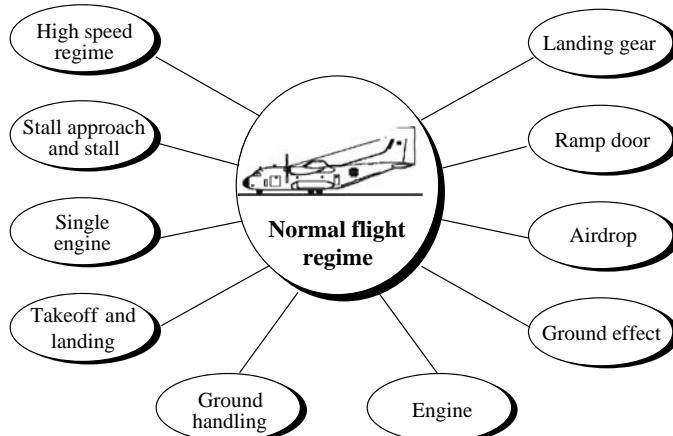


Fig. 12.33 Schematic of comprehensive model identification procedure.

The two approaches of database generation, verification and update, are more or less equivalent. The choice between the two is mainly dictated by the familiarity and background of the team of engineers involved in this task. The expected quality of the a priori model will also be a decisive factor, also the form (table look-up or derivative) because of the significant differences in the computational overhead. To a lesser extent, the quality of recorded data plays a role; as already discussed in Chapter 6, the two-step approach is primarily based on the least squares method, which requires good quality measurements and extensive data checking and a correction step to eliminate measurement errors. The second approach is less sensitive to these errors, as they can be determined during the estimation of the aerodynamic parameters. It also incorporates in the estimation procedure the integration of the six degrees-of-freedom equations of aircraft motion, which are very similar to those finally implemented on the training simulator. It is also possible to directly transfer the complete database including the equations of motion from estimation to simulator. It may also be better suited for modeling of more complex aerodynamic phenomena such as ground effect, stall, landing, and takeoff cases, which are found to be highly sensitive to influencing parameters, for example height above ground. Even very small errors in the measurement of such an influencing variable will affect the quality of the estimated model significantly; hence, in such cases the use of integrated variables may be preferable.

Irrespective of the approach we adopt, the importance of flight validation and updates is well recognized and accepted by database generators. This necessity arises mainly from the fact that, although the analytical predictions, wind-tunnel measurements on a scaled model, or extrapolation of existing data from similar configurations provide aerodynamic databases valid over the entire flight envelope, they are often associated with certain limitations arising from model scaling, Reynolds number, dynamic derivative and cross coupling effects. Moreover, in some instances, particularly for old generation aircraft, only limited information is available. In recent years, most aircraft manufacturers consider a simulator data package as an integral part of the aircraft development program.^{4,48}

Consistent with the goal of this chapter, which is to provide appreciation of various aerodynamic modeling and parameter estimation problems, we present some snapshots of results which were generated during the development of comprehensive databases for the military transport aircraft Transall C-160 (Fig. 12.9), and the commuter aircraft Dornier 328 (Fig. 12.23), in both cases meeting the FAA Level D quality criteria.^{17,33} This general approach was also followed in the generation of an aerodynamic database for the ATTAS in-flight simulator.^{14,15}

A. Transall C-160 Military Transport Aircraft

As a part of the data gathering for the Transall C-160 (see Fig. 12.9), 30 flights each of roughly 2.5 h were carried out with the instrumented aircraft. A total of 37 configurations covering five flap settings, three center of gravity locations, five different speeds at five altitude levels were tested. Roughly 1000 flight maneuvers were analyzed to derive a rigid-body aerodynamic database valid over the operational flight regime.^{16–18} In Secs. II–VII we have already discussed the modeling of specific aerodynamic effects. Here, in this section we present two examples highlighting nonlinear effects in the basic aerodynamic model, derived from the flights in the normal flight regime.

The lift, drag, and pitching moment coefficients are estimated from the equations of motion pertaining to the longitudinal motion, using the motion variables of the lateral-directional motion as pseudo-control inputs. As elaborated in Chapter 3, Sec. VI, introducing variables of the other modes of motion as pseudo controls allows working with the complete nonlinear equations of motion in each case. The longitudinal derivatives are estimated for a two-point model considering, as described in Sec. II, wing–body and tail separately. The underlying equations for the lift and pitching moment coefficients are given by

$$C_L = C_{L0} + C_{L\alpha W}\alpha(t) + \frac{S_H}{S} C_{LH} \quad (12.30)$$

where the lift due to the horizontal tail is given by

$$C_{LH} = C_{L\alpha H}\alpha_H + C_{L\delta e}\delta_e \quad (12.31)$$

The pitching moment coefficient is given by

$$C_m = C_{m0} - \frac{r_H^* S_H}{\bar{c}} C_{LH} + C_{mqW} \frac{q\bar{c}}{V} \quad (12.32)$$

The effective angle of attack at the horizontal tail, α_H , accounting explicitly for the downwash lag effect, is modeled as follows:

$$\alpha_H = \alpha + i_H - \frac{\partial \varepsilon_H}{\partial \alpha} \alpha(t - \tau_\alpha) + \frac{\partial \varepsilon_H}{\partial C_S} C_S(t - \tau_{CS}) + \alpha_{dyn} \quad (12.33)$$

where the various variables have already been defined in previous sections. Equations (12.30)–(12.32) are basically the same as Eqs. (12.4) and (12.5), which we used in Sec. II for the downwash lag effect investigations. Similarly, Eq. (12.33) is same as Eq. (12.12), but for the term corresponding to the speed brake.

The derivatives appearing in Eqs. (12.30)–(12.33) are estimated from multi-run evaluation at each trim point by combining a 3–2–1–1 elevator input maneuver exciting the short period motion, an elevator pulse input maneuver exciting the phugoid motion, and a thrust input maneuver. A typical plot of the point-identification for the estimated elevator control effectiveness as a function of trim angle of attack is shown in Fig. 12.34 for the landing flap of 0 deg., that is, clean configuration. Estimates obtained from eight different flights clearly show that the elevator effectiveness is a function of the angle of attack, and the dependency is nonlinear.

An extended model accounting for the dependency of elevator effectiveness on the angle of attack yields the match shown in Fig. 12.35, when applied to simulate two different maneuvers, one a doublet input with not so large elevator deflection and the other a stall approach in which large elevator doublet inputs were simultaneously applied while pulling the stick. The match between the flight measurements (continuous lines) and model prediction (dotted lines) is very good for the first maneuver, whereas the second maneuver shows some discernible discrepancies. From the time-expanded plot of a portion of the maneuver shown in the same figure, it is apparent that the model response overshoots, but only when the deflections are large. This is typical of nonlinear characteristics of control surfaces, as shown in an inset in the same figure, having reduced effectiveness for larger deflections, which is not yet accounted for.

Now, to account for the nonlinear control surface effectiveness, we extend further the model of Eq. (12.31) for the lift generated at the horizontal tail as

$$C_{LH} = C_{LaH}\alpha_H + [C_{L\delta e} + C_{L\delta\alpha}\alpha_H + C_{L\delta\alpha_2}\alpha_H^2 + C_{L\delta\beta}\delta_e^2]\delta_e \quad (12.34)$$

where the derivatives $C_{L\delta\alpha}$ and $C_{L\delta\alpha_2}$ account for the nonlinear dependency seen in Fig. 12.34, and the derivative $C_{L\delta\beta}$ the nonlinearity in control surface effectiveness as schematically depicted in Fig. 12.35. The extended model of Eq. (12.34) yields the response match shown in Fig. 12.36 for the same two flight maneuvers.

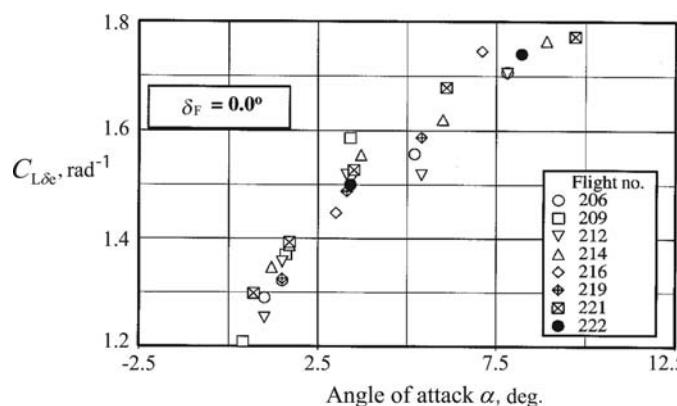


Fig. 12.34 Transall C-160 elevator control effectiveness.

SELECTED ADVANCED EXAMPLES

439

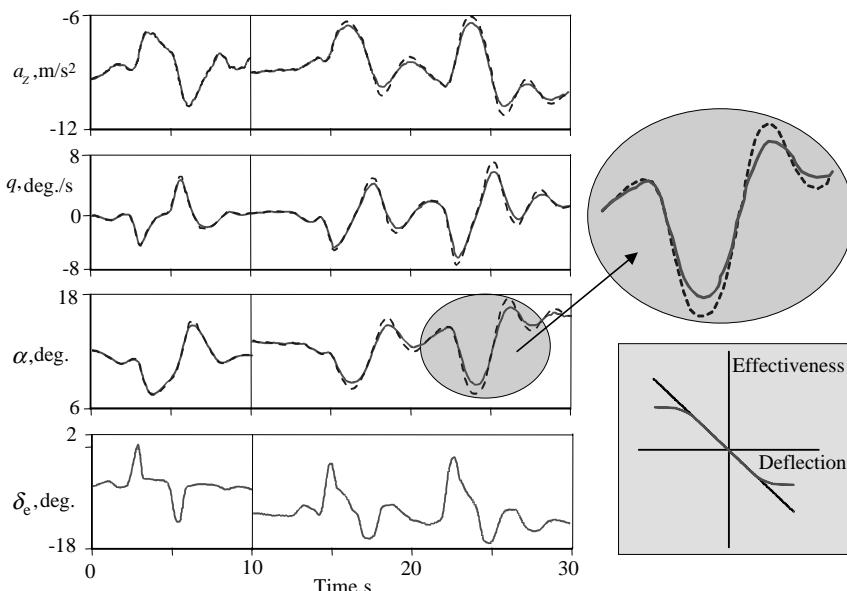


Fig. 12.35 Identification of linear model for elevator effectiveness (—, flight measured; -·-, model estimated).

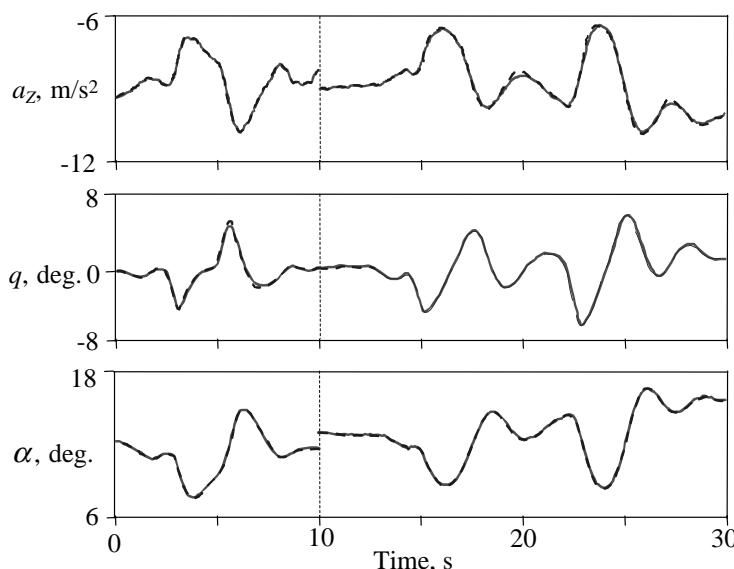


Fig. 12.36 Identification of nonlinear model for elevator effectiveness (—, flight measured; -·-, model estimated).

Compared with Fig. 12.35, the match between model prediction and flight measurement is now excellent.¹⁷

The second example pertains to the lateral-directional motion. Once again, following the same philosophy of point-identification at a large number of trim points, strong nonlinear effects and dependency of parameters pertaining to the lateral-directional motion on the angle of attack were identified. Without going into any further specific details, we state here the final identified model for the yawing moment coefficient:^{4,17}

$$\begin{aligned} C_n = & C_{n0} + [C_{n\beta} + C_{n\beta NL}(\alpha - \alpha_S)]\beta \\ & + (C_{np} + C_{np\alpha}\alpha)p^* \\ & + [C_{nr} + C_{nr NL}(\alpha - \alpha_S)]r^* \\ & + (C_{n\delta_a} + C_{n\delta_a\alpha}\alpha)\delta_a + (C_{n\delta_r} + C_{n\delta_r\alpha}\alpha)\delta_r \end{aligned} \quad (12.35)$$

where α is the angle of attack, β the angle of sideslip, p^* and r^* the normalized roll and yaw rates, δ_a the aileron deflection, δ_r the rudder deflection, and $C_{(.)}$ the unknown aerodynamic parameters. Nonlinearities in the weathercock stability $C_{n\beta NL}$ are modeled as follows by two slopes and an angle α_S , which is also treated as unknown:

$$C_{n\beta NL} = \begin{cases} C_{n\beta 1} & \text{for } \alpha > \alpha_S \\ C_{n\beta 2} & \text{for } \alpha \leq \alpha_S \end{cases} \quad (12.36)$$

A similar formulation is also applied to the yaw damping parameter, $C_{nr NL}$. Figure 12.37 shows a comparison of the analytical/wind-tunnel (WT) predictions with the flight estimated weathercock stability.⁴ Thus, in both the cases systematic approach and judicious choice of flight maneuvers led to unmasking of the model deficiencies and then improvements through model extensions and parameter estimation to a model valid over the entire operational flight envelope.

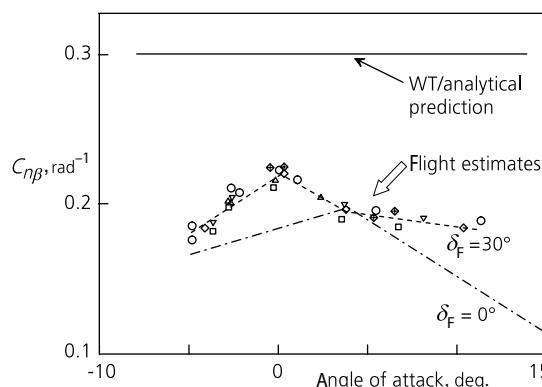


Fig. 12.37 Transall C-160 Weathercock stability.

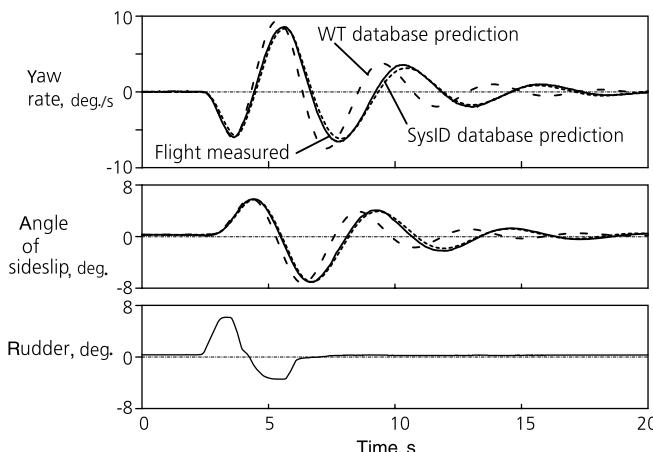


Fig. 12.38 Validation test for the Dutch roll dynamics of Transall C-160.

The fidelity of the final model was demonstrated on roughly 110 proof-of-match tests defined by FAA.^{19,20} We have already addressed this aspect in Chapter 11, discussing the tolerances and the procedure to be followed to run such verifications. As a simple example of verification of an aerodynamic database generated from flight data, we evaluate the validation test 2d7.11, pertaining to the Dutch roll. Simulations using the predictions and estimated parameters are compared in Fig. 12.38 with the flight recorded aircraft response. The WT-predictions yield Dutch roll dynamics having a period of 4.18 s and a damping of 0.207; and the flight-estimated database yields a period of 5.04 s and 0.202 damping factor. A comparison of these values with those approximated from a flight recorded response (period of 5.12 s and damping of 0.198) clearly demonstrates that only the flight estimated database meets the FAA tolerances of ± 0.5 s or $\pm 10\%$ of period and ± 0.02 on damping for this oscillatory mode.⁴

B. Dornier 328 Regional Transport Aircraft

For aircraft such as the Dornier 328 with reversible flight controls, in addition to rigid-body aerodynamics it becomes necessary to identify hinge moments for each of the controls. We briefly touched upon the aspects of model validation for aircraft with mechanical controls in Chapter 11, Sec. VI, and briefly elaborated the concepts of stand-alone and integrated models in Fig. 11.4. In the case of the high performance Dornier 328 (see Fig. 12.23), a high-wing, high-tail regional transport turboprop aircraft, roughly 1200 flight maneuvers were analyzed to derive a rigid-body aerodynamic database.³³ It includes the high angle of attack regime with stall hysteresis, large steady sideslip effects, ground effects, and those due to control surface malfunctions. Nonlinear influences

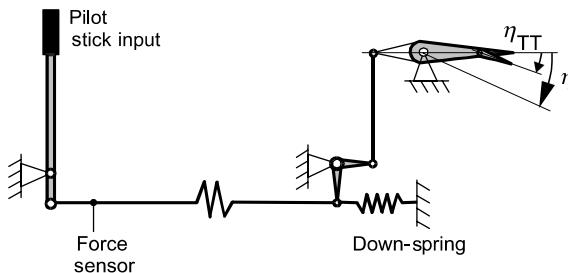


Fig. 12.39 Schematic of elevator control system.

due to the propeller slipstream were found to be very dominant. The same set of dynamic maneuvers is analyzed to derive dynamic models including friction and hinge moment databases for the three primary flight controls.⁵⁴ Since the development of the rigid-body aerodynamic model was based on the approach already covered in the previous section, we concentrate in this section on the hinge-moment database, and few more validation tests. As a typical example, modeling of the elevator flight control is briefly presented here.^{4,54}

In the elevator control system, the pilot forces are transmitted from the stick to the elevator by cables and rods; see Fig. 12.39. The system is preloaded by a down-spring. For the dynamic model, the mass and moments of inertia of the connecting rods and cables are lumped together with those of the elevator surface (including trim tab) into an equivalent moment of inertia Θ_{elv} . As the pilot force is measured in the first rod, the stick inertia effects are already included in this input signal. This leads to a quasi-static transmission of the input force to the elevator axis. The elevator dynamics can, therefore, be formulated as a one degree-of-freedom system using deflection and deflection rate about the swivel axis as system states:⁵⁴

$$\ddot{\delta}_{\text{e}} = \frac{1}{2}(ME_{\text{elv}} + MF_{\text{elv}})/\Theta_{\text{elv}} \quad (12.37)$$

where δ_{e} is the elevator deflection, MF_{elv} the lumped friction moment of the entire elevator control path, and ME_{elv} the sum of all external moments consisting of aerodynamic hinge moment MH_{elv} , forces transmitted to the aft part, inertia effects due to dynamic motions of the aircraft, and down-spring moment.

The aerodynamic hinge moment MH_{elv} is given by

$$MH_{\text{elv}} = \frac{1}{2}CH_{\text{elv}}\bar{q}S_{\text{elv}}\bar{c}_{\text{elv}} \quad (12.38)$$

where \bar{q} is the dynamic pressure, S_{elv} the elevator surface area, and \bar{c}_{elv} the elevator reference chord. Without listing all parameters explicitly, the elevator

hinge moment coefficient was identified as

$$\begin{aligned}
 CH_{\text{elv}} = & CH_{\text{elv},0}(\alpha_{\text{elv}}, Ma, C_{\text{SP}}, \delta_F) \\
 & + CH_{\text{elv},\dot{\delta}_e}(\alpha_{\text{elv}}, Ma, C_{\text{SP}}, \delta_e, \dot{\delta}_F) \\
 & + CH_{\text{elv},\dot{\delta}_{\text{TT}}}(\alpha_{\text{elv}}, Ma, \dot{\delta}_{\text{TT}}, \dot{\delta}_F) \\
 & + CH_{\text{elv},\dot{\alpha}}(\delta_F)\dot{\alpha}_{\text{elv}}^* + CH_{\text{elv},\dot{\delta}_e}(\delta_F)\dot{\delta}_e^* \\
 & + CH_{\text{elv},\beta}(\beta_{\text{elv}}, \delta_F) + CH_{\text{elv},\text{GE}}(\delta_e, \delta_F)\sigma_{\text{elv}}
 \end{aligned} \quad (12.39)$$

where the subscript elv denotes variables related to the elevator and the asterisk normalized rates.

Without going into too much detail, it would suffice to say that the elevator hinge moment coefficient is a function mainly of elevator deflection δ_e , trim tab deflection δ_{TT} , slipstream thrust coefficient C_{SP} , Mach number Ma , flap position δ_F , and local flow angles α_{elv} and β_{elv} . The force gradient $CH_{\text{elv},\dot{\delta}_e}$ is modeled separately for positive and negative deflections. An angle of attack rate influence $CH_{\text{elv},\dot{\alpha}}$ was found and aerodynamic damping is provided by an elevator rate term $CH_{\text{elv},\dot{\delta}_e}$. The elevator hinge moment for symmetrical flight was identified by evaluating dynamic maneuvers (elevator doublets and push–pulls), trims at normal, forward, and aft CG-positions, and elevator mistrims. About 60 maneuvers at different speed and thrust levels were analyzed simultaneously for each flap position. The influence of sideslip was identified from steady sideslip maneuvers and no additional modeling was required for single engine cases.

The identification of the rigid-body aerodynamics and of the hinge moments carried out based on the analytical model postulates yields four sets of derivatives for four flap positions which are finally combined in a single database incorporating a linear interpolation procedure for flap position only. This so-called derivative-model is, however, rarely used in the flight simulators, mainly because the database in the so-called table-model form was hitherto mostly generated from wind-tunnel measurements. Following the conventional way in the present case, as desired by the aircraft manufacturer, the flight identified database is converted into a table model. In this form the rigid-body aerodynamics database consists of 124 tables out of which 14 are five-dimensional, 28 are four-dimensional, 45 are three-dimensional, 27 are two-dimensional, and 10 are one-dimensional functions. The elevator, aileron, and rudder hinge moments require just 8, 18, and 27 tables, respectively. Although the derivative model is better suited to parameter estimation; keeping track of a large number of derivatives being estimated is not easy, particularly those representing the secondary effects. Furthermore, over-parametrization of the model, although leading to a better response match, is detrimental to identifiability of parameters and can lead to unrealistic estimates.

The two alternative model forms, the derivative model and the table model, are equivalent in terms of accuracy, but the derivative model is significantly smaller in size and computationally faster than the table model because the linear interpolation is reduced to that for just one independent variable. The

stand-alone rigid-body aerodynamics in the table model form required about eight times the computational time compared with the derivative model, the ratio becoming as large as 12–14 for the integrated model using both the rigid-body aerodynamics and the hinge moments. Although as realized in the present case, implementation of computationally complex large databases is feasible with modern computers, the differences in the two approaches are significant and pave the way to considering alternative implementations of the flight data-identified models, that is, a derivative model, in flight simulators.³³

Apart from the 170 flight maneuvers specified in the ATG covering part of the flight envelope, a large number (roughly 1100) of additional in-flight maneuvers in ground effect at various altitudes above ground for different power settings under symmetric and asymmetric (one-engine) flight conditions were evaluated to establish global validity of the model. In each of these cases, an end-to-end simulation was performed, incorporating force-driven flight controls, providing position-driven rigid-body models, and incorporating the identified rigid-body and hinge moment databases. Two selected critical examples which highlight the different features of the database are presented here: 1) steady-state sideslip maneuver and 2) critical engine failure on takeoff.^{4,33}

The first example pertains to flight maneuvers at large sideslip angles and serves the purpose of exhibiting correct reproduction of the inter-relationship of steady-state lateral and directional flight characteristics. Figures 12.40 and

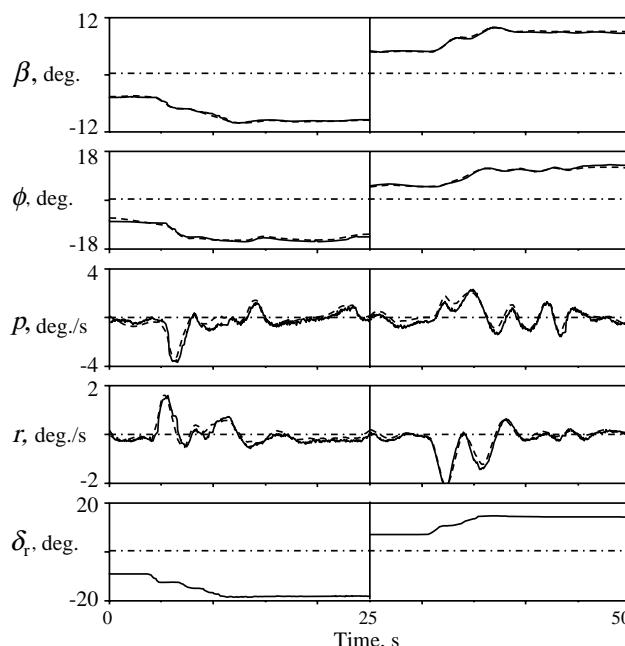


Fig. 12.40 Stand-alone rigid-body aerodynamics (—, flight measured; ----, model estimated).

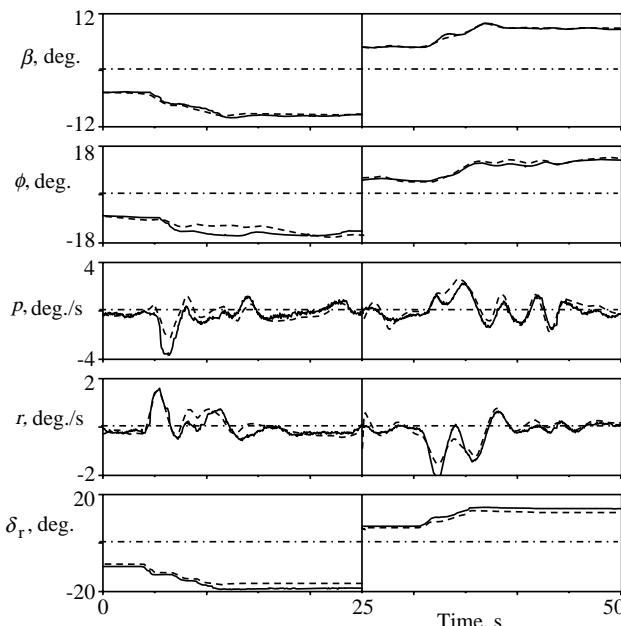


Fig. 12.41 End-to-end match with integrated model (—, flight measured; - - -, model estimated).

12.41 show typical variables pertaining to the lateral-directional motion for two time segments with negative and positive angles of sideslip for an approach configuration of 20 deg. flaps with landing gear down; β is the angle of sideslip, p the roll rate, r the yaw rate, ϕ the bank angle, and δ_r the rudder deflection. Fairly large rudder deflections of about 20 deg. were applied, resulting in steady sideslip up to 11 deg. Each maneuver is of 25 s duration and covers both the transition phase from one level of sideslip to another as well as trim phases. Figure 12.40 shows the match between the flight measured and model-predicted responses for the stand-alone rigid-body model (i.e., six degree-of-freedom equations of aircraft motion excited through the flight measured control surfaces including the identified rigid-body aerodynamics). The exceptionally good match in this case and also that in Fig. 12.25 for the stall maneuver demonstrate the excellent fidelity of the rigid-body aerodynamic database.

For the same case, Fig. 12.41 shows the end-to-end match with the integrated model (i.e., six degree-of-freedom equations of aircraft motion coupled with dynamic models of the flight controls incorporating identified databases for rigid body and hinge moments). The distinction between the two evaluations shown in Figs. 12.40 and 12.41 is apparent from the bottom-most plot, which shows just the measured rudder deflection in Fig. 12.40 as an input to the model, whereas in Fig. 12.41 the model predicted rudder deflection is compared with the flight measurement. In Fig. 12.41 the computed control deflections are then the inputs to the rigid-body aerodynamics. Although some deterioration in

the end-to-end match is observed in Fig. 12.41 compared with Fig. 12.40, the overall match between flight measured and model predicted variables is good and within the FAA-tolerances of ± 2 deg. for bank angle and ± 1 deg. for side-slip; the aileron deflection not shown here was within $\pm 10\%$ or ± 2 deg., the wheel force bias within ± 1.3 daN and the pedal force bias within ± 2.2 daN. It may be pointed out that it was necessary to account for asymmetric propeller effects, particularly on the vertical tail, at such extreme flight conditions.

The deterioration of the match in Fig. 12.41 compared with Fig. 12.40 clearly shows the difficulties faced by the database developers in demonstrating the end-to-end match obtained by cascading two highly complex models, although each may have an excellent fidelity, as shown in Fig. 12.40 for rigid-body. Any attempts to further refine the stand-alone models appear unrealistic.³³ In the worst case, and if critical, some tuning may be possible using the integrated model.

The second example is the validation test, generally denoted 1b5, called critical engine failure on takeoff, and pertains to simulation of the response to an engine failure during the high activity task of takeoff; responses to rudder and aileron are of particular interest. Figure 12.42 shows the results for a few

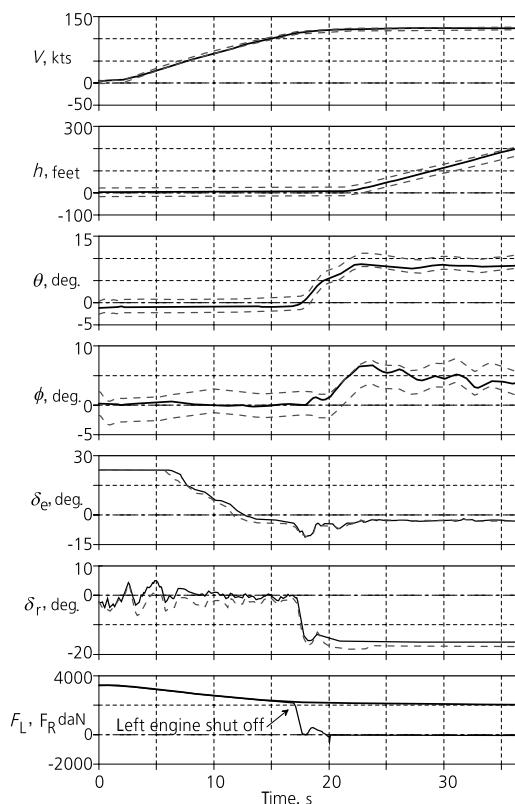


Fig. 12.42 Proof-of-match for critical engine failure on takeoff (-----, flight measured \pm tolerances; ——, model estimated).

SELECTED ADVANCED EXAMPLES

447

pertinent variables. The complete sequence from stand-still, acceleration, rotation, and climb to 200 ft above ground is simulated as a single maneuver. The results were obtained without any closed-loop controller (also termed math pilot). The use of closed-loop controller helps in such demanding validation tests to meet the required tolerances. They should, however, be used with caution, because they mask the modeling deficiencies.⁵⁵

From the bottom-most plot in Fig. 12.42 (F_L and F_R , the thrust due to the left and right engines), it is observed that the critical left engine is shut down immediately after rotation. To maintain the bank and heading the pilot applies about 15 deg. rudder, δ_r in Fig. 12.42, and about 6 deg. ailerons, not shown. Except for a small lead in building up the bank angle, which is still within the tolerance, the match for all the variables is good. Specifically, the airspeed is within the tolerance of ± 3 kts, altitude within ± 20 feet, pitch angle within ± 1.5 deg, and bank angle within ± 2 deg. Thus, the database meets the FAA fidelity requirements.

As mentioned at the outset of this section, the task of generating the databases was highly complex and iterative in nature. In general, the primary aerodynamic derivatives can be extracted well; estimation of the secondary effects needs some careful choice of the maneuvers. In particular, in such cases care needs to be taken to avoid over-parameterization, which yields better response match but not necessarily physically justifiable estimates. It needs to be ensured that the identified model is parsimonious. Analytical models for highly complex phenomena such as stall hysteresis and ground effects have proved very efficient from a parameter estimation viewpoint and also from the ability to capture the physics of the phenomenon. Based on these investigations, as a rule of thumb, it can be said that, depending upon the type of aircraft (jet or propeller; with or without reversible flight controls), roughly 1000 maneuvers or 30–50 flight hours are required to generate high-fidelity databases suitable for level D training simulators. Similar rough estimates have been reported from other investigations as well.⁵⁶

These two successful applications of high-fidelity database generation and other similar cases reported in the literature are the high-end applications of the system identification methodology. Of course, such exercises are not without intricacies. In this connection, the lessons learned from these two exercises, reported in Ref. 57, are worth repeating here:

- 1) The development of an integrated model is much harder than the development of stand-alone models.
- 2) The amount of effort required to arrive at a suitable model depends on the types of maneuver and on the propeller slipstream interference effects.
- 3) The development of an FAA-adequate friction model for flight control systems seems to be impossible (for pilot force-driven end-to-end testing).
- 4) A gradient-based optimization method is not quite suitable to handle systems with static friction in an identification context.
- 5) The direct identification approach works, but there are some inherent, practical limitations.

To summarize, validation and if necessary update of existing aerodynamic databases from flight test data, as well as generation of completely new models, has been one of the most extensive and important applications of aircraft parameter estimation in the past. As pointed out in Ref. 1, this will continue to be one of the major applications of system identification in the future.

IX. X-31A Model Validation and Update

The U.S./German experimental aircraft X-31A is a highly control augmented fighter with enhanced maneuverability; see Fig. 12.43. Post-stall maneuvering is enabled by applying advanced technologies such as high angle-of-attack aerodynamics and flight control system-integrated thrust vectoring.⁵⁸ In Chapter 9, Secs. XIV and XVI.B, we addressed the challenges encountered in estimating the aerodynamic derivatives of the basic airframe. To recapture the essentials once again, it was demonstrated that 1) highly correlated control surface deflections, thrust vectors vanes, and states due to integrated flight control laws resulted in poor estimates with large standard deviations and large scatter, 2) owing to data collinearity, it was possible to estimate only combined derivatives from pilot input maneuvers, 3) single surface excitation provided significantly improved estimates, and 4) from a variety of algorithms in the time domain (artificially stabilized output error method, filter error method, equation decoupling, regression analysis, and extended and unscented Kalman filter techniques), the least squares and filter error methods accounting for process noise were found to be preferable.

A detailed description of X-31A system identification is beyond the scope of this book. A thorough treatment of various aspects is found in Refs. 47 and 59. Principally, the results of X-31A system identification have been used to validate and in several cases update the wind-tunnel predicted database. We provide here two typical results. In Fig. 12.44a we find the estimates of the dihedral effect obtained from pilot input maneuvers and those from the SSE inputs, which once again strongly support the conclusion drawn in Chapter 9, Sec. XIV, namely that SSE yields drastically improved estimates. It is also evident that the flight estimates did not confirm the wind-tunnel-predicted large value of the dihedral effect between 30 and 45 deg. of angle of attack. Similarly,



Fig. 12.43 X-31A at 70 deg. angle of attack.

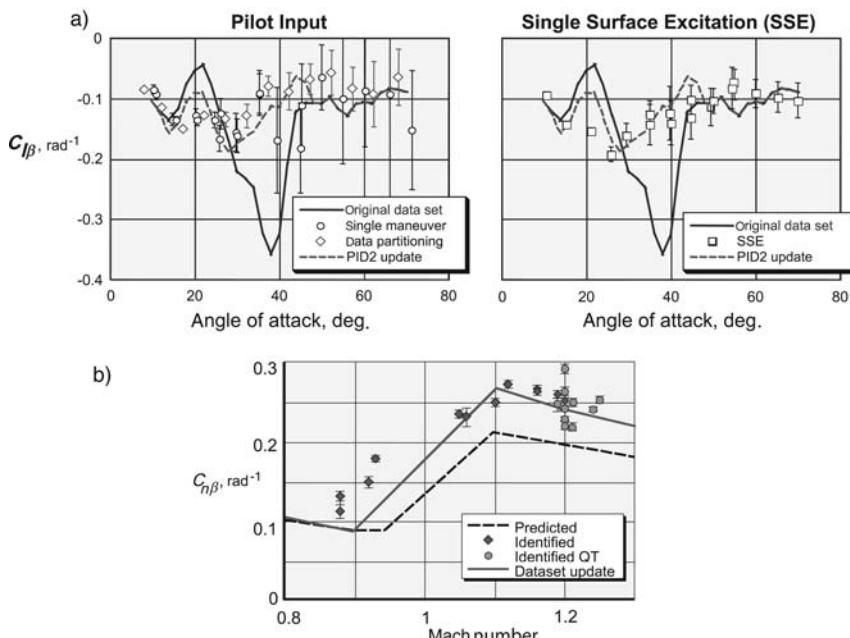


Fig. 12.44 Examples of X-31A database updates. a) Dihedral effect, b) directional stability.

noticeable discrepancies were also observed in the estimates of the directional stability (Fig. 12.44b; QT, quasi-tailless configuration).

In conclusion, diligent modeling and a suitable choice of estimation technique provided flight-validated aerodynamic characteristics, including post-stall regime and thrust vector control. Furthermore, the crux of the problem pertaining to highly augmented aircraft was the data correlation and insufficient excitation, and an efficient solution to this problem was provided by separate surface excitation. System identification techniques provided improved results for flight test planning, flight envelope expansion, and a database for simulation and control law modification and validation.^{2,59}

Two other advanced applications in the area of unstable aircraft and post-stall maneuvering using thrust vectoring are the X-29A and F-18 HARV. The prime research objective of the X-29A program was to test the predicted aerodynamic advantages of the unique forward swept-wing configuration. The aircraft is statically unstable in the longitudinal axis, with a negative static margin of up to 35% at subsonic speeds (time to double of about 0.15 s). In the lateral-directional axes it is statically stable.^{60,61} The F-18 High Angle of Attack Research Vehicle (HARV) with multi-axis thrust vectoring is a widely used test bed for research investigations.^{62,63} In both these cases the estimation of aerodynamic stability and control derivatives of the open loop aircraft posed problems which we discussed earlier in this section and also in Chapter 9. The uncommanded motion

caused by unsteady aerodynamic called for the use of estimation methods accounting for process noise, like least squares or filter error. The high-gain feedback led to correlated motion and control variables, and under such cases the single-surface excitation provided significantly improved estimates of the aerodynamic derivatives. For exact details of these challenging projects, the reader is referred to the respective literature cited above.

X. Wake Vortex Aircraft Encounter Model[†]

Wake vortex encounters are critical situations that may result in flight conditions dominantly affecting the aircraft reaction during safety-critical flight phases. Optimization of air traffic distribution and volume in the vicinity of airports, particularly under a mix of light, medium, and heavy aircraft, calls for better understanding of the basic physics underlying this phenomenon.^{4,64} For this purpose, validated models for aerodynamic flow field characteristics and aircraft wake vortex encounter reaction models are indispensable.

Aerodynamic interaction models (AIM) are an indispensable part of flight mechanic simulation models concerned with *wake vortex encounter* investigations, for example, simulator studies or encounter risk assessment. Two typical AIMs are the strip method and the lifting surface method.^{65,66} Both can be adapted to various aircraft configurations and compute the wake vortex flow-field-induced additional forces and moments that are added to the forces and moments computed by the basic aircraft aero model. For reliable simulation results, the quality of the AIM model is of prime importance. Until now, quality has often been proven by comparing model forces and moments to static wind tunnel measurements, in which the follower aircraft is set to distinct positions behind a wake-generating configuration.

Within the 5th European framework technology project S-WAKE,⁶⁶ one objective was to validate the AIMs using data from *full-scale flight tests*, which were carried out using three aircraft: ATTAS from DLR, a Do-128 of the Technical University of Braunschweig, and a Cessna Citation from NLR, the Netherlands. The task of the leading aircraft, ATTAS (ICAO separation class medium), was to generate and visualize the wake with a wing-mounted smoke generator. The follower aircraft (separation class: light) encountered the ATTAS wake at different, distinct horizontal distances (Fig. 12.45). Both follower aircraft were fully instrumented with inertial sensors and several flow probes in order to measure the flow field characteristics of the wake vortex as well as the aircraft response during each encounter. About 100 encounters were flown under steady atmospheric conditions, and bank angles up to 80 deg. could be observed, with typical bank angles of 30–40 deg. during a lateral encounter. Bank angle is only one aspect of wake encounter. It results in a bump and may be uncomfortable, but only for a short duration and it does not usually lead to loss of control, because the banking motion is averaged out. More important is the lateral acceleration resulting from such an encounter,

[†]Contribution from D. Fischenberg, DLR, Institute of Flight Systems.

SELECTED ADVANCED EXAMPLES

451

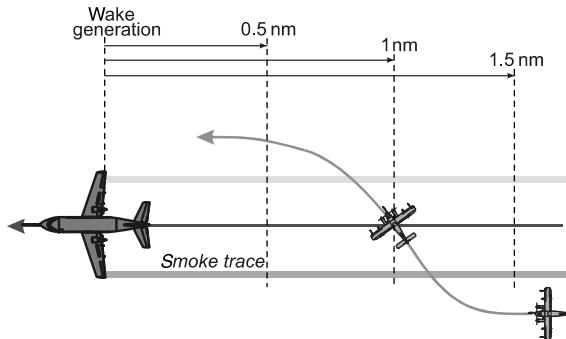


Fig. 12.45 Schematic of encounters flown.

because it may lead to injury to crew or passengers. Similarly, vertical motion is important for separation.

The flight data analysis comprised a two-step procedure, depicted in Fig. 12.46, consisting of 1) flow-field characterization through an analytical wake vortex model incorporated in the flight path reconstruction and estimating the vortex-model parameters, and 2) encounter model validation through simulation, using prevalidated basic rigid-body model and identified wake vortex characteristics in analytical form. The link between the two steps is shown by dotted line in the figure.

The most critical part of this complex task is the determination of the wake model parameters, which comprises four subparts of a single overall computational procedure, shown in Fig. 12.47. During the first part the classical flight path reconstruction (FPR) technique elaborated in Chapter 10, Sec. II with

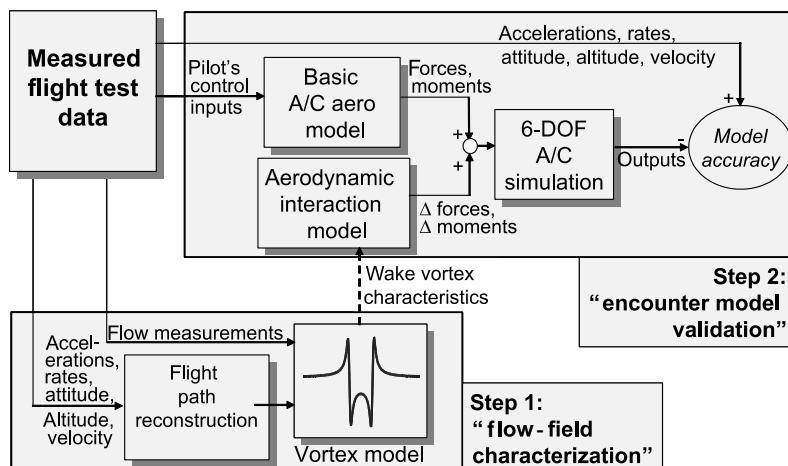


Fig. 12.46 Schematic of two-step procedure for vortex model identification and validation.

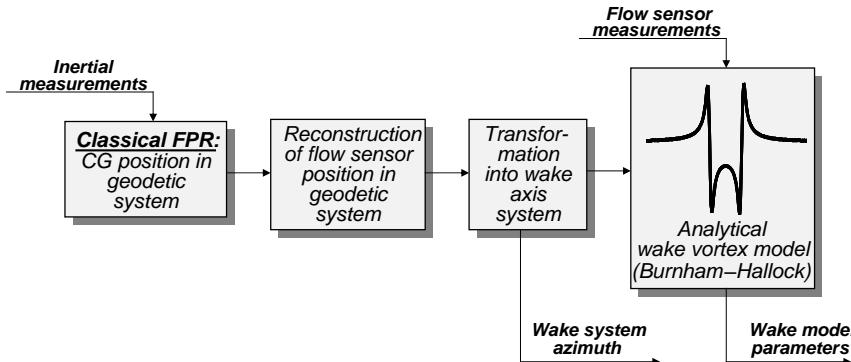


Fig. 12.47 Computational parts of wake model parameter estimation.

$(u, v, w, \phi, \theta, \psi, x_g, y_g, h)$ as the state variables and the measured linear accelerations (a_x, a_y, a_z) and angular rates (p, q, r) as the inputs is applied to obtain CG flight path in the geodetic coordinate system, whereby the input sensor biases are also eliminated. In the second part, the x_g, y_g, h positions of each flow angle sensor are reconstructed. In the third part, these sensor positions are transformed into the wake axis system, which is the same as the geodetic system but twisted through ψ_w , the wake system azimuth angle. In the last part, the resulting sensor positions in the wake $y-z$ plane are used as inputs to the analytical wake model, fitting its outputs to the measured flow sensor information (e.g., angle of attack and sideslip angles) in order to identify the wake model parameters, namely the circulation, core radius, and position of left and right vortex in the wake axis system.

There are two important issues related to the practical application of the above procedure. First, it requires accurately precalibrated measurements of the flow angles. Second, the modeling and identification are very sensitive to the wake system azimuth ψ_w . This azimuth angle cannot be measured, but is roughly equal to the heading of the aircraft generating the wake. Since ψ_w is a critical parameter, which has to be known accurately, it is desirable to identify the same for each individual wake fly through. Identification of ψ_w is possible if flow angle measurements at several different locations, for example nose and wing tips, are available. If flow angles are measured at a single location, which is mostly the case, identification is still possible, but only when the lateral distance between the two vortices is known a priori, which should be $\pi/4$ of the wing span of aircraft generating wake, assuming elliptical wing lift distribution.

A. Determination of Flow-field Characteristics

Precise determination of the flow-field characteristics to which the follower aircraft is subjected to during each of its individual encounters is a prerequisite and critical to aerodynamic encounter model validation. Two different analytical velocity distributions (*Burnham-Hallock* and *Lamb-Oseen*) were investigated

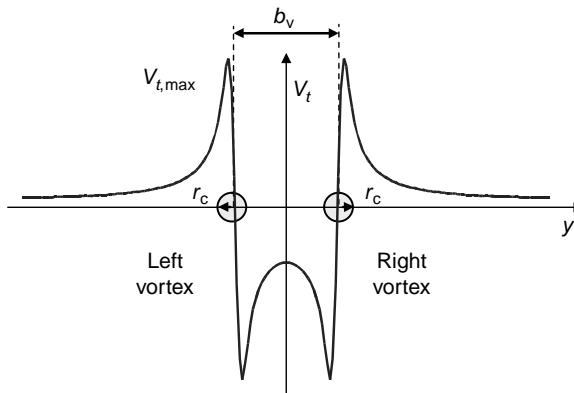


Fig. 12.48 Analytical wake vortex model.

to match the measured flow field characteristics.^{67,68} First, the flight path was reconstructed precisely using system identification techniques. Then, the parameters of the analytical wake vortex model were identified.⁶⁹ The model consists of two idealized, superimposed counter-rotating single vortices; see Fig. 12.48, characterizing the aircraft wake vortex system after wake roll-up. The model parameters are vortex circulation Γ , core radius r_C , lateral vortex separation b_V , and vortex location in space. The tangential velocity of one vortex as a function of the distance from the core, $V_t(r)$, is described in terms of the circulation Γ and the core radius r_C :

$$\text{Lamb-Oseen: } V_t(r) = \frac{\Gamma}{2\pi r} \left(1 - e^{-1.2544r^2/r_C^2} \right) \quad (12.40)$$

$$\text{Burnham-Hallock: } V_t(r) = \frac{\Gamma}{2\pi} \frac{r}{r_C^2 + r^2} \quad (12.41)$$

Both vortex models allow computation of flow measurements for most wake vortex ages. As an example, Fig. 12.49 shows the match of the measured velocity components from three flow probes during a Do-128 lateral wake encounter, 0.53 nm behind ATTAS. In this plot, it can be seen that the encounter is flown from the left to the right, as the right wing sensor enters the flow field first, followed by the center-mounted noseboom sensor and then the left wing sensor. In general, compared with the Lamb-Oseen, the Burnham-Hallock model is slightly better suited to characterize the wake induced velocities, especially close to the vortex core with high gradients.

Figure 12.50 summarizes the identified core radius and circulation of the Burnham-Hallock model for Do-128 encounters from three flights. The estimates are plotted as a function of normalized time $t^* = t/t_0$, which indicates the vortex age (t_0 , ATTAS = 11 s). In conformance with the theory, Fig. 12.50 shows the expected decay of circulation, as well as the increase of the core

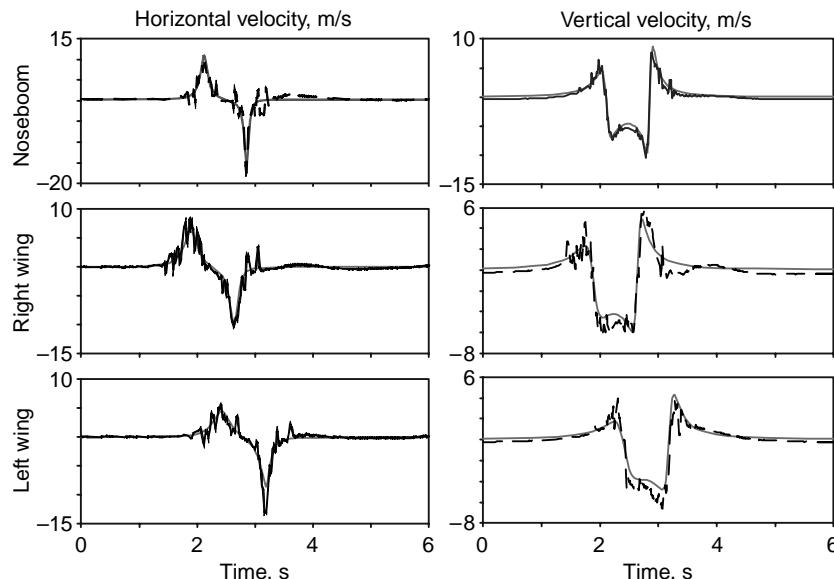


Fig. 12.49 Wake velocity components during lateral encounter (----, measured; —, Burnham–Hallock model).

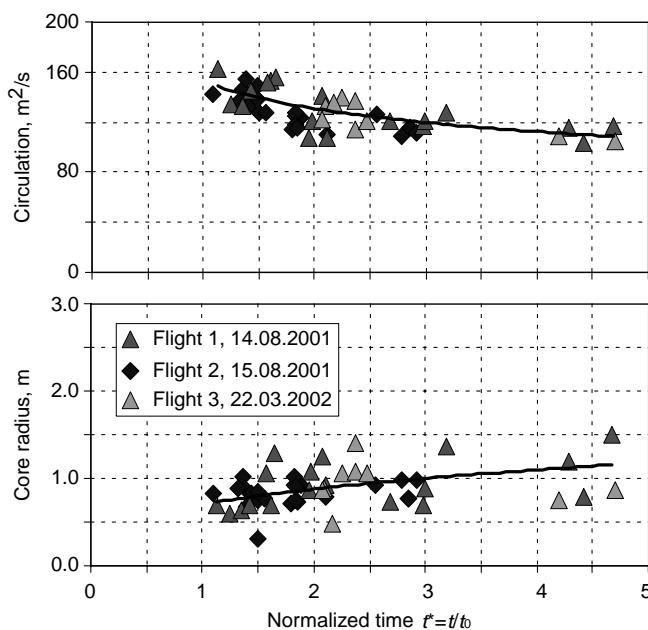


Fig. 12.50 Flight-estimated vortex model parameters for Burnham–Hallock velocity distribution.

radius. The initial core radius is about 0.75 m, which is 3.5% of the wake generating wing span; it is somewhat smaller than the commonly stated value of 5%.

B. Validation of Aerodynamic Interaction Model

Validation of the *strip method*, accounting for incremental aerodynamic forces and moments induced by the wake vortex flow field, is presented here.⁶⁵ These increments are added to the basic aircraft forces and moments. For validation, the above determined vortex characteristics and the measured control surface deflections were used as inputs.⁶⁶ In this case too, the general procedure elaborated in Chapter 11, Sec. VI was applied, allowing small biases in the control deflections; the initial conditions of the attitude angles were set to the measured values and for the other states small biases were permitted. For the Do-128 aircraft, the wing, and the horizontal and vertical tail are divided into total 28 strips: wing, 16; horizontal tail, 8; vertical tail, 4. Flow-field-induced incremental angles of attack and therefrom resulting incremental lift are computed for each strip. Integrating all the single strip forces, the total forces and the moments are obtained.

In Fig. 12.51, typical model outputs are compared with the corresponding measured values. The most important model outputs for risk assessment, that is, vertical acceleration, roll rate, and bank angle, are computed with good accuracy. The match for the pitching motion is also found to be adequate. Some systematic deficiencies are observed in the side force (lateral acceleration) and yawing motion, probably due to missing fuselage effects in the strip model. The simple strip method seems to be fairly adequate for flight mechanic encounter modeling. It should be noted that these results were obtained with a relatively simple aircraft configuration (rectangular wing planform). Aircraft with wing sweep, for example, may be simulated with lower quality using this method.

XI. Phoenix RLV Demonstrator

The Phoenix flight test vehicle was developed and tested within the framework of the German ASTRA (Advanced Systems & Technologies for RLV Applications) Program, which is a precursor to the European Future Launcher Preparatory Program (FLPP).^{70,71} The primary objective of the Phoenix project was to demonstrate the unpowered automatic landing of a representative reusable launch vehicle (RLV) configuration. The secondary goal was to generate a flight-validated database incorporating representative models of such vehicles and their systems as well as development tools for future applications. The project had its kick-off in April 2001 and reached its climax with the flight test campaign in May 2004. Although a detailed discussion of several interdisciplinary topics is not within the scope of this book, for the specific aspects of flight data analysis presented in this example, we draw heavily upon the more detailed discussion in Refs. 72 and 73.

RLV configurations compromise between partly contradicting demands derived from various system optimization aspects as well as the different flight phases covering ascent, orbital phase, reentry, hypersonic flight, and landing. The resulting vehicle typically is characterized by a compact shape with small span wings and stabilizers, and a rearward center of gravity location. Such

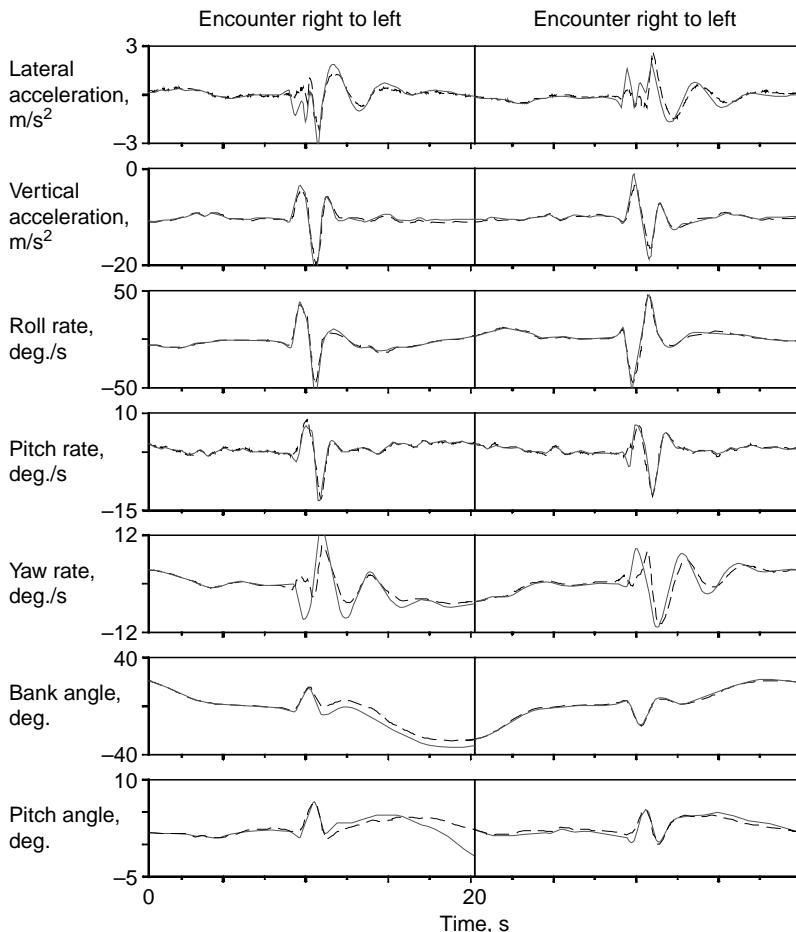


Fig. 12.51 Strip method validation of two typical Do-128 lateral encounters (—, flight measured; - - -, simulation model).

properties challenge an automatic horizontal landing on a runway, which in addition must be performed without engine thrust. The important characteristics and their impact on the landing phase can be summarized as follows: 1) low L/D (5.5 for Phoenix), causing steep approach path; 2) low achievable C_L , causing high landing velocity (71 m/s for Phoenix); 3) small span, causing high roll sensitivity; and 4) aft CG position, causing a statically unstable configuration (time to double amplitude <0.5 s for Phoenix). The test vehicle shape was derived from the suborbital RLV concept Hopper, one of the three RLV concepts currently pursued in Europe. Besides GNC (Guidance, Navigation, and Control) and avionics systems for autonomous approach and landing, it is equipped with flight test instrumentation to measure local aerodynamic flow and structural stress.

Two other projects, namely X-40A and ALFLEX, have also demonstrated successful free-flight and automatic landing experiments in the past.^{74,75} Compared with Phoenix, the ALFLEX vehicle was smaller and lighter, with lower wing loading, resulting in lower landing speed, and did not have an extendable landing gear system. Moreover, it was equipped with a ground-supported microwave-based landing system. The Phoenix configuration is more demanding and incorporates a fully autonomous DGPS-based autoland system, which is more advanced. Three free flights, launched from a carrier helicopter at 2.4 km altitude, were successfully completed in May 2004; see Fig. 12.52. The data obtained from these tests established the source for the verification of the vehicle model. In particular, methods of system identification were applied to upgrade the aerodynamic and the wheel-ground contact and ground roll model.

A. Final Configuration and Sensors

The Phoenix vehicle originated from the FESTIP (Future European Space Transportation Investigation Program) Hopper shape, but was downscaled by a factor of 7. Wing loading and CG location were retained to keep the same absolute speed and aerodynamics as well as the unstable configuration of the full scale vehicle. Thus, all technological challenges to master the automatic landing of such shapes are preserved. Iterative wind-tunnel test campaigns, including full-scale vehicle tests in DNW-LLF (German–Dutch Wind-Tunnel, Large Low-speed Facility) resulted in the final airframe design and aerodynamic database.^{73,76} The final airframe consisted of a conventional aluminum structure with several composite parts, such as covers for the fuselage, the elevons and the bodyflap. The basic dimensions are overall length including nose boom 7.8 m, span 3.84 m, overall height with landing gear retracted 2.56 m, mass 1200 kg, and center of mass at 70% of the bare fuselage length from the nose. Electromechanical rotary actuators drive the aerodynamic control surfaces and the nose gear steering. The landing gear is a tricycle configuration with telescopic deployment mechanism and oleo shock absorbers. The main gear wheels are equipped with brakes.

Phoenix is equipped with an extensive Flight Test Instrumentation system. Besides measuring all the control deflections (inboard elevons, outboard elevons, rudder, body-flap, and speed brakes), landing gear and nosegear positions, variables pertaining to aircraft motion, like accelerations, angular rates, velocity, attitude, and position were measured using an integrated GPS/INS platform. A five-hole probe mounted on a noseboom provided the measurements of flow variables (angle of attack, angle of sideslip, and dynamic pressure). The static pressure and total air temperature sensors were also mounted on the boom. Differential GPS, radar altimeter and laser ranger were used to improve the achievable height accuracy at different segments of altitudes and height above runway.

The five-hole probe providing the measurements of feedback variables for the flight control was calibrated during the full-scale wind-tunnel test campaign prior to free flights. Three nominal speeds covering all configurations were investigated. Applying estimation algorithms, calibration factors, and bias parameters, including time delays, were determined. A linear calibration factor for angle of attack yields the error between the probe measured and the nominal angle shown in Fig. 12.53. The nonlinear nature of the error necessitated a nonlinear

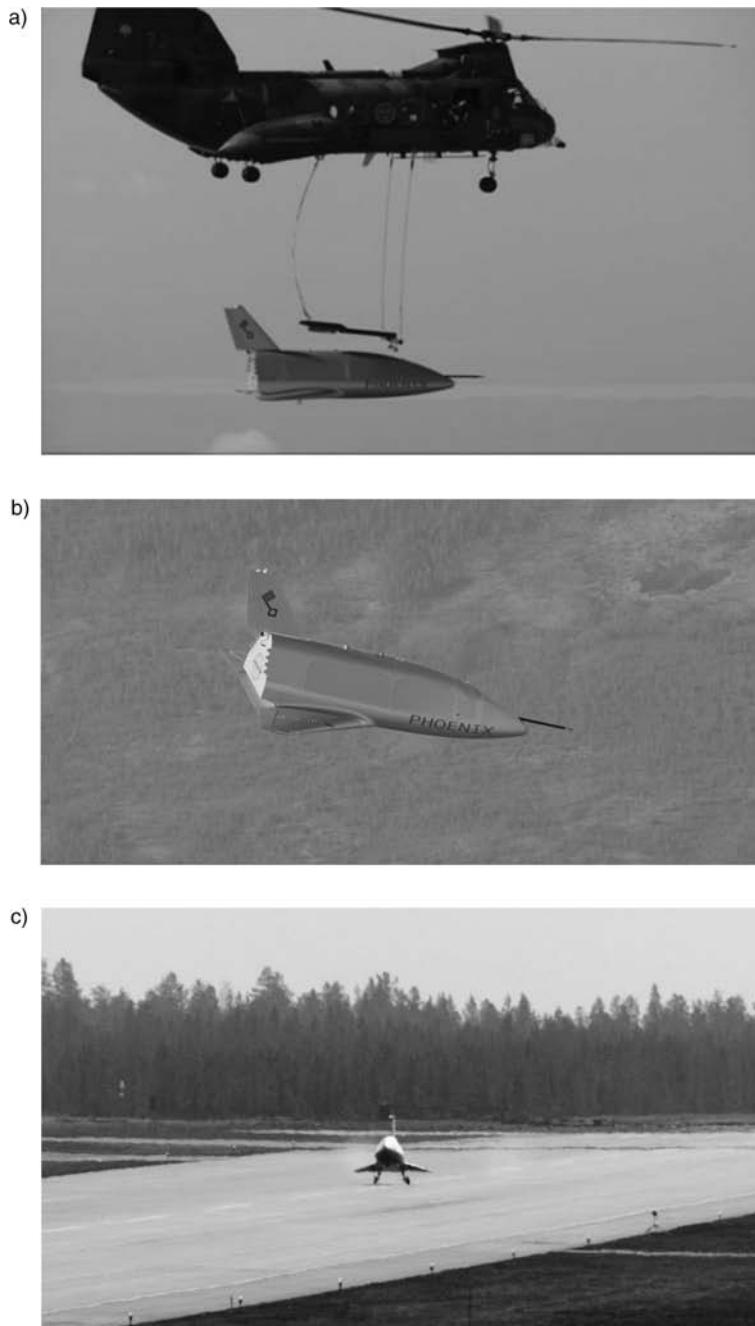


Fig. 12.52 Phoenix RLV demonstrator free flight. a) Phoenix release from carrier helicopter, b) Phoenix in free flight, c) Phoenix touchdown.

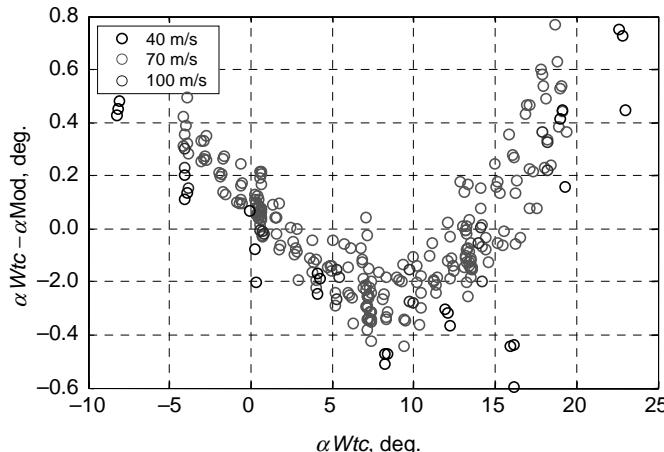


Fig. 12.53 Errors in linear calibration of angle-of-attack.

calibration factor to achieve the desired accuracy of better than 0.5 deg. throughout the range; moreover the measurement of the angle of attack was also found to be a function of the sideslip angle. Correction factors were also necessary for the angle of sideslip. Likewise, measured dynamic pressure had to be corrected for errors in the static pressure probe.

B. Basics of Aerodynamic Model Validation and Update

The two-step procedure illustrated in Fig. 12.32 is adopted to verify and update the aerodynamic model from free flight data. The computational steps have already been elaborated in Sec. VIII. Besides the two characteristics mentioned there, namely 1) direct comparison of flight determined force and moment coefficients with the wind-tunnel predictions, and 2) correction terms can be identified as deltas, in the present case the use of the regression method is preferable for the additional reasons of its applicability to unstable configurations, and to account for atmospheric turbulence in the estimation.

Following the procedure described in Chapter 6, Sec. IX.A, the aerodynamic force and moment coefficients were computed from measured body-fixed accelerations and the angular accelerations. The angular accelerations ($\dot{p}, \dot{q}, \dot{r}$) were obtained by numerical differentiation of the measured angular rates. Implicit to this data preprocessing step, sensor positions and CG positions have to be accounted for while computing the force and moment coefficient referred to the aerodynamic reference point. This data pre-processing step, depicted in the upper part of Fig. 12.32, yields flight-derived force and moment coefficients (CX, CY, CZ, CLX, CMY, CNZ). Special emphasis was also placed on reverification of the wind-tunnel calibration of the airflow data, namely angle of attack, angle-of-sideslip and dynamic pressure measured by the five-hole probe, and also of the static pressure probe on the noseboom. The flow variables were verified through flight path reconstruction applied to free-flight data. The wind-tunnel

calibration of the five-hole probe measured dynamic pressure required about 10.2% correction, which was found to result from the erroneous measurement of the static pressure measured by a probe on the noseboom. Similar corrections have also been reported in other investigations.⁷⁷ The plausibility of this correction in the static pressure was verified by comparison with balloon measurements taken just prior to the free-flight.

C. Comparison of Preflight ADB and Flight-derived Coefficients

Three free flights performed on 8, 13, and 16 May 2004 are analyzed here. The attention is restricted first to the in-air aerodynamic. In order to facilitate verification of landing gear effects separately, the first part of data analysis was based on 50 s long time segments corresponding to the portions starting just after the drop up to just before the landing gear is extended. The guidance maneuver in the longitudinal axis results in a fair amount of pitching motion. The altitude above ground varies from 2400 m to about 120 m.

Applying the procedure depicted in Fig. 12.32, aerodynamic force and moment coefficients according to preflight aerodynamic database (ADB) predictions (version V3.1)⁷⁶ and those from the flight-measured accelerations and angular rates are computed and compared in Fig. 12.54. The table-lookup database and application rules as derived from wind-tunnel and other computational investigations were taken over without any modifications.⁷⁸ The vertical lines at 50 and 100 s demarcate the three flights. The total coefficients CX and CZ are plotted in Fig. 12.55 as functions of angle of attack. From Figs. 12.54 and 12.55, it is evident that there are some discernible differences between the flight derived and wind-tunnel-predicted forces and moments pertaining to the longitudinal motion. Such a comparison was not possible for (CY, CLX, CNZ), because there was no dynamic motion of the lateral-directional mode.

The rough order of magnitude of discrepancy in the vertical force coefficient (lift curve slope) was observed to be of the order of 9–10%. Briefly, lift generated in flight is higher than predicted and the component due to pitch rate q is not adequately accounted for in the pre-flight ADB. The discrepancies are less than about 3% in the pitching-moment curve slope. Owing to the nonlinear relationship, the error in the axial force coefficient CX could not be quantified as in the other cases. From Figs. 12.54 and 12.55, the error in CX is evident for the clean configuration (initial portion of each segment); with increasing speed brakes the mismatch between flight test and preflight database decreases, implying that the basic longitudinal force coefficient was underestimated in the wind-tunnel, whereas the impact of the speed brakes was overestimated. The preflight ADB does not account for drag due to pitch rate, which seems not to be negligible. The aerodynamic zero terms CX0, CZ0, and CMY0 also show deviations. The magnitudes of the deviations are typical of the two approaches, namely wind-tunnel and flight test-derived information.⁷⁹

Through a systematic approach of evaluating multiple cross plots, a better appreciation of a possible dependence of these deviations, the so-called deltas (i.e., the deviations between the flight derived values and pre-flight predictions), on different motion variables was acquired. Thus, an engineering approach was

SELECTED ADVANCED EXAMPLES

461

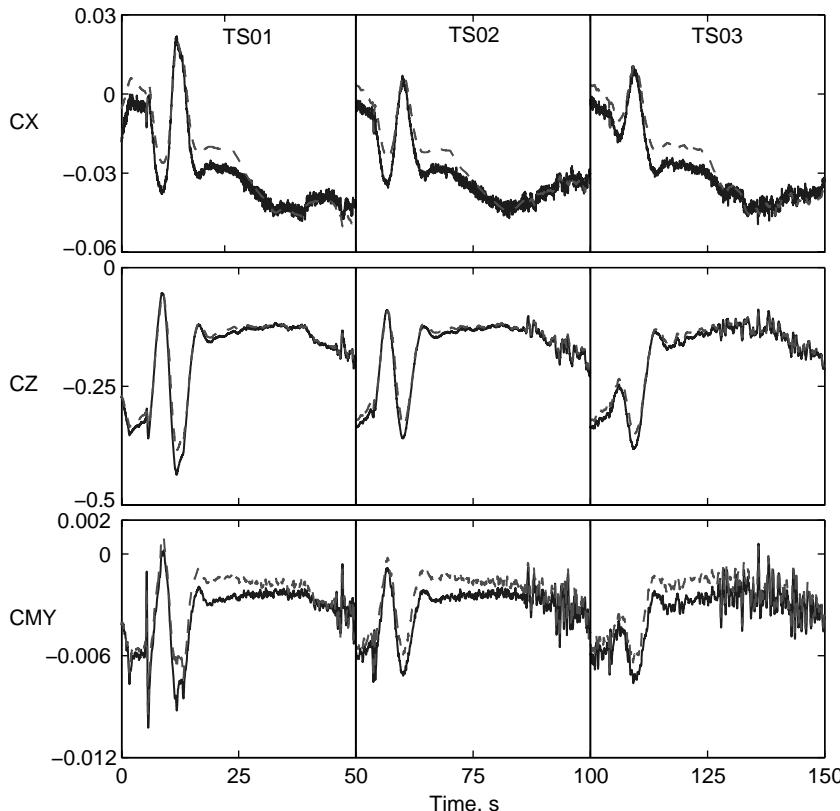


Fig. 12.54 Comparison of aerodynamic force and moment coefficients (----, wind-tunnel predicted; —, flight derived).

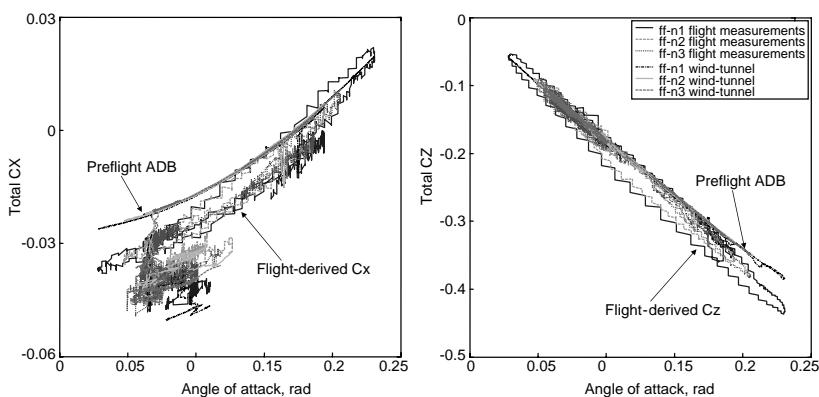


Fig. 12.55 Comparison of wind-tunnel-predicted and flight-derived aerodynamic force and moment coefficients.

adopted to determine the primary influencing parameters, rather than the rigorous stepwise regression procedure. The deviations in CX were found to be functions of angle of attack, pitch rate, and speed brake; those in CZ were found to be dependent on angle of attack, pitch rate, and body flap deflection; and those in CMY were observed to be dependent on angle of attack and body flap.

D. Identification of Deltas for Force and Moment Coefficients

Based on the inferences drawn from the comparison, the updates for CX, CZ, and CMY are identified by applying a stepwise procedure. The resulting model in terms of derivatives turned out to be as follows:

$$\begin{aligned}\Delta CX &= CX_0 + CX_\alpha \alpha + CX_q qV/L_{ref} + CX_{\delta_{sb}} \delta_{sb} \\ \Delta CZ &= CZ_0 + CZ_\alpha \alpha + CZ_q qV/L_{ref} + CZ_{\delta_{bf}} \delta_{bf} \\ \Delta CMY &= CM_0 + CM_\alpha \alpha + CM_{\delta_e} \delta_e + CM_{\delta_{bf}} \delta_{bf}\end{aligned}\quad (12.42)$$

where δ_{bf} is the body flap deflection, δ_{sb} the speed brakes, and L_{ref} the reference length. The parameters CZ_α and CM_α are modeled nonlinearly, having one value for angle of attack up to 0.175 rad and another value for higher angle of attack. Parameters $(CX_0, CX_\alpha, CX_q, CX_{\delta_{sb}})$, $(CZ_0, CZ_\alpha, CZ_q, CZ_{\delta_{bf}})$, and $(CM_0, CM_\alpha, CM_{\delta_e}, CM_{\delta_{bf}})$ are estimated.

The results of parameter estimation are plotted in Fig. 12.56 showing the force and moment coefficients.⁷³ A comparison of Figs. 12.54 and 12.56 clearly shows the improvements obtained through the updates given in Eq. (12.42). It is observed that the residuals, that is, the deviations between the flight-derived coefficients and those computed with updates are no longer deterministic in nature, and now represent the measurement noise. The cross plots of total coefficients CX and CZ as a function of angle of attack showed good agreement. It is necessary to mention here that, during this phase of flight, the speed brakes and body flap are deflected simultaneously and accordingly they are correlated. As such, from the available flight data it is not possible to identify the components in CX due to speed brakes and body flap separately. The correction term is modeled with respect to speed brake only.

E. Landing Gear Effects

As the next step of data analysis, the in-air aerodynamic database, updated from flight segments in which the landing gear was not extended, is now used to simulate the complete time segment including the landing gear extension phase, down to roughly 3–4 m above ground level. The final 3–4 m before touchdown is not included in the analysis to avoid ground effect, which is dominant for heights up to about span width. Under this consideration any deviations that may be observed during the segment with landing gear can be attributed to landing gear effects. Each time segment is now 64 s long, the first 50 s being the same portion that was analyzed to update the preflight database.

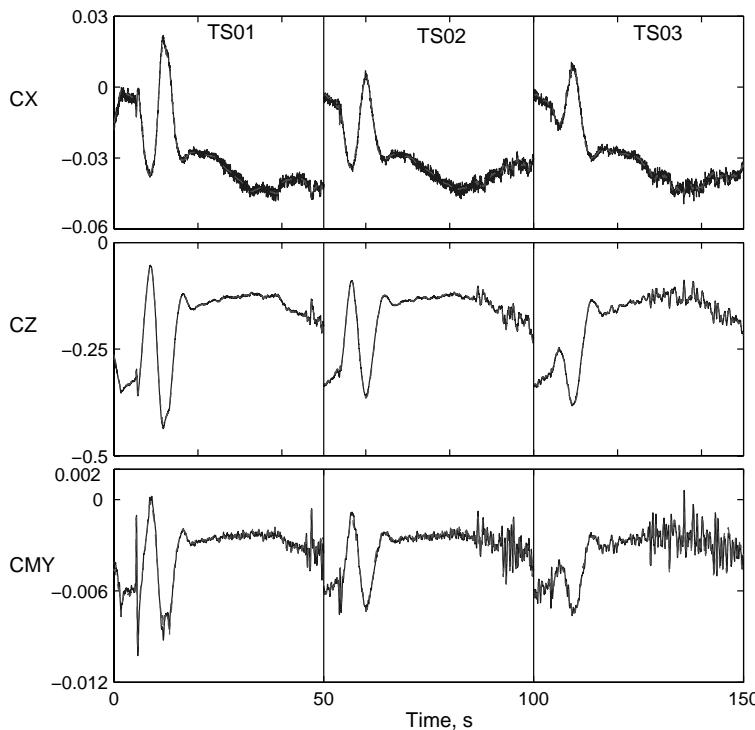


Fig. 12.56 Comparison of aerodynamic force and moment coefficients (-----, model estimated (wind-tunnel + updates); ——, flight-derived).

The input to the aerodynamic model is the measured signal for landing-gear-command, which is a discrete signal, changing from 0 to 1 for retracted and fully deployed positions. The actual deployment is modeled as a simple first-order process. The gear extension time is of the order of 1 s. Now keeping the in-air aerodynamics consisting of the pre-flight ADB plus the updates given in Eq. (12.42) fixed, the three flights with longer time segments including the landing gear are simulated. From the time history plots (not shown here) for the three flights with the updated database, but including wind-tunnel-predicted landing gear effect it was observed that for each flight the initial part of 50 s is matched nicely, but the last part in each case showed deviation in CX, CZ and CMY. The cross plots of total CX, CZ and CMY as functions of angle of attack clearly showed some mismatches during the phase with extended landing gear. As a typical example, Fig. 12.57 shows the deviation in the CX and CZ coefficients as a function of angle of attack. From Fig. 12.57 it becomes apparent that the portion of flight during which the landing gear is extended (marked as LG down) show discernible deviations, whereas for the initial 50 s segments in each flight (marked as Clean configuration) the error is randomly distributed around zero.

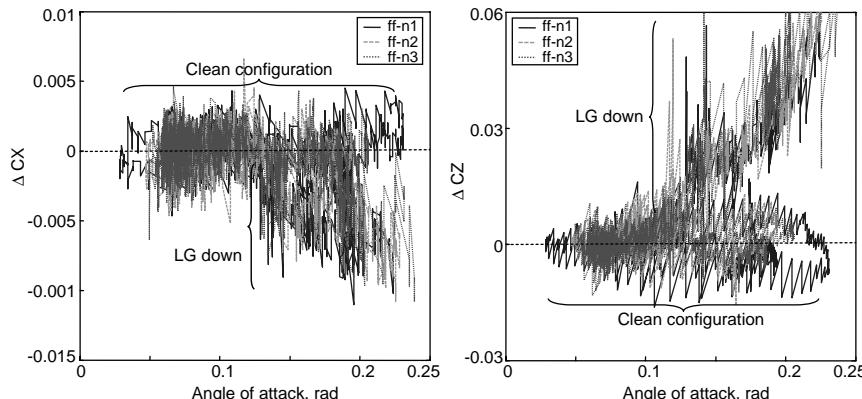


Fig. 12.57 Simulation of free flight including extended landing gear phase.

Based on this information, the correction model presented in Eq. (12.42) is now expanded to include corrections for landing gear as well.

$$\begin{aligned}\Delta CX &= CX_0 + CX_\alpha \alpha + CX_q qV/L_{ref} + CX_{\delta_{sb}} \delta_{sb} + CX_{\alpha LG} \alpha \delta_{LG} \\ \Delta CZ &= CZ_0 + CZ_\alpha \alpha + CZ_q qV/L_{ref} + CZ_{\delta_{bf}} \delta_{bf} + CZ_{\alpha LG} \alpha \delta_{LG} \\ \Delta CMY &= CM_0 + CM_\alpha \alpha + CM_{\delta_e} \delta_e + CM_{\delta_{bf}} \delta_{bf} + CM_{\alpha LG} \alpha \delta_{LG}\end{aligned}\quad (12.43)$$

where δ_{LG} is the landing gear position. The additional three parameters, namely $CX_{\alpha LG}$, $CZ_{\alpha LG}$, and $CM_{\alpha LG}$ are now estimated from the extended time segments, keeping the parameters $(CX_0, CX_\alpha, CX_q, CX_{\delta_{sb}})$, $(CZ_0, CZ_\alpha, CZ_q, CZ_{\delta_{bf}})$, and $(CM_0, CM_\alpha, CM_{\delta_e}, CM_{\delta_{bf}})$ fixed.

The correction terms based on Eq. (12.43) yielded an improved match for the force and moment coefficients (not shown here). The corresponding errors are shown in Fig. 12.58. The appropriateness of the additional correction terms for

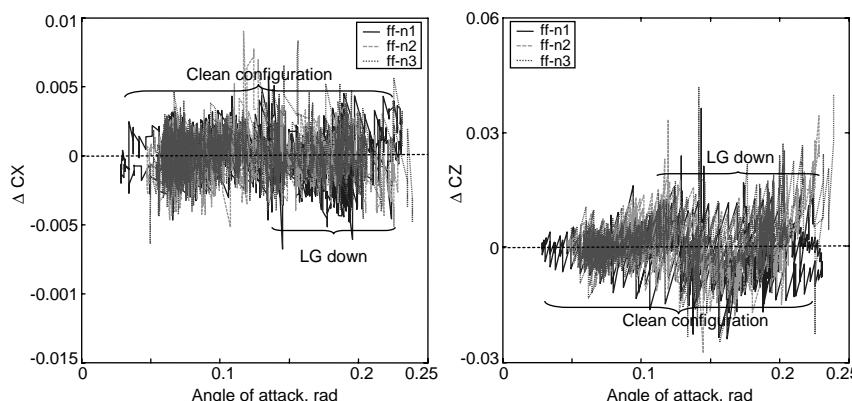


Fig. 12.58 Identification of landing gear effects.

the landing gear effect is evident from the comparison of Fig. 12.57 with Fig. 12.58. From Fig. 12.58, it is observed that the errors observed in Fig. 12.57 are mostly eliminated. There were some minor deviations to be seen in CZ and CMY at higher angles of attack with landing gear down. However, owing to a limited amount of data, any further modeling is not attempted. Nevertheless, it is an indicator that there are some nonlinear effects at angles of attack higher than 0.175 rad. From the analysis pertaining to the landing gear system, it can be concluded that the wind-tunnel measurements underpredicted the change in drag and overpredicted the change in lift due to the landing gear. The landing gear effects as modeled here are valid for the configuration as tested on the present vehicle. They result mainly from the door-flap, which was not retracted after the gear is fully down. In future applications of Phoenix, it is expected that the landing gear and gear-door mechanisms will be modified. In such a case the landing gear updates may not be appropriate after configuration changes.

To summarize, wind-tunnel calibration of the air-data-sensors was performed at three typical speeds and nonlinear sensitivity factors for the flow angles were estimated to meet the accuracy requirements. The flight recorded data from the three free flights and ground roll was analyzed by applying system identification methods. After verification of the wind-tunnel-predicted aerodynamic characteristics, updates for the forces and moments pertaining to the longitudinal motion were modeled and estimated. The rough orders of corrections required to the prediction were 3–10%. It was found that the lift generated in flight is higher than predicted and the component due to pitch rate is not adequately accounted for in the preflight aerodynamic database. The results from system identification emphasize that the basic longitudinal force coefficient for clean configuration was underestimated in the wind-tunnel, whereas the impact of the speed brakes was overestimated. An update of sideforce, rolling and yawing moment coefficients pertaining to lateral-directional motion was not possible, because there was no dynamic motion of the lateral-directional modes. The analysis of the landing gear system showed some discrepancies, which could not be directly interpreted as deviations in the wind-tunnel predictions because of the non-retracted door-flap configuration which was not tested in the wind-tunnel. The landing gear effects as modeled here are valid for the configuration tested, which may not be applicable to future vehicles.

XII. Rotorcraft Modeling and Simulation

Having dealt with several applications of fixed-wing (both jet and propeller type, as well as both stable and unstable) aircraft and a RLV demonstrator, we now turn our attention to the last case study related to rotorcraft. From an analytical modeling standpoint, rotorcraft are considered to be more complex and from a system identification viewpoint more difficult due to high aerodynamic cross coupling and rotor coupling effects, and due to poor quality of air flow measurements at low speed and hover conditions. Rotorcraft system identification is a subject in itself; an excellent review of this topic is found in Refs. 80–82. We consider here briefly three applications to demonstrate the possibilities of rotorcraft modeling and estimation by applying time domain methods.

A. Modeling Approaches

Various applications demand analytical rotorcraft models and databases of different complexities and fidelities. Three approaches, namely system identification (SID), simulation (SIM), and integrated SIM and SID, depicted in Fig. 12.59, are possible to arrive at a suitable model description for a specific application.^{21,83,84} Typical characteristics of the three approaches are indicated in the same figure.

In general, helicopter simulation models (SIM models, right column in Fig. 12.59) are generic models, incorporating a component by component description of 1) main rotor, 2) tail rotor, 3) fuselage, 4) empennage-horizontal stabilizer, and 5) fin-vertical stabilizer; they may also include other configuration-specific and other subsystems. The aerodynamic forces and moments due to each are computed using different procedures.⁸⁵ For example, the blade segmental aerodynamic forces and moments due to the main rotor are determined from a fully nonlinear transformation of inflow velocity components with respect to the blade element profile. The aerodynamic forces and moments are then determined by two- or three-dimensional profile aerodynamics; the blade profile coefficients are normally taken from tables dependent on the local inflow angle and Mach number. Both quasi-steady and unsteady effects are accounted for. The inflow angle is also a function of induced velocities, which are computed based on the momentum theory by accounting for downwash distribution and dynamic wake distortion. In contrast to the main rotor, owing to the high rotation speed, the dynamic inertia forces from the tail rotor do not influence the flight dynamic response. Therefore the tail-rotor model description used is normally

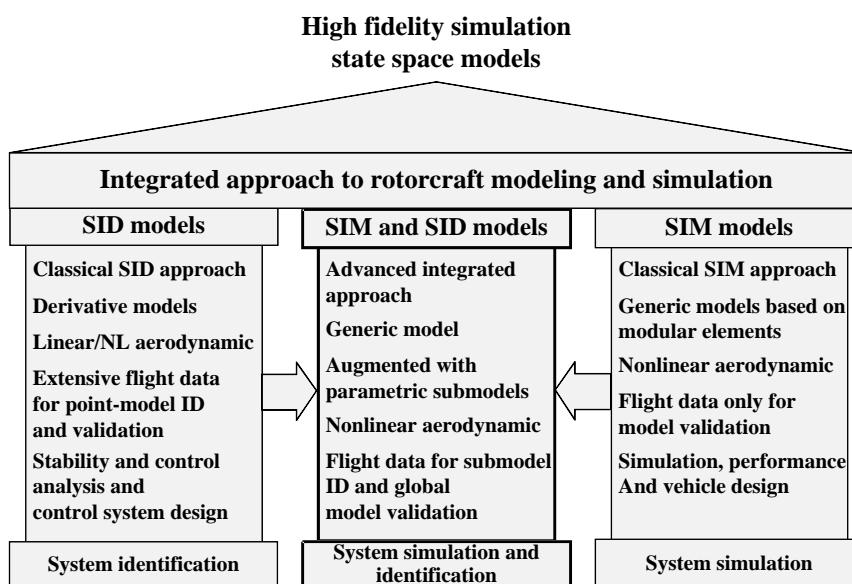


Fig. 12.59 Three-column approach to rotorcraft modeling and simulation.

a pure aerodynamic description using either the above-mentioned analytical approach or a blade element approach. The aerodynamics forces and moment due to fuselage, and horizontal and vertical stabilizer modules are normally determined using aerodynamic coefficient tables, depending on angle of attack and sideslip, and normalized for dynamic pressure. In each case the forces and moments are derived from extensive theoretical formulation and wind-tunnel tests; flight data are usually used only for model validation purposes. In general, SIM models are highly complex, but they cover the complete flight envelope.

The approach called SID, left column of Fig. 12.59, is based on the classical six degrees-of-freedom equations of motion, usually incorporating linear derivatives estimated from small excursion maneuvers. It is also possible to expand on the aerodynamic model to include nonlinearities and dependencies on flight conditions leading to model valid over larger operating range. As a typical example, we consider here identification of rigid-body aerodynamics of the EC-135 helicopter (Fig. 12.60). For this purpose a comprehensive flight test program was carried out by applying multistep and sweep control inputs at different forward speeds and in hover.⁴ We consider here the flight at 60 kts forward speed. The state, input and output vectors consisted of

$$\text{State vector } \mathbf{x} = [u \ v \ w \ p \ q \ r \ \phi \ \theta]^T$$

$$\text{Control input vector } \mathbf{u} = [\delta_{\text{lon}} \ \delta_{\text{lat}} \ \delta_{\text{ped}} \ \delta_{\text{col}}]^T$$

$$\text{Output vector } \mathbf{y} = [a_x \ a_y \ a_z \ u \ v \ w \ p \ q \ r \ \phi \ \theta \ \dot{p} \ \dot{q} \ \dot{r}]^T$$

where δ_{lon} , δ_{lat} , δ_{ped} and δ_{col} denote the longitudinal, lateral, pedal, and collective inputs, respectively; all other variables have been used and defined earlier in the book.



Fig. 12.60 EC-135 FHS, flying helicopter simulator.

Identification of aerodynamic database at a chosen flight condition is based on a multiple run analysis by applying the Gauss–Newton method in the time domain, concatenating several dynamic multistep inputs as well as frequency sweeps with longitudinal, lateral, collective, and pedal inputs. Besides the linear derivatives, the aerodynamic model incorporated angle of attack dependent parameters pertaining to the lateral directional motion and also nonlinear effects, for example the weathercock stability $C_{n\beta}$ (equivalently, the dimensional derivative N_v) is different for positive or negative angles of sideslip. As an example, we show here the equation for the specific moment \tilde{N} of the yawing motion:

$$\begin{aligned}\tilde{N} = & N_0 + N_u u + N_v v + N_w w + N_p p + N_q q + N_r r \\ & + N_{\delta\text{lon}} \delta_{\text{lon}} + N_{\delta\text{lat}} \delta_{\text{lat}} + N_{\delta\text{ped}} \delta_{\text{ped}} + N_{\delta\text{col}} \delta_{\text{col}} \\ & + N_{v\alpha} \alpha v + N_{r\alpha} \alpha r + N_{\delta\text{ped}\alpha} \alpha \delta_{\text{ped}} + N_{\delta\text{ped}\beta} \beta \delta_{\text{ped}} \\ & + N_{\delta\text{lon}\alpha} \alpha \delta_{\text{lon}} + N_{\delta\text{lon}\beta} \beta \delta_{\text{lon}}\end{aligned}\quad (12.44)$$

The first two rows in Eq. (12.44) are the linear derivatives and those in the third and fourth row their α - and β -dependencies. Equation (12.44) also includes, as already mentioned, nonlinearity in the derivative N_v for directional stability. Starting from a linear model, the additional parameters in the third and fourth rows of Eq. (12.44) were determined in a step-by-step procedure from the error analysis. They were necessary because maneuvers analyzed had larger excursions around the trim point.

The model predictive capability of the identified rigid-body model is demonstrated in Fig. 12.61 for two flight maneuvers, lateral and pedal input.⁴ As already pointed in Chapter 11, Sec. VI, for such validations only the trim conditions are matched, keeping the aerodynamic model fixed. As seen from Fig. 12.61, the predictions for complementary data, that is, flight maneuvers not used in aerodynamic identification, are fairly good. Such so-called low-frequency rigid-body six-degree-of-freedom models are generally adequate for flying qualities investigations and autopilot functional design.⁸⁴

High bandwidth models are necessary for high-fidelity simulation and for high-precision fly-by-wire/light flight control law design. They are also necessary for an in-flight simulator incorporating model-following control based on feedforward regulation providing a more accurate mode control. Extension of the six-degree-of-freedom rigid-body (SID) model through additional time delays is one possibility. Such an extension is easier to deal with and faces no problems in parameter estimation. They are, however, not suitable for model-following control, which involves inversion of the mathematical model of the host aircraft. Inversion of the model with time delays amounts to a model with time lead. For a causal system this is not realizable. From nonlinear rotor dynamics investigations, it is known that the response behavior of the rotor contributes significantly to the time delays. Hence, we have to adopt a much more complex approach based on augmenting the rigid-body model with rotor degrees of freedom to meet the high bandwidth requirements.^{84,86,87} Two approaches are possible to include the rotor degrees of freedom in the model,

SELECTED ADVANCED EXAMPLES

469

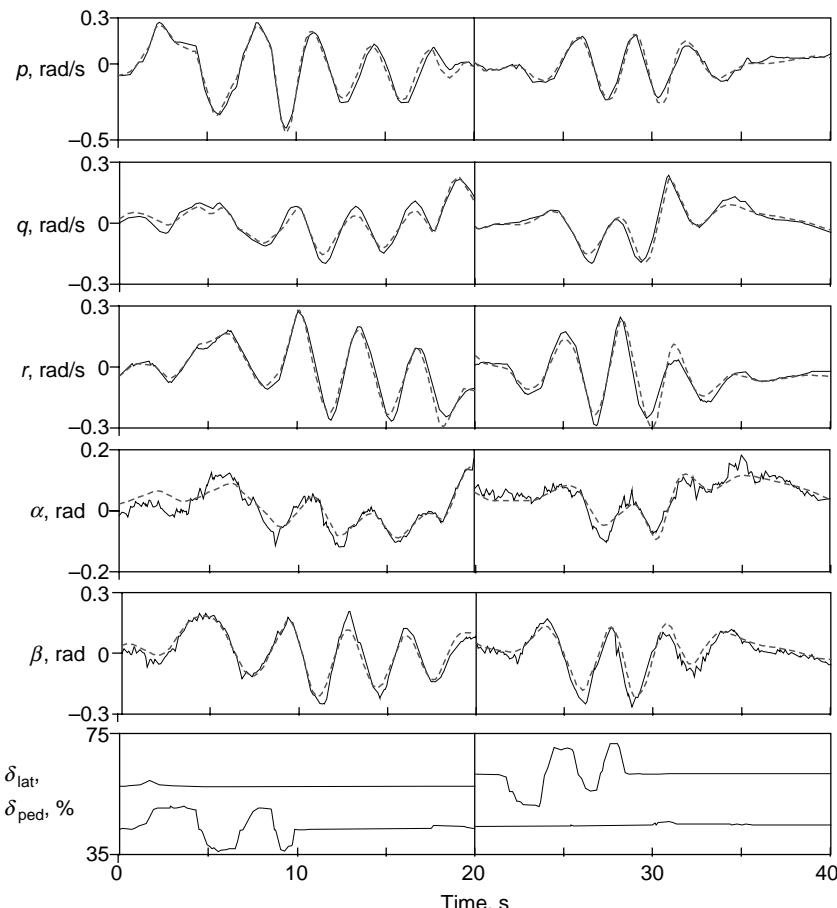


Fig. 12.61 EC-135 model predictive capability (—, flight data; -----, model estimated) (source: Ref. 4).

- 1) implicit first-order approximation of the main rotor, and 2) explicit second-order flapping equations for the main rotor.

Looking at the roll response described within a classical six-degree-of-freedom model, the basic equation for roll response can be written as:

$$\dot{p} = -L_p p + L_{\delta p} \delta_{\text{control}} \quad (12.45)$$

which indicates that a step input into the control results in a step response in the roll acceleration.

The approach of first-order approximation of the main rotor can be demonstrated by the high correlation between the flapping motions of the rotor tip path plane (lateral and longitudinal flapping) and the body-fixed rotational (roll

and pitch) accelerations.⁸⁸ For example, the correlation between the roll acceleration and lateral flapping, which mainly exists for helicopters with rigid rotors and high hinge offsets, can be expressed as

$$\ddot{p} = L_{b1}b_1 \quad (12.46)$$

and

$$\dot{b}_1 = -\frac{1}{\tau_b}b_1 - p + \frac{L_{\delta b}}{\tau_b}\delta_{\text{control}} \quad (12.47)$$

where b_1 is the lateral flapping, δ_{control} the control input at the blade root, and τ_b a flapping time constant. This coupled differential equation indicates that a step in the control input leads to a first-order response of the rotor itself coupled with the body response driven by the rotor flapping. Differentiating Eq. (12.46) and substituting Eq. (12.47) and then again making use of Eq. (12.46) leads to a second-order differential equation for the rotor/body motion:

$$\ddot{p} = L_{b1}\dot{b}_1 = -\frac{1}{\tau_b}\dot{p} - L_{b1}p + \frac{L_{b1}L_{\delta b}}{\tau_b}\delta_{\text{control}} \quad (12.48)$$

which can be equivalently reformulated as

$$(\dot{\tilde{p}}) = \tilde{L}_p\dot{p} + \tilde{L}_{pp}p + \dots + \tilde{L}_{\delta p}\delta_{\text{control}} \quad (12.49)$$

where \tilde{L}_0 denotes the new set of lateral system parameters. Thus, either the two first-order equations (12.46) and (12.47) are now used to model the roll motion and lateral flapping, or depending on the application in the control system design the equivalent second-order equation (in roll rate p), Eq. (12.49), can be used for system identification.

The derivatives appearing in Eq. (12.49) are not the same as those appearing in the classical rolling motion equation of the rigid-body motion. To obtain a relationship between the two, an analytical estimation of the new parameters is made. For this purpose, we extend the roll response of the classical six-degree-of-freedom model given by Eq. (12.45) through a short-term behavior typical of helicopters, which leads to:⁸⁶

$$\tau_b\ddot{p} = -\dot{p} - L_{pp}p + L_{\delta p}\delta_{\text{control}} \quad (12.50)$$

where the flapping time constant τ_b is estimated to be roughly one-quarter of the rotor speed. Now, dividing both sides of Eqs. (12.50) by τ_b , and comparing it with Eqs. (12.48) and (12.49), we obtain:

$$\tilde{L}_p = -\frac{1}{\tau_b}, \quad \tilde{L}_{pp} = -\frac{L_p}{\tau_b} = -L_{b1}, \quad \tilde{L}_{\delta p} = \frac{L_{\delta p}}{\tau_b} = \frac{L_{b1}L_{\delta b}}{\tau_b} \quad (12.51)$$

Thus, when we use Eq. (12.49) for parameter estimation purposes, the classical parameters are scaled through the flapping time constant τ_b and the flapping effects appear indirectly through these scaled parameters.⁸⁶

Extending the same procedure to the longitudinal motion requires two additional equations \dot{q} and (\ddot{q}) for the rotor/body motion in terms of derivatives M_{a1} and $\dot{M}_{(0)}$. Incorporation of these second-order models in the parameter estimation is little tricky, because the estimation programs require models postulated as first-order differential equations. This is elegantly done by treating \dot{p} and \dot{q} as the state variables, leading to a state vector given by

$$\text{State vector } x = [u \ v \ w \ p \ \dot{p} \ q \ \dot{q} \ r \ \phi \ \theta]^T$$

compared with the state vector $x = [u \ v \ w \ p \ q \ r \ \phi \ \theta]^T$ for the pure rigid-body motion. The control input and output vectors remain the same as before.

The main advantage of this approach is that the extended model implicitly represents dynamics of the rotor degrees of freedom as a first-order system. Together with the eighth-order state vector of the rigid-body motion defined earlier, above approximation of the rotor coupling in the pitch and roll leads to a 10th-order model, covering a wider frequency range. As rotor dynamics are implicitly modeled, the time delays are significantly reduced to the pure influence of the actuator dynamics.

The second approach, based on extending the six-degree-of-freedom model through explicit rotor degrees of freedom (SID), is illustrated in Fig. 12.62. Considering the blade flapping motions in terms of the tip path plane variables, the states of the rigid-body motion are extended by

$$x_a = [a_1 \ b_1 \ a_0 \ \dot{a}_0 \ \dot{b}_1 \ \dot{a}_1]^T$$

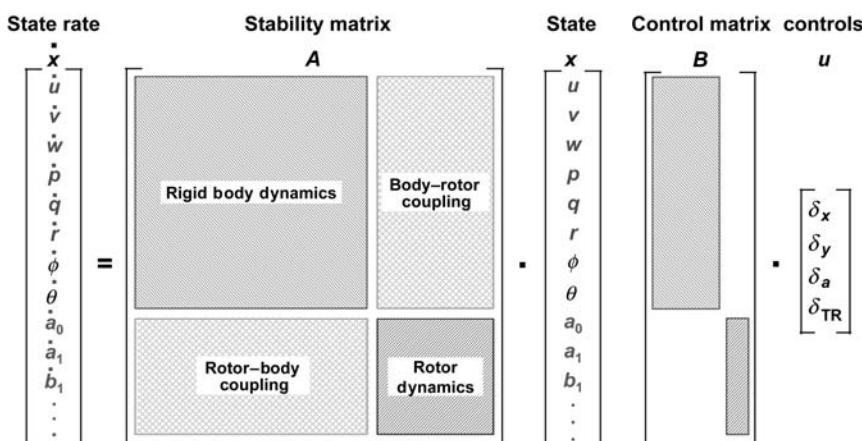


Fig. 12.62 Extended rigid-body and rotor model structure.

where a_1 and b_1 denote the longitudinal and lateral flapping, respectively, and a_0 the coning motion, each of which is modeled as a second-order system, leading to an extended model with nine degrees of freedom. An additional second-order filter function applied to the lateral control input is also included to model approximately the rotor blade lead/lag motion influence. Thus, together with the eighth-order state vector of the rigid-body motion defined earlier; it results in a 16th-order model. The SID model structure consists of two sets of equations, representing the fuselage and rotor characteristics. The individual submatrices describe rigid-body and rotor behavior and the corresponding coupling. This model structure provides a detailed representation of the rotorcraft and is closer to reality. The limitation of such an explicit model is that for parameter estimation purpose measurements of the blade motions are necessary. In the absence of such measurements the other approach based on an implicit first-order approximation is to be used, which does not require the blade measurements.

In the case of BO-105, see Fig. 12.63, the measurements of blade motions were available and hence we can apply the more detailed nine-degree-of-freedom SID model. The results of analyzing a maneuver with lateral input are summarized in Fig. 12.64. The high frequency modeling resulting from the nine-degree-of-freedom SID model can be judged best from the roll response. From the time-expanded plot of roll acceleration in Fig. 12.64b we observe that only the nine-degree-of-freedom model can match the amplitude peaks and the almost undamped high frequency of rotor lead/lag mode, although with low amplitude.

These differences in terms of the model fidelity become more evident from the frequency responses of the error function discussed in Chapter 11, Sec. VII. For the present case, Fig. 12.65 provides the magnitude and phase plots of the error function, p_m/p , that is, ratio of measured roll rate to the roll rate predicted by the model, for six- and nine-degree-of-freedom and the generic models. We recollect from Chapter 11, Sec. VII that a 0 dB magnitude and 0 deg. phase angle of the above ratio throughout the frequency range indicate a perfect model.



Fig. 12.63 Flying rotorcraft test bed BO-105 S123.

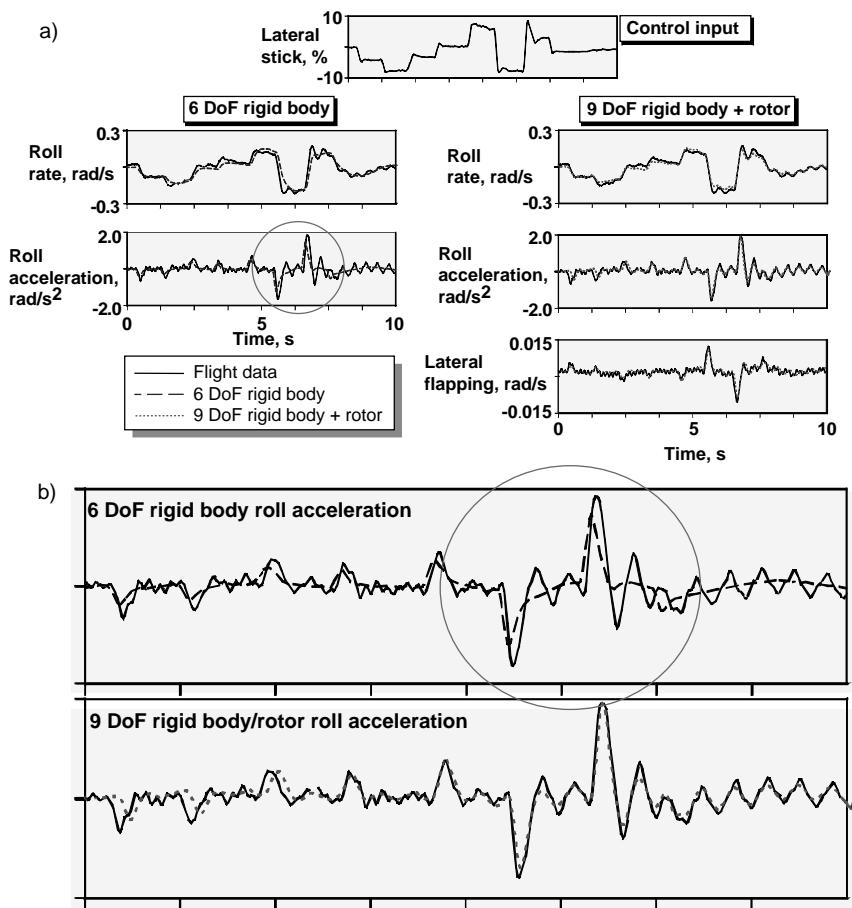


Fig. 12.64 Comparison of 6 and 9 degrees-of-freedom models. a) Time histories for different SID models, b) enlarged time scale plot.

Accordingly, we observe that the controller bandwidth based on the nine-degree-of-freedom model with rotor dynamics is close to the ideal solution. We also notice that the bandwidth of the nonlinear (NL) generic SIM model is also good, whereas that of the rigid-body model is limited in range, as determined from the suggested boundaries of the unnoticeable dynamics discussed in Chapter 11, Sec. VII.

B. Rotor Wake Modeling

During roll and pitch maneuvers in hover and at low speeds, unsymmetrical vortex compression and dilatation act on the induced velocity field in the proximity of the main rotor. These act on the effective angle of attack at the

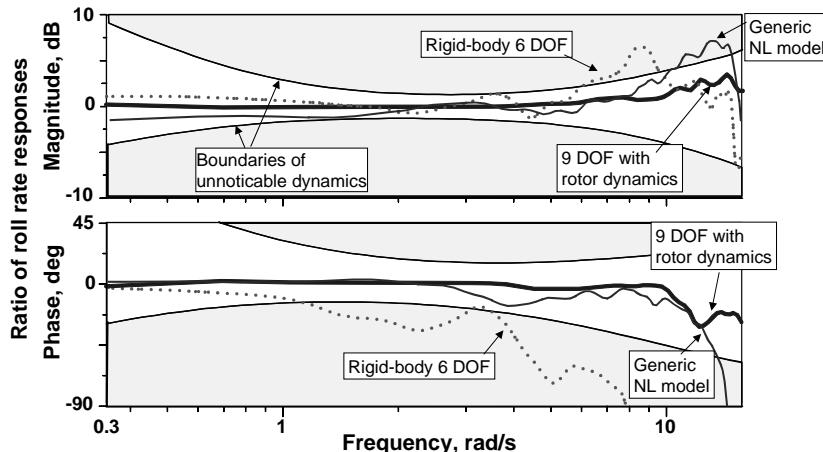


Fig. 12.65 BO-105 model validation in frequency domain (source: Ref. 78).

blade sections, which directly affect the aerodynamic rotor loads. The gyroscopic behavior of the rotor due to the blade flapping dynamics forced by these loads leads to strong cross coupling effects of the helicopter due to the wake distortion. Development of suitable flight dynamic models describing this phenomenon is a research topic in rotorcraft modeling to obtain improved dynamic fidelity in off-axis response. A combined SIM and SID approach (middle column of Fig. 12.59) allows extension of the generic nonlinear simulation models with simpler parametric terms for complex phenomenon which cannot be readily brought in the format of analytical models.⁸⁴

The helical vortex sheet and blade tip vortex propagation can be described either by geometrically prescribed or free wake formulation with discrete vortices or with an equivalent vortex ring/sheet formulation. Figure 12.66 shows the principle of a pitching helicopter in hover. The just-mentioned models require an extensive and complex formulation for the aerodynamics within the flight dynamics model. For various applications a less complex model is more appropriate. By analysis of the local phenomena with respect to their global effect on the helicopter dynamic response, it is possible to parameterize them and formulate a global model associated with the dynamics of the inflow. One possible approach is a parametric extension of the Pitt and Peters dynamic wake model which can be represented as:^{84,89–91}

$$\underbrace{\underline{M} \dot{\underline{\lambda}} + \hat{\underline{L}}^{-1} \underline{\lambda} = \underline{c}}_{\text{Pitt and Peters inflow dynamics}} + \frac{1}{\Omega} \hat{\underline{L}}^{-1} \begin{bmatrix} 0 \\ K_p(p - \dot{\beta}_s) \\ K_q(q - \dot{\beta}_c) \end{bmatrix} \quad (12.52)$$

Inflow+wake distortion dynamics

where \underline{M} is the apparent mass matrix associated with the acceleration term from momentum theory, $\hat{\underline{L}}$ the gain matrix, $\underline{\lambda} = [\lambda_0, \lambda_s, \lambda_c]^T$ the inflow ratio

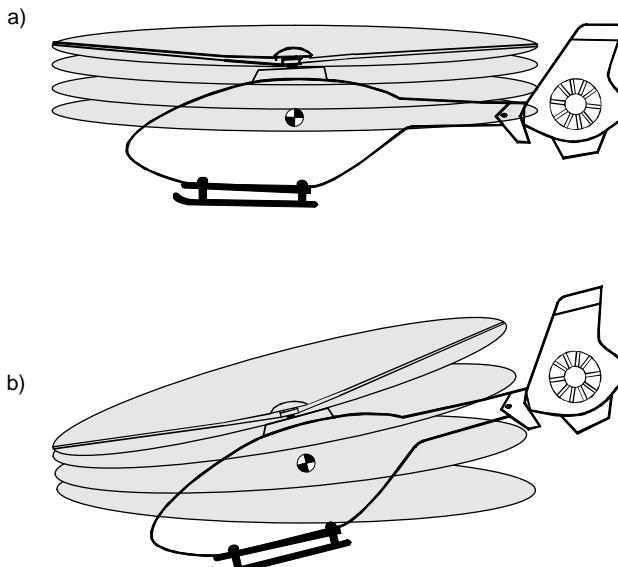


Fig. 12.66 Schematic of wake propagation. a) Pure hover, b) pitching motion in hover.

describing the first harmonic terms, $\underline{c} = [c_T, c_l, c_m]^T$ the rotor load coefficients with respect to rotor thrust and aerodynamic pitch and roll moment, and Ω the main rotor rotation speed. The part denoted as Pitt and Peters inflow dynamics describes the dynamic behavior of the three inflow components due to changes in the corresponding global aerodynamic rotor loads. K_p and K_q are the wake distortion parameters for lateral and longitudinal distribution of the induced velocity, and β_s and $\dot{\beta}_c$ the lateral and longitudinal flapping rates. The last term on the right-hand side of Eq. (12.52) is the parametric term that feeds back the total roll and pitch rates of the rotor tip path plane with respect to the surrounding air to the induced velocity distribution over the rotor disk. In the extension to the classical Pitt and Peters model, the dynamics of the inflow is now additionally driven by the pitch and roll motion of the rotor itself. The wake distortion parameters can either be estimated by linearization of the nonlinear formulation or determined by system identification techniques. Several studies based on different wake prediction theories yield values for wake distortion parameter in hover ranging from 0.5 to 1.75. At hover, neglecting the fuselage aerodynamic, the two parameters in roll and pitch axis are the same. With increasing forward speed, the wake is dragged (washed) out of the influencing zone of the main rotor, reducing the distortion effect. Numerical studies yield K_p and K_q , shown in Fig. 12.67 as a function of the advance ratio $\mu (= V_H/\Omega R)$, where V_H is the forward speed in meters per second, Ω the main rotor rotation speed in radians per second and R the rotor radius in meters.⁹²

Investigations were carried out for hover and forward flight using flight data from a BO-105-S123 helicopter; see Fig. 12.63. Upgrading the Pitt and Peters

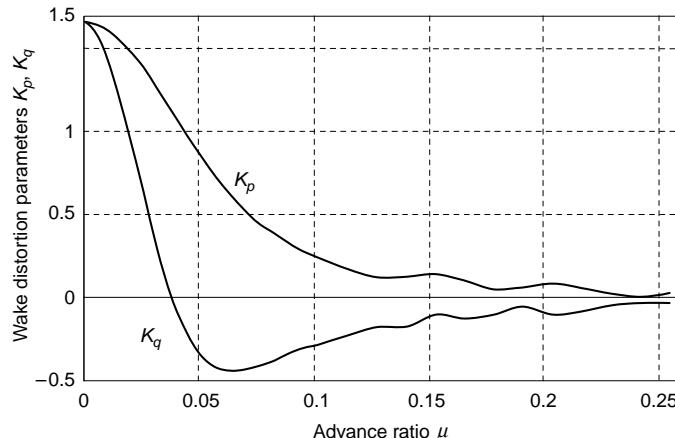


Fig. 12.67 Theoretical estimates of wake distortion parameters.

dynamic flow model to parametric wake distortion (PWD) according to Eq. (12.52) and using the nominal values of $K_p = K_q = 1.5$, simulation yields the pitch and roll rate responses to longitudinal and lateral inputs shown in Fig. 12.68. Two flight maneuvers, demarcated by the vertical lines, are shown, the first with a longitudinal and the second with a lateral pilot input. As seen from the match for the pitch rate for the second maneuver, the updated PWD model with theoretical values for the two parameters is seen to capture the basic phenomenon, although some discrepancies in matching the peak are still noticeable.

Applying the Gauss–Newton method we now estimate the two wake distortion parameters from the same set of data. No convergence problems were encountered and the two parameters were estimated without correlation and with low standard deviations, yielding the response match shown for comparison purposes in the same figure, Fig. 12.68. The match for the off-axis response is now much better. The estimates of two parameters turn out to be $K_p = 2.5$ and $K_q = 1.6$. The estimate of K_q matches well the theoretical estimate as seen in Fig. 12.67 for $\mu = 0$, whereas that of K_p is higher than the expected value. As already mentioned, the results plotted in Fig. 12.68 have been generated using PWD model with two different values for wake distortion parameters. Accounting for the wake distortion dynamics is necessary in such case, as evident from Fig. 12.69, showing on the left side the match for pitch rate by considering on the inflow dynamics only [see Eq. (12.52)]; the plot on the right-hand side is the same as that shown in Fig. 12.68 for the optimized wake distortion parameters.

In the next step, we carry out the same investigations for a forward speed of 40 m/s. In this case, based on the theory, we first neglect the PWD, because as seen from Fig. 12.67 these parameters are almost zero for the advance ratio of roughly 0.18 for this case ($V_H = 40$ m/s, and for BO-105 the main rotor radius and rotation speed are given by $R = 4.91$ m and $\Omega = 44.4$ rad/s). The results

SELECTED ADVANCED EXAMPLES

477

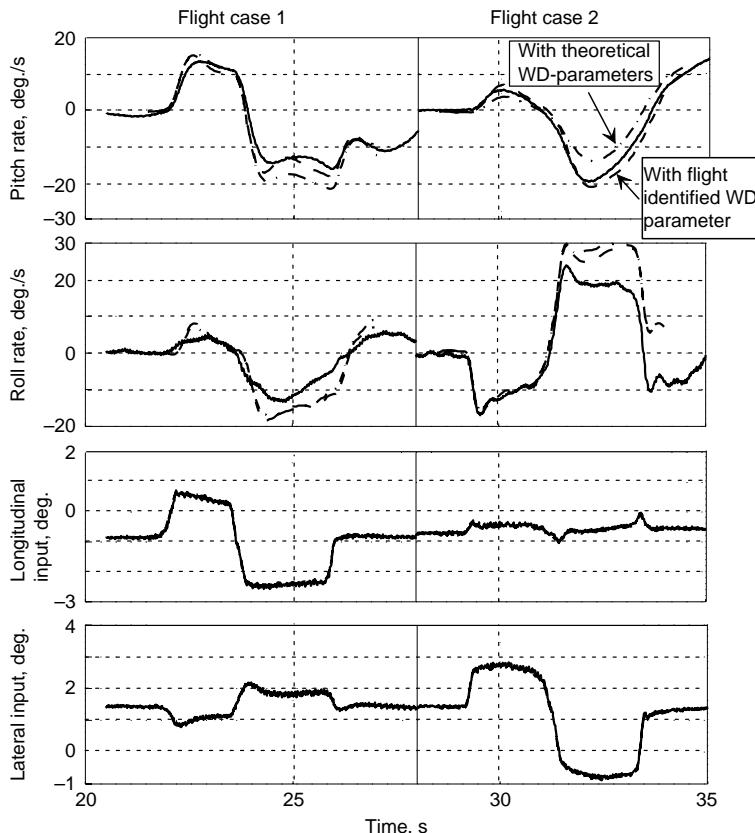


Fig. 12.68 Improvement of simulation fidelity at hover by parametric wake distortion model; (—, flight measured; -----, model estimated).

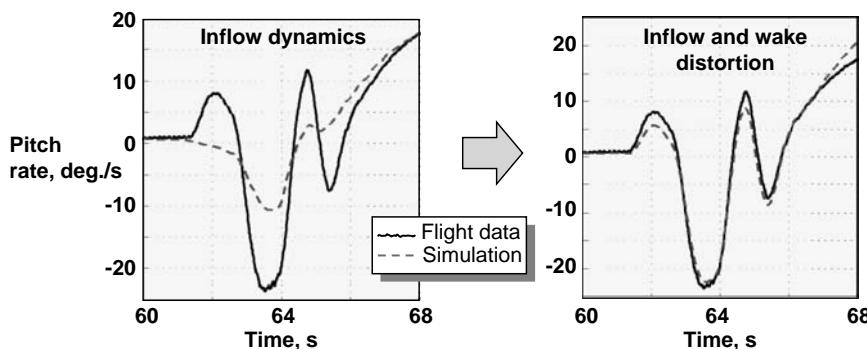


Fig. 12.69 Off-axis response at hover (—, flight measured; -----, model estimated).

of simulation without PWD and those from estimation of these two parameters are shown in Fig. 12.70 for two maneuvers, the first with a longitudinal input and the second with a lateral pilot input; once again the corresponding blade root measurements are plotted. The model response without wake distortion shows considerable discrepancies compared with the measured data, particularly in the off-axis response of pitch rate resulting from lateral input. Estimation of wake distortion parameters leads to significantly improved match for the off-axis pitch rate response due to lateral input, as seen from the second maneuver of Fig. 12.70. The two parameters turn out to be $K_p = 1.1$ and $K_q = 1.6$. These estimates do not conform to the wake prediction theory; as just pointed out, they should be roughly zero.

The anomaly in the estimates from flight data and theoretical predictions for the case of forward speed represents a classical issue often faced in modeling

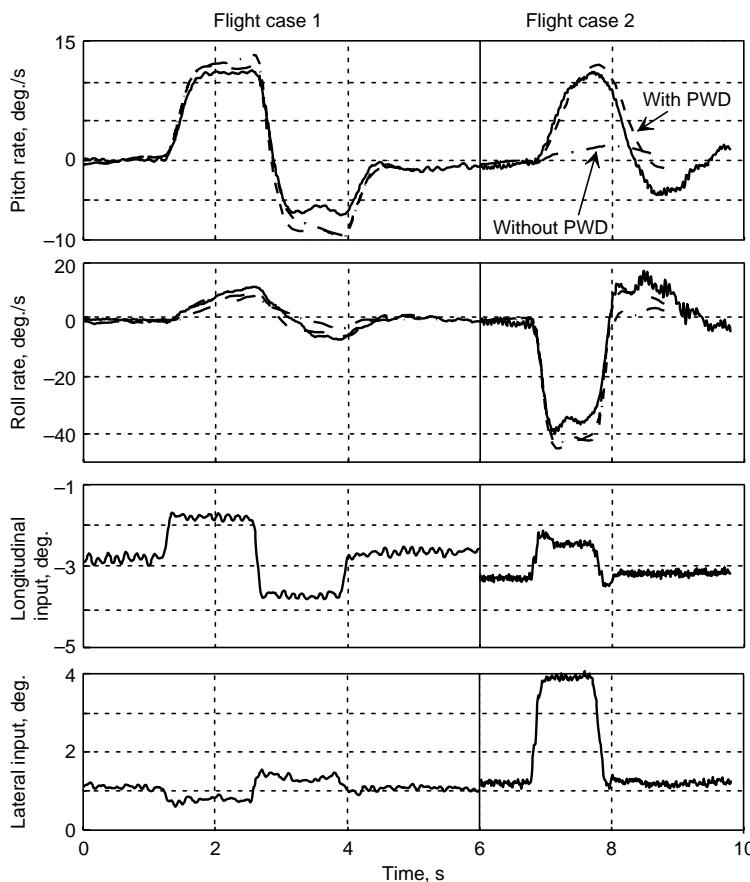


Fig. 12.70 Flight at forward speed of 40 m/s (—, flight measured; ----, model estimated).

applying system identification methods. In the present case, the estimated parameters do not represent the wake distortion phenomenon which mainly occurs at hover and at low speeds; rather they account for some other unmodeled effects, for example those resulting from rigid blade formulation, as assumed in the present investigations. The estimation method optimizes these parameters, which we could call equivalent parameters, by minimizing the error between the measured and the estimated responses. The interpretation is not part of the estimation procedure, but the task of the analyst. Further studies revealed that a model based on an elastic blade formulation significantly reduced the aforementioned discrepancies in theory and experimental results.

Although the above investigations have been carried out with a BO-105 helicopter, the modeling aspects are general and valid for other rotorcraft as well. The results shown in Figs. 12.68 and 12.70 are derived from a rotor model with a single flapping mode and shows the optimum achievable with the model complexity used. Although the basic response characteristics are fairly well captured by the above model extension, some deviations in the amplitude resulting from dynamic input are observed. Recent investigations have demonstrated that higher-order free wake models and more blade deflection modes lead to further improvements in the on- as well as off-axis response predictions.⁹³

To summarize, in this section we addressed the various approaches to rotorcraft modeling and presented extensions of rigid-body model to account for rotor degrees of freedom. Two alternative forms were presented; the exact choice depends on the availability of blade measurements. Wake distortion phenomenon, which is dominant in hover, is briefly studied, and estimation method in the time domain is applied to a parametric wake distortion model. The subtle issues related to use of such a model to flight at higher forward speed have been described.

XIII. Concluding Remarks

In this chapter we have provided an account of different types of aerodynamic modeling and parameter estimation problems, 1) downwash and other transit time lag effects, 2) aerodynamic effects of secondary order, 3) modeling of non-linear effects resulting from large control deflections, including those under abnormal conditions, 4) unsteady aerodynamics and quasi-steady stall, 5) high precision aerodynamic databases for level-D flight simulators, 6) database validation and update of an unstable aircraft, 7) wake vortex encounter, 8) reusable launch vehicle demonstrator, and 9) high bandwidth rotorcraft and parametric wake distortion modeling. In each case, results obtained from flight data analysis and applying time-domain parameter-estimation methods have been discussed. It is evident from these advanced applications that a judicious combination of conventional and specially designed flight maneuvers and analytical model formulations supported by a suitable choice of the estimation algorithm is the key to the successful generation of flight validated sub-models or complete databases. The major thrust has been to derive aerodynamic models, to explain the physics of the underlying phenomenon. In some cases, to capture the observed response characteristics, we have resorted to simplified or empirical model extensions. Much more complex model formulations including those based on computation fluid dynamics are, of course, possible, for example for the unsteady

aerodynamics or ground effect, but they are not directly amenable to system identification from flight data based on state space models, as addressed here. The models identified using the approach pursued here are suitable for several applications, including flight simulators, design of control laws, and handling qualities or other flight mechanical investigations.

A brief summary of the main outcome has been presented at the end of each subtopic. Here, we would like to repeat the conclusions drawn in Refs. 1 and 2, namely, since the quest for better understanding of aerodynamic phenomena will continue in the future, it can be said that today the scope of flight vehicle modeling will mainly be limited by the ability and skill of the analyst to interpret the modeling discrepancies, to look beyond the obvious, and to formulate mathematical models purported to underlie the physical phenomena, and little by the methods of parameter estimation. This has been demonstrated in this chapter through a number of applications of widely diversified nature, which was the main goal of this chapter. It is also substantiated by the numerous successful applications reported in the literature.

References

- ¹Jategaonkar, R. V., "Flight Vehicle System Identification—Engineering Utility," (Guest Editorial), *Journal of Aircraft*, Vol. 42, No. 1, 2005, p. 11.
- ²Hamel, P. G. and Jategaonkar, R. V., "Evolution of Flight Vehicle System Identification," *Journal of Aircraft*, Vol. 33, No. 1, 1996, pp. 9–28.
- ³Iliff, K. W., "Parameter Estimation for Flight Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 5, 1989, pp. 609–622.
- ⁴Jategaonkar, R. V., Fischenberg, D., and von Gruenhagen, W., "Aerodynamic Modeling and System Identification from Flight Data – Recent Applications at DLR," *Journal of Aircraft*, Vol. 41, No. 4, 2004, pp. 681–691.
- ⁵Wang, K. C. and Iliff, K. W., "Retrospective and Recent Examples of Aircraft Parameter Identification at NASA Dryden Flight Research Center," *Journal of Aircraft*, Vol. 41, No. 4, 2004, pp. 752–764.
- ⁶Morelli, E. A. and Klein, V., "Application of System Identification to Aircraft at NASA Langley Research Center," *Journal of Aircraft*, Vol. 42, No. 1, 2005, pp. 12–24.
- ⁷Cowley, W. L. and Glauert, H., "The Effect of the Lag of the Downwash on the Longitudinal Stability of an Airplane and on the Rotary Derivative M_q ," R.&M.718, British A. R. C., 1921.
- ⁸Etkin, B., *Dynamics of Atmospheric Flight*, John Wiley & Sons, New York, 1972.
- ⁹McCormick, B. W., *Aerodynamics, Aeronautics, and Flight Mechanics*, John Wiley & Sons, New York, 1995.
- ¹⁰Maine, R. E. and Iliff, K. W., "Maximum Likelihood Estimation of Translational Acceleration Derivatives from Flight Data," *Journal of Aircraft*, Vol. 16, No. 10, 1979, pp. 674–679.
- ¹¹Jategaonkar, R. V. and Gopalratnam, G., "Two Complementary Approaches to Estimate Downwash Lag Effects from Flight Data," *Journal of Aircraft*, Vol. 28, No. 8, 1991, pp. 540–542.
- ¹²Mönnich, W., "Ein 2-Punkt-Aerodynamikmodell für die Identifizierung," *Proceedings of the Symposium on 'Systemidentifikation in der Fahrzeugdynamik,' DFVLR-Mitt. 87–22*, Paper No. 3.1, Nov. 1987.

¹³Hanke, D., Wilhelm, K., and Meyer, H.-L., "Development and Application of In-Flight Simulator Aircraft for Flying Qualities Research at DFVLR," *Proceedings of the NAECON '86 Symposium on "Developing Technologies for Revolutionary Applications,"* Paper 117, Dayton, OH, 19–23 May, 1986.

¹⁴Jategaonkar, R. V., "Identification of the Aerodynamic Model of the DLR Research Aircraft ATTAS from Flight Test Data," DFVLR-FB 90–40, July 1990.

¹⁵Jategaonkar, R. V. "Identification of Actuation System and Aerodynamic Effects of Direct-Lift-Control Flaps," *Journal of Aircraft*, Vol. 30, No. 5, 1993, pp. 636–643.

¹⁶Krag, B., Jategaonkar, R. V., Mönnich, W., and Fischenberg, D., "Estimation of an Aerodynamic Data Base for a New C-160 Transall Flight Simulator from Flight Data," *Proceedings of the RAeS Symposium on 'Data Issues for Flight Simulators—An On-going Problem,'* London, 1993, pp. 7.1–7.12.

¹⁷Jategaonkar, R. V., Mönnich, W., Fischenberg, D., and Krag, B., "Identification of C-160 Simulator Data Base from Flight Data," *Proceedings of the 10th IFAC Symposium on System Identification and Parameter Estimation*, Copenhagen, July 1994, pp. 1031–1038.

¹⁸Jategaonkar, R. V., Mönnich, W., Fischenberg, D., and Krag, B., "Identification of Speed Brake, Air-Drop, and Landing Gear Effects from Flight Data," *Journal of Aircraft*, Vol. 34, No. 2, 1997, pp. 174–180.

¹⁹"Airplane Simulator Qualification," FAA Advisory Circular, AC 120-40C, Draft Version, July 1995.

²⁰"Joint Aviation Requirements—Aeroplane Flight Simulators," JAR-STD 1A, Westward Digital Ltd., Cheltenham, April 1997.

²¹Hamel, P. G. and Jategaonkar, R. V., "The Role of System Identification for Flight Vehicle Applications: Revisited," RTO-MP-11, Paper 2, March 1999.

²²Tobak, M., "On the Use of the Indicial Function Concept in the Analysis of Unsteady Motions of Wings and Wing-Tail Combinations," NACA Report 1188, 1954.

²³Tobak, M. and Schiff, L. B., "On the Formulation of the Aerodynamic Characteristics in Aircraft Dynamics," NASA TR R-456, 1976.

²⁴Klein, V. and Noderer, K. D., "Modeling of Aircraft Unsteady Aerodynamic Characteristics, Part 1—Postulated Models," NASA TM-109120, May 1994.

²⁵Klein, V., "Modeling of Longitudinal Unsteady Aerodynamics of a Wing-Tail Combination," NASA CR-1999-209547, Sept. 1999.

²⁶Abramov, N., Goman, M., and Khrabrov, A., "Aircraft Dynamics at High Incidence Flight with Account of Unsteady Aerodynamic Effects," AIAA Paper 2004-5274, Aug. 2004.

²⁷Leishman, J. G. and Nguyen, K. Q., "State Space Representation of Unsteady Airfoil Behavior," *AIAA Journal*, Vol. 28, No. 5, 1990, pp. 836–844.

²⁸Goman, M. and Khrabrov, A., "State-Space Representation of Aerodynamic Characteristics of an Aircraft at High Angles of Attack," *Journal of Aircraft*, Vol. 31, No. 5, 1994, pp. 1109–1115.

²⁹Fischenberg, D., "Identification of an Unsteady Aerodynamic Stall Model from Flight Test Data," AIAA Paper 95-3438, Aug. 1995.

³⁰Greenwell, D. I., "A Review of Unsteady Aerodynamic Modelling for Flight Dynamics of Manoeuvrable Aircraft," AIAA Paper 2004-5276, Aug. 2004.

³¹Fischenberg, D. and Jategaonkar, R. V., "Identification of Aircraft Stall Behavior from Flight Test Data," RTO-MP-11, March 1999, Paper No. 17.

³²Leishman, J. G. and Beddoes, T. S., "A Semi-Empirical Model for Dynamic Stall," *Journal of the American Helicopter Society*, Vol. 34, No. 3, 1989, pp. 3–17.

- ³³Jategaonkar, R. V. and Mönnich, W., "Identification of DO-328 Aerodynamic Database for a Level D Flight Simulator," AIAA Paper 97-3729, Aug. 1997.
- ³⁴Ray, P. A., "Federal Aviation Administration Flight Simulation Standards—A Regulatory Update," AIAA Paper 95-3389, Aug. 1995.
- ³⁵Skow, A. M., Titiriga, A. Jr., and Moore, W. A., "Forebody/Wing Vortex Interactions and Their Influence on Departure and Spin Resistance," CP-247, AGARD, Oct. 1978, pp. 6-1–6-26.
- ³⁶Orlik-Rückemann, K. J., "Aerodynamic Aspects of Aircraft Dynamics at High Angles of Attack," *Journal of Aircraft*, Vol. 20, No. 9, 1983, pp. 737–751.
- ³⁷Singh, J. and Jategaonkar, R. V., "Identification of Lateral-Directional Behavior in Stall from Flight Data," *Journal of Aircraft*, Vol. 33, No. 3, 1996, pp. 627–630.
- ³⁸Wieselsberger, C., "Wing Resistance Near the Ground," NACA TM-77, 1922.
- ³⁹Prandtl, L., "Der induzierte Widerstand von Mehrdeckern," Ergebnisse der Aerodynamischen Versuchsanstalt Göttingen, II. Lieferung, 1923.
- ⁴⁰Schweikhard, W., "A Method for In-Flight Measurement of Ground Effect on Fixed-Wing Aircraft," *Journal of Aircraft*, Vol. 4, No. 2, 1967, pp. 101–104.
- ⁴¹Campbell, J. P., Hassel, J. L., Jr., and Thomas, J. L., "Ground Effects on Lift for Turbofan Powered-Lift STOL Aircraft," *Journal of Aircraft*, Vol. 15, No. 2, 1978, pp. 78–84.
- ⁴²Staufenbiel, R., "Some Nonlinear Effects in Stability and Control of Wing-in-Ground Effect Vehicles," *Journal of Aircraft*, Vol. 15, No. 8, 1978, pp. 541–544.
- ⁴³Staufenbiel, R. W. and Schlichting, U.-J., "Stability of Airplane in Ground Effect," *Journal of Aircraft*, Vol. 25, No. 4, 1988, pp. 289–294.
- ⁴⁴Curry, R. E., "Dynamic Ground Effect for a Cranked Arrow Wing Airplane," NASA TM-4799, Aug. 1997.
- ⁴⁵Fischenberg, D., Mönnich, W., Krag, B., and Jategaonkar, R. V., "Aspects of C-160 Simulator Model Determination and Validation On and Close to the Ground," AIAA Paper 94-3404, Aug. 1994.
- ⁴⁶Fischenberg, D., "Ground Effect Modeling Using a Hybrid Approach of Inverse Simulation and System Identification," AIAA Paper 99-4324, Aug. 1999.
- ⁴⁷Rohlf, D., "Global Model Approach for X-31 VECTOR System Identification," *Journal of Aircraft*, Vol. 42, No. 1, 2005, pp. 54–62.
- ⁴⁸Allen, L. D., "Evolution of Flight Simulators," AIAA Paper 93-3545, Aug. 1993.
- ⁴⁹Mulder, J. A., Baarspul, M., Breeman, J. H., Nieuwpoort, A. M., Verbraak, J. P. T., and Steeman, P. S. J. M., "Determination of the Mathematical Model for the New Dutch Government Civil Aviation Flying School Flight Simulator," *Proceeding of the 18th Annual Symposium of the Society of Flight Test Engineers*, Amsterdam, Paper 15, 1987.
- ⁵⁰Baillie, S. W., Hui, K., and DeLeeuw, J., "The Flight Test and Data Analysis Program for the Development of a Boeing/deHavilland Dash 8 Simulator Model," AGARD CP-519, Paper 30, Oct. 1992.
- ⁵¹Neville, K. W. and Stephens, A. T., "Flight Update of Aerodynamic Math Model," AIAA Paper 93-3596, Aug. 1993.
- ⁵²Trankle, T. L. and Bachner, S. D., "Identification of a Nonlinear Aerodynamic Model of the F-14 Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 6, 1995, pp. 1292–1297.
- ⁵³Rohlf, D., "Direct Update of a Global Simulation Model with Increments via System Identification," RTO-MP-11, March 1999, Paper 28.
- ⁵⁴Weiss, S., Gockel, W., Mönnich, W., and Rohlf, D., "Identification of Dornier-328 Reversible Flight Control Systems," AIAA Paper 98-4163, Aug. 1998.

- ⁵⁵Shikany, D. A., "Math Pilots in Validation Data Packages—Appropriate Use or Potential Abuse?" *Proceedings of the RAeS Symposium on "Data for Simulation—The Weakest Link?,"* London, Paper 12, 6–7 Nov. 2002.
- ⁵⁶Hall, F. C., "Flight Testing for Simulator Data," AIAA Paper 91-2930, Aug. 1991.
- ⁵⁷Mönnich, W. and Jategaonkar, R. V., "Data Base Development for Level D Simulators—Lessons Learned," RTO-MP, Paper 14, March 1999.
- ⁵⁸Ross, H., "X-31 Enhancement of Aerodynamics for Maneuvering Beyond Stall," CP-497, AGARD, Paper 2, Nov. 1991.
- ⁵⁹Weiss, S., Friehmelt, H., Plaetschke, E., and Rohlf, D., "X-31A System Identification Using Single-Surface Excitation at High Angles of Attack," *Journal of Aircraft*, Vol. 33, No. 3, 1996, pp. 485–490.
- ⁶⁰Smith, R. E., "Flying Qualities Development for the X-29A Demonstrator," International symposium on "In-Flight Simulation for the 90's," DLR German Aerospace Research Establishment, Braunschweig, Germany, 1–3 July 1991.
- ⁶¹Iliiff, K. W., "X-29A Lateral-Directional Stability and Control Derivatives Extracted From High-Angle-of-Attack Flight Data," NASA TP-3664, Dec. 1996.
- ⁶²Iliiff, K. W. and Wang, K. C., "Flight-Determined Subsonic Longitudinal Stability and Control Derivatives of the F-18 High Angle of Attack Research Vehicle (HARV) With Thrust Vectoring," NASA/TP-97-206539, Dec. 1997.
- ⁶³Iliiff, K. W. and Wang, K. C., "Flight-Determined, Subsonic, Lateral-Directional Stability and Control Derivatives of the Thrust-Vectoring F-18 High Angle of Attack Research Vehicle (HARV), and Comparison to the Basic F-18 and Predicted Derivatives," NASA/TP-1999-206573, Jan. 1999.
- ⁶⁴Vicroy, D., Brandon, J., Greene, G., Rivers, R., and Stewart, E., "Characterizing the Hazard of a Wake Vortex Encounter," AIAA Paper 97-0055, Jan. 1997.
- ⁶⁵Pete, K. R., Smith, S. T., and Vicroy, D., "Model Validation of Wake-Vortex/Aircraft Encounters," AIAA Paper 2000-3979, Aug. 2000.
- ⁶⁶Fischenberg, D., "S-WAKE: Results of Flight Test Data Analysis," DLR-IB 111-2002/11, Sept. 2002.
- ⁶⁷Burnham, D. C. and Hallock, J. N., "Chicago Monoacoustic Vortex Sensing System, Volume IV: Wake Vortex Decay," DOT/FAA/RD-79-103 IV, Springfield, VA, Vol. 4, 1982.
- ⁶⁸Lamb, H., *Hydrodynamics*, Cambridge University Press, Cambridge, MA, 1932, pp. 590–592.
- ⁶⁹Fischenberg, D., "Bestimmung der Wirbelschleppencharakteristik aus Flugmessdaten," *Proceedings of the DGLR Annual Meeting*, Stuttgart, Paper DGLR-2002-170, 23–26 Sept. 2002 (in German).
- ⁷⁰Spies, J., "HOPPER—ein ASTRA Systemkonzept," *Proceedings of the DGLR Annual Meeting*, Leipzig, Paper No. DGLR-JT2000-070, 18–21 Sept. 2000 (in German).
- ⁷¹Brücker, E., Dittmann, R., and Noack, E., "ASTRA—das deutsche Technologieprojekt für zukünftige Raumtransportsysteme," *Proceedings of the DGLR Annual Meeting*, Leipzig Paper No. DGLR-JT2000-117, 18–21 Sept. 2000 (in German).
- ⁷²Gockel, W., Kyr, P., Janovsky, R., and Roenneke, A., "Reusable RLV Demonstrator Vehicles - Phoenix Flight Test Results and Perspectives," *Proceedings of the 55th International Astronautical Congress*, Vancouver, Canada, 4–8 Oct. 2004.
- ⁷³Jategaonkar, R. V., Behr, R., Gockel, W., and Zorn, C., "Data Analysis of Phoenix RLV Demonstrator Flight Test," AIAA Paper 2005-6129, Aug. 2005.
- ⁷⁴Janicik, J. L., "The Phase I Space Maneuver Vehicle Test Program—Leading the United States into 21st Century Space Test and Evaluation," AIAA Paper 1999-4539, Sept. 1999.

- ⁷⁵Yanagihara, M., Shigemi, M., and Suito T., "Estimating Aerodynamic Characteristics of Automatic Landing Flight Experiment Vehicle Using Flight Data," *Journal of Aircraft*, Vol. 36, No. 6, 1999, pp. 926–933.
- ⁷⁶Behr, R., "Phoenix: Aerodynamic Data Base, Version 3.1," EADS-ST Report PHX-DP-01, Issue 1, Jan. 2004.
- ⁷⁷Parameswaran, V., Jategaonkar, R. V., and Press, M., "Five-Hole Flow Angle Probe Calibration from Dynamic and Tower Flyby Maneuvers," *Journal of Aircraft*, Vol. 42, No. 1, 2005, pp. 80–86.
- ⁷⁸Gockel, W., "Phoenix Definition and Justification File—Plant Model," EADS-ST Report PHX-DDD-008, Issue 3, Jan. 2004.
- ⁷⁹Cobleigh, B. R., "Development of the X-33 Aerodynamic Uncertainty Model," NASA/TP-1998-206544, April 1998.
- ⁸⁰Hamel, P. G. (ed.), "Rotorcraft System Identification," AGARD AR-280, Sept. 1991.
- ⁸¹Hamel, P. G. and Kaletka, J., "Rotorcraft System Identification—An Overview of AGARD FVP Working Group 18," AGARD CP-552, Paper 18, 1995.
- ⁸²Tischler, M. B., "System Identification Methods for Aircraft Flight Control development and Validation," RTO MP-11, Paper 3, March 1999.
- ⁸³Rohlf, M., von Grünhagen, W., and Kaletka, J., "Nonlinear Rotorcraft Modeling and Identification," RTO-MP-11, Paper 23, March 1999.
- ⁸⁴Hamel, P. G., "Advances in Aerodynamic Modeling for Flight Simulation and Control Design," International Association of Applied Mathematics and Mechanics: GAMM-Mitteilungen, Wiley-VCH, Berlin, Vol. 23, No. 1–2, 2000, pp. 7–50.
- ⁸⁵Bramwell, A. R. S., Done, G., and Balmford, D., *Bramwell's Helicopter Dynamics*, AIAA, Reston, VA, 2001.
- ⁸⁶Kaletka, J. and von Grünhagen, W., "System Identification of Mathematical Models for the Design of a Model Following Control System," *Vertica*, Vol. 13, No. 2, 1989, pp. 213–228.
- ⁸⁷Tischler, M. B., "Identification Requirements for High-Bandwidth Rotorcraft Flight Control System Design," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 5, 1990, pp. 835–841.
- ⁸⁸von Grünhagen, W., Bouwer, G., Pausder, H.-J., Henschel, F., and Kaletka, J., "A High Bandwidth Control System for Helicopter In-Flight Simulator ATTHeS—Modeling, Performance and Applications," in *Advances in Aircraft Flight Control*, edited by M. B. Tischler, Taylor & Francis, London, 1996, pp. 73–101.
- ⁸⁹Pitt, D. M. and Peters, D. A., "Theoretical Prediction of Dynamic Inflow Derivatives," *Vertica*, Vol. 5, No. 1, 1981, pp. 21–34.
- ⁹⁰Hamers, M. and von Grünhagen, W. "Nonlinear Helicopter Model Validation Applied to Real Time Simulations," *Proceedings of the 53rd Annual AHS Forum*, Vol. 2, American Helicopter Society, Alexandria, VA, 1997, pp. 958–972.
- ⁹¹Kraemer, P., Gimonet, B., and von Grünhagen, W., "A Systematic Approach to Non-linear Rotorcraft Model Identification," *Journal of Aerospace Science and Technology*, Vol. 6, No. 8, 2002, pp. 579–590.
- ⁹²Basset, P.-M. and Tchen-Fo, F., "Study of the Rotor Wake Distortion Effects on the Helicopter Pitch-Roll Cross-Couplings," *Proceedings of the 24th European Rotorcraft Forum*, Marseille, Sept. 1998.
- ⁹³Theodore, C. and Celi, R., "Helicopter Flight Dynamic Simulation with Refined Aerodynamics and Flexible Blade Modeling," *Journal of Aircraft*, Vol. 39, No. 4, 2002, pp. 577–586.

“Science is not formal logic—it needs the free play of the mind
in as great a degree as any other creative art. It is true that
this is a gift which can hardly be taught. But its growth
can be encouraged in those who already possess it.”¹

Max Born (1882, 1970),
German physicist, Nobel Laureate

Epilogue

THE LEADING idea of this book has been to provide an overview of the system identification methodology as applied to flight vehicles, focusing on the key methods of parameter estimation in the time domain and emphasizing nonlinear modeling. Although a certain degree of mathematical treatment is unavoidable, we have attempted to keep the theoretical development simple and understandable, including that for the methods in the stochastic framework. The major thrust has been on practical issues, and on explaining in simple terms the need, purpose, background theory, and scope of the various topics covered in this book. The book is not intended as a cookbook, but certain guidelines demonstrated and recommended herein help the reader to arrive at an adequate model. In a few cases the discussion has been on a philosophical level, but this was intentional, with the goal of sharing the process learnt and experience gathered during the exercises.

It is our strong belief that system identification should be a goal-oriented task. Hence, at the outset we need to define the specific purpose for which the model is required, because this determines to a large extent the amount of effort that will be spent on model building. A better understanding of the coordinated Quad-M approach helps one to arrive at an adequate model with a minimum amount of effort. We recapture here the principle outcomes of our discussions:

1) It has been demonstrated that, without getting bogged down in theoretical details, through simple basic understanding and the existing a priori knowledge of the modes of aircraft motion, adequate inputs can be designed in most cases. Multistep inputs are necessary to excite the dynamic motion; however, their exact shape and time steps are not critical; empirical rules are quite adequate. Besides the classical dynamic maneuvers, other system identification maneuvers, for example acceleration–deceleration, pushdown–pullups, steady heading steady sideslips, and so on, are useful in system identification.

2) Data compatibility check helps to verify and improve the quality of the recorded data. In most cases, the standard output error method accounting only for measurement noise is adequate for this purpose. The data gathering, consisting of performing maneuvers, and recording, checking and improving data, mainly limits the scope of the aerodynamic modeling.

3) The output error method is most widely used to estimate the aerodynamic parameters. Extensions and practical issues discussed here are helpful in catering for specific problems one may encounter in practice. It can be applied to large-scale problems without difficulties. If problems are encountered, in most of the cases they are due to some other cause, like poor information content in the data being analyzed or a wrong model postulate. The more advanced filter error method accounting for state noise is required in only a few cases, for example for data with atmospheric turbulence or stochastic vortex shedding from aircraft nose at high angles of attack. The least squares method provides an alternative approach, but it needs high-precision low-noise measurements, which are free of systematic bias and scale factor errors. Furthermore, state space representation and output error method incorporating simulation is more efficient and closer to many other applications of the model. In the specific case of highly unstable aircraft, however, techniques other than the output error method are preferable, such as filter error, least squares, or extended Kalman filter.

4) Basically, there are no restrictions on postulating a model to be identified. Highly complex and nonlinear models can be identified from data, provided adequate information is available in the flight data.

In short, we have a fairly good understanding of the basic theory and parameter estimation algorithms; the challenge is in the applications. There is no doubt that in the majority of the cases flight vehicle system identification can be performed accurately and reliably. We have covered numerous examples, providing an appreciation of a wide variety of aerodynamic modeling and parameter estimation problems. The choice of the examples presented may appear biased, but it is unavoidable, because we wish to report only on those cases with which we have worked. As a consequence, the views represented in this book mostly reflect the philosophy practiced by the authors and at their organization. Others may not totally agree with some of the issues discussed, but the reported schemes have worked in practice, providing acceptable solutions in some of the most demanding tasks. In this sense it seems to be a reasonable philosophy. In some of the more advanced and emerging applications, the ability to formulate mathematical models purported to underlie the physical phenomenon and the skill to interpret the modeling discrepancies, to look beyond the obvious will mainly limit the scope of modeling. In other words, experience and engineering judgment will be called for, and in this sense system identification remains a scientific art, but an art supported by well-established methodology.

We would venture a guess about the likely direction of future system identification activities, although forecasting is rather risky.² Three classical topics, 1) generation of flight validated high fidelity databases for simulators and other applications, 2) modeling of nonlinear aerodynamics, and 3) modeling of unstable aircraft, will continue to be the prime areas of applications. Furthermore, better updating techniques will be needed to accommodate the slope type, that is, derivative, estimates on nonslope databases in look-up table form for force and moment coefficients. The use of distributed mass models will pave the way for estimation and validation of full flexible aircraft models combining flight mechanics and structural models. Real-time parameter estimation is re-emerging as yet

another area of current and future research. Some theoretical aspects will focus on estimation of modeling uncertainties. Other advances will result from the actual needs, for example better air data measurements. In the distant future, another thrust will possibly be on integrating system identification and computational fluid dynamics methodologies. These may call for new modeling concepts, and associated measurement and experimental techniques to enable parameter estimation. In short, we will face some most demanding and fascinating modeling problems in the future and we are convinced that in these cases system identification will provide answers to unmasking the modeling deficiencies, reducing developmental risk and improving flight safety issues.

References

¹Born, M., "The Teaching of Theoretical Physics in Universities," *Reports on Progress in Physics*, Vol. 8, 1941, pp. 1–10.

²Jategaonkar, R.V., "Flight Vehicle System Identification—Engineering Utility," (Guest Editorial), *Journal of Aircraft*, Vol. 42, No. 1, 2005, p. 11.

This page intentionally left blank

Appendix A**Power Spectrum of a Multistep Input Signal**

AMULTISTEP input signal of arbitrary shape can be synthesized by a sequential combination of equidistant pulse inputs (see Fig. A.1a), and each pulse of a fixed duration as a difference of two time-displaced step signals; see Fig. A.1b. Accordingly, a multistep input signal $u(t)$ in time domain can be expressed as¹⁻³

$$u(t) = \sum_{k=1}^{n_p} V_k \{ \sigma[t - (k - 1)\Delta t] - \sigma(t - k\Delta t) \} \quad (\text{A.1})$$

where n_p is the number of pulses (for example, seven for the case depicted in Fig. A.1a), V_k the amplitudes of the individual pulses, $\sigma(t)$ the unit step at time t , and Δt the time step (width) of the each pulse.

The frequency domain transformation of Eq. (A.1) is given by

$$u(\omega) = \sum_{k=1}^{n_p} V_k \left[\frac{1}{j\omega} e^{-j(k-1)\omega\Delta t} - \frac{1}{j\omega} e^{-jk\omega\Delta t} \right] \quad (\text{A.2})$$

where ω is the angular frequency, e the base of the natural logarithm (a constant), and j the imaginary number. Equation (A.2) can be rewritten as

$$u(\omega) = \Delta t \frac{1 - e^{-j\omega\Delta t}}{j\omega\Delta t} \sum_{k=1}^{n_p} V_k e^{-j(k-1)\omega\Delta t} \quad (\text{A.3})$$

In terms of the normalized frequency $\Omega = \omega\Delta t$, Eq. (A.3) leads to

$$u(\omega) = \Delta t \frac{e^{j\Omega/2} - e^{-j\Omega/2}}{j\Omega} e^{-j\Omega/2} \sum_{k=1}^{n_p} V_k e^{-j(k-1)\Omega} \quad (\text{A.4})$$

Using the Euler's formula for the complex exponential function in terms of trigonometric functions, $e^{jx} = \cos x + j \sin x$, where x is a real number, it turns out

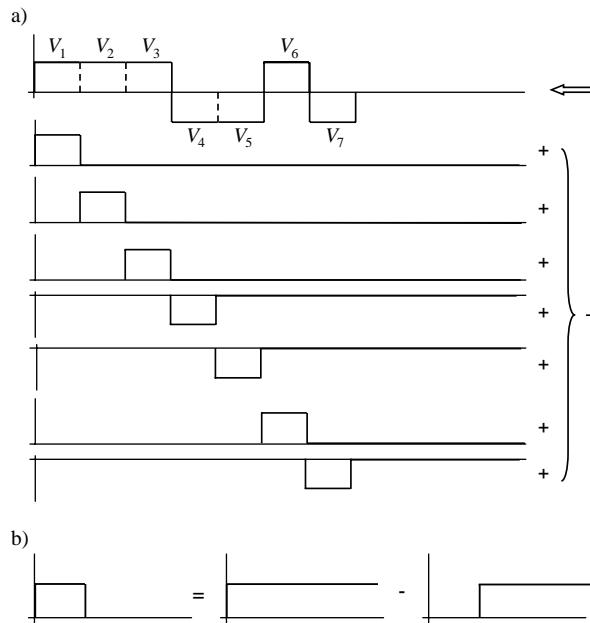


Fig. A.1 Schematic decomposition of multistep input. a) Multistep input signal, b) pulse input.

that $e^{j\Omega/2} - e^{-j\Omega/2} = 2j \sin(\Omega/2)$, which when substituted in Eq. (A.4), after minor simplification yields

$$u(\omega) = \Delta t \frac{\sin(\Omega/2)}{(\Omega/2)} \sum_{k=1}^{n_p} V_k e^{-j(2k-1)\Omega/2} \quad (\text{A.5})$$

Now, applying the same Euler's formula once again, we can expand the exponential term within the summation on the right-hand side of Eq. (A.5), which leads to

$$u(\omega) = \Delta t \frac{\sin(\Omega/2)}{(\Omega/2)} \sum_{k=1}^{n_p} V_k \left[\cos \frac{(2k-1)\Omega}{2} + j \sin \frac{(2k-1)\Omega}{2} \right] \quad (\text{A.6})$$

The power (or energy) spectrum, $E(\omega)$, of an arbitrary multistep signal represented in the time domain by Eq. (A.1) can be computed easily using its

frequency domain representation of Eq. (A.6) as follows:

$$\begin{aligned} E(\omega) &= |u(\omega)|^2 = \text{Re}^2[u(\omega)] + \text{Im}^2[u(\omega)] \\ &= \Delta t^2 \left[\frac{\sin(\Omega/2)}{(\Omega/2)} \right]^2 \left\{ \left[\sum_{k=1}^{n_p} V_k \cos\left(\frac{2k-1}{2}\Omega\right) \right]^2 \right. \\ &\quad \left. + \left[\sum_{k=1}^{n_p} V_k \sin\left(\frac{2k-1}{2}\Omega\right) \right]^2 \right\} \end{aligned} \quad (\text{A.7})$$

Now, applying the trigonometric identity, known as the Carnot formula, $\sin^2 u = (1 - \cos 2u)/2$, to the term $\sin^2(\Omega/2)$, and expanding on the terms of square of the sums within flower bracket, after some minor simplification, Eq. (A.7) leads to the desired form:

$$E(\omega) = 2\Delta t^2 \frac{1 - \cos \Omega}{\Omega^2} \left[\sum_{k=1}^{n_p} V_k^2 + 2 \sum_{i=1}^{n_p-1} \cos i\Omega \sum_{l=1}^{n_p-i} V_l V_{l+i} \right] \quad (\text{A.8})$$

References

- ¹Koehler, R. and Wilhelm, K., "Auslegung von Eingangssignalen für die Kennwertermittlung," DFVLR-IB 154-77/40, Dec. 1977 (in German).
- ²Koehler, R., Personal correspondence (nondocumented), DLR Institute of Flight Mechanics, Braunschweig, 1977.
- ³Proskawetz, K.-O., "Optimierung stufenförmiger Eingangssignale im Frequenzbereich für die Parameter Identifizierung," ZFW, Vol. 9, No. 6, Nov.–Dec. 1985, pp. 362–370 (in German).

This page intentionally left blank

Appendix B

Identifiability of Initial Conditions and Bias Parameters

GIVEN A linear model postulate of the form

$$\dot{x} = Ax + B[u - \Delta u], \quad x(t_0) = x_0 \quad (\text{B.1})$$

$$y = Cx + D[u - \Delta u] + \Delta z \quad (\text{B.2})$$

we would like to investigate, in general, the identifiability of the parameters appearing in matrices A , B , C , and D , the initial conditions x_0 , and the zero shifts Δu and Δz . However, as discussed in Chapter 3, Sec. V.A, proper flight maneuvers ensure identifiability of aerodynamic derivatives, that is, of A , B , C , and D . Hence, we consider here the issue related to other constant terms x_0 , Δu and Δz only. They are in total $(n_x + n_u + n_y)$ in number, where n_x is the number of states, n_u the number of control inputs, and n_y the number of outputs. The state, control, and output variables are functions of time; however, we drop the time dependence, (t) , in the notation throughout this section.

Identifiability of any parameter in general is determined by the observability of the system. A system is said to be observable if the initial state x_0 can be determined uniquely by examining the system output $y(t)$ for $t > t_0$ in a finite interval over some period of time $0-T$. If the initial conditions are determined, the system behavior at any time t in the future is uniquely determined. To investigate identifiability, we apply the conventional observability analysis. We follow here the commonly applied Kalman criterion.¹ Drawing upon the discussion presented in Ref. 2, the identifiability investigations for a system represented in Eqs. (B.1) and (B.2) are performed in two steps; the first deals with initial conditions only and the second with zero shifts and initial conditions together.

Accordingly, we start by considering the following system without bias terms:

$$\dot{x} = Ax; \quad x(t_0) = x_0 \quad (\text{B.3})$$

$$y = Cx \quad (\text{B.4})$$

where the control input u has also been dropped without loss of generality, because it does not affect these investigations. The general procedure of checking observability, leading to the Kalman criterion, consists of deriving a system equations in terms of A and C , which can be solved for x . This is possible by starting from the observation equation, Eq. (B.4), and successively differentiating and substituting the right hand side of Eq. (B.3) for \dot{x} . This leads to

$$\begin{aligned} y &= Cx \\ \dot{y} &= C\dot{x} = CAx \\ \ddot{y} &= CA\dot{x} = CA^2x \\ &\dots \end{aligned} \tag{B.5}$$

Now, y is known from the measurements, hence its derivatives \dot{y}, \ddot{y}, \dots are also known. We can then rewrite Eq. (B.5) in a familiar matrix form $Ax = b$ as

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} x = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ \vdots \end{bmatrix} \tag{B.6}$$

Equation (B.6) can be solved for x uniquely provided the matrix on the left-hand side

$$Q = [C^T \ (CA)^T \ (CA^2)^T \ \dots \ (CA^{n-1})^T]^T \tag{B.7}$$

has rank n_x . The matrix Q defined in Eq. (B.7) is commonly called the observability matrix. A unique solution for x , ensured by rank n_x of observability matrix, implies that we can estimate the initial conditions x_0 completely.

Having investigated the necessary conditions for identifiability of x_0 , in the second step we get back to the complete system model including initial conditions and zero shifts. By multiplying out the terms in Eqs. (B.1) and (B.2), and, as discussed earlier, dropping out the terms involving control inputs u , we get

$$\dot{x} = Ax - B\Delta u; \quad x(t_0) = x_0 \tag{B.8}$$

$$y = Cx - D\Delta u + \Delta z \tag{B.9}$$

We once again follow a similar procedure of successive differentiation and substitution. Starting from Eq. (B.9), differentiating the same, substituting Eq. (B.8) for \dot{x} and using the knowledge that Δu and Δz are zero because the bias parameters are assumed to constant over the period of observation, we

obtain

$$\begin{aligned}
 y &= Cx - D\Delta u + \Delta z \\
 \dot{y} &= C\dot{x} - 0 + 0 \\
 &= CAx - CB\Delta u \\
 \ddot{y} &= CA\dot{x} - 0 \\
 &= CA^2x - CAB\Delta u \\
 &\dots
 \end{aligned} \tag{B.10}$$

which can be written in matrix form as:

$$\underbrace{\begin{bmatrix} C & -D & I \\ CA & -CB & 0 \\ CA^2 & -CAB & 0 \\ \vdots & \vdots & \vdots \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} x \\ \Delta u \\ \Delta z \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ \vdots \end{bmatrix} \tag{B.11}$$

where I is the $n_y \times n_y$ identity matrix. To allow estimation of all the initial conditions and bias parameters, the system matrix \mathcal{A} in Eq. (B.11) must have rank $(n_x + n_u + n_y)$.

Owing to the presence of the identity matrix, the first row of \mathcal{A} yields rank n_y . The rest of the matrix consists of

$$\begin{bmatrix} CA & -CB \\ CA^2 & -CAB \\ CA^3 & -CA^2B \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} [A \quad -B] \tag{B.12}$$

It now remains to find the rank of the matrix product on the right-hand side of Eq. (B.12). The first term on the right-hand side of Eq. (B.12) is the observability matrix Q appearing in Eq. (B.7), and we already know it has rank n_x , provided the system is fully observable. The second term on the right-hand side has a maximum rank of n_x . Now, from matrix algebra results, we know that the rank of a matrix product cannot be greater than the smallest rank of the multiplied matrices, which implies that the rest matrix of Eq. (B.12) has a maximum rank of n_x . Accordingly, the system matrix \mathcal{A} of Eq. (B.11) has a maximum rank of $(n_y + n_x)$. This implies that a maximum of $(n_y + n_x)$ terms can be determined independently and not $(n_x + n_u + n_y)$, as ideally desired.

References

¹Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.

²Plaetschke, E., "Maximum-Likelihood-Verfahren," Lecture Notes: "Praktische Einführung in die Systemidentifizierung," Carl-Cranz-Gesellschaft, Oberpfaffenhofen, 11–13 June 1985, Section 5 (in German).

This page intentionally left blank

Appendix C

Derivation of the Likelihood Function

THE PRINCIPLE of likelihood is based on a probability density function.

Although the method is applicable to any form of the density function distribution, for mathematical tractability we consider the Gaussian (normal) distribution, which is completely determined by the mean and covariance matrix. It is the most widely used assumption in practical cases.

The probability density function of a Gaussian (normally) distributed, scalar real random variable x is given by^{1,2}

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right] \quad (\text{C.1})$$

where $m = E\{x\}$ and $\sigma^2 = E\{(x-m)^2\}$ denote the mean value and variance respectively, and E the expected value.

Consider two statistically independent, Gaussian variables x_1 and x_2 with zero mean and variances σ_1 and σ_2 , respectively. Then the joint probability density function is given by

$$\begin{aligned} p(x_1, x_2) &= p(x_1) \cdot p(x_2) \\ &= \frac{1}{2\pi\sigma_1\sigma_2} \exp\left[-\frac{x_1^2}{2\sigma_1^2} - \frac{x_2^2}{2\sigma_2^2}\right] \end{aligned} \quad (\text{C.2})$$

Define two random variables y_1 and y_2 , which are obtained from x_1 and x_2 through the rotational transformation

$$\begin{aligned} y_1 &= x_1 \cos \varphi - x_2 \sin \varphi \\ y_2 &= x_1 \sin \varphi + x_2 \cos \varphi \end{aligned} \quad (\text{C.3})$$

satisfying the condition $(x_1^2 + x_2^2) = (y_1^2 + y_2^2)$. Since we assumed that the random variables x_1 and x_2 have zero mean, the mean values of the random variables y_1 and y_2 are also zero. Their variances are given by the expected values of Eq. (C.3):

$$\begin{aligned} \mu_{11} &= E\{y_1^2\} = \sigma_1^2 \cos^2 \varphi + \sigma_2^2 \sin^2 \varphi \\ \mu_{22} &= E\{y_2^2\} = \sigma_1^2 \sin^2 \varphi + \sigma_2^2 \cos^2 \varphi \end{aligned} \quad (\text{C.4})$$

and the nonzero covariance is given by

$$\mu_{12} = E\{y_1 y_2\} = (\sigma_1^2 - \sigma_2^2) \sin \varphi \cos \varphi \quad (C.5)$$

The joint probability density function of y_1 and y_2 is now obtained as

$$p_2(y_1, y_2) = p_1\{x_1(y_1, y_2), x_2(y_1, y_2)\} \begin{vmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_2}{\partial y_1} \\ \frac{\partial x_1}{\partial y_2} & \frac{\partial x_2}{\partial y_2} \end{vmatrix} = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left[-\frac{(y_1 \cos \varphi + y_2 \sin \varphi)^2}{2\sigma_1^2} - \frac{(-y_1 \sin \varphi + y_2 \cos \varphi)^2}{2\sigma_2^2}\right] \quad (C.6)$$

In terms of the second moments of y_1 and y_2 , the above expression can be expressed as

$$p(y_1, y_2) = \frac{1}{2\pi(\mu_{11}\mu_{22} - \mu_{12}^2)^{\frac{1}{2}}} \exp\left[\frac{-\mu_{22}y_1^2 + 2\mu_{12}y_1y_2 - \mu_{11}y_2^2}{2(\mu_{11}\mu_{22} - \mu_{12}^2)}\right] \quad (C.7)$$

Combining the two variables y_1 and y_2 into a vector y and the second moments into covariance matrix R , that is,

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}; \quad R = \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{12} & \mu_{22} \end{bmatrix} \quad (C.8)$$

the probability density function can be expressed in the following concise form:

$$p(y_1, y_2) = \frac{1}{2\pi\sqrt{|R|}} \exp\left[-\frac{1}{2}y^T R^{-1} y\right] \quad (C.9)$$

This result can be readily extended to the multidimensional distribution. The joint probability distribution function of n Gaussian distributed random variables x_i with mean m_i is given by

$$p(x_1, \dots, x_n) = \frac{1}{(2\pi)^{n/2} \sqrt{|R|}} \exp\left[-\frac{1}{2}(x - m)^T R^{-1} (x - m)\right] \quad (C.10)$$

where $x^T = (x_1, \dots, x_n)$ and $m^T = (m_1, \dots, m_n)$ denote the vectors of random variables and of the mean values respectively, and

$$R = \begin{bmatrix} r_{11} & \cdot & \cdot & \cdot & r_{1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ r_{1n} & \cdot & \cdot & \cdot & r_{nn} \end{bmatrix} \quad (\text{C.11})$$

the covariance matrix. The elements of R are the second moments

$$r_{ij} = E\{(x_i - m_i)(x_j - m_j)\} = \sigma_i \sigma_j \rho_{ij} \quad (\text{C.12})$$

where ρ_{ij} are called the correlation coefficients, which automatically implies that $\rho_{ii} = 1$.

The likelihood function is now defined as the conditional probability density $p[z(t_1), \dots, z(t_N) | \Theta, R]$ of the n_y dimensional measurement vector at N discrete time points for a given parameter vector Θ and given measurement error covariance matrix R . The conditional probability density of the measurements at a particular measurement time point t_k is given by [using Eq. (C.10) derived above]

$$p[z(t_k) | \Theta, R] = \frac{1}{(2\pi)^{n_y/2} \sqrt{|R|}} \times \exp \left[-\frac{1}{2} \{z(t_k) - y(t_k)\}^T R^{-1} \{z(t_k) - y(t_k)\} \right] \quad (\text{C.13})$$

where y is the vector of observation variables (model output).

Assuming the error $v = z - y$ at different time points to be statistically independent,

$$E\{v(t_k)v^T(t_\ell)\} = R\delta_{k\ell} \quad (\text{C.14})$$

results in the following likelihood function

$$\begin{aligned} p[z(t_1), \dots, z(t_N) | \Theta, R] &= \prod_{k=1}^N p[z(t_k) | \Theta, R] \\ &= \{(2\pi)^{n_y} |R|\}^{-N/2} \exp \left[-\frac{1}{2} \sum_{k=1}^N [z(t_k) - y(t_k)]^T R^{-1} [z(t_k) - y(t_k)] \right] \quad (\text{C.15}) \end{aligned}$$

References

¹Davenport, W. B. and Root, W. L., *Introduction to the Theory of Random Signals and Noise*, McGraw-Hill, New York, 1958.

²Plaetschke, E., "Maximum-Likelihood-Verfahren," Lecture Notes: "Praktische Einführung in die Systemidentifizierung," Carl-Cranz-Gesellschaft, Oberpfaffenhofen, 11–13 June 1985, Section 5 (in German).

This page intentionally left blank

Appendix D

Statistical Properties of Maximum Likelihood Estimates

I. Asymptotic Consistency

THE MAXIMUM likelihood estimates $\hat{\Theta}_{ML}$ are asymptotically consistent, that is, $\hat{\Theta}_{ML}$ converges in probability to the true values Θ . In the following, we investigate this property.

We know from the properties of the probability functions that

$$\int p(z|\Theta) dz = 1 \quad (D.1)$$

Partial differentiation of Eq. (D.1) with respect to Θ , and interchanging the order of integration and differentiation assuming sufficient regularity conditions,¹ yields

$$\int \frac{\partial p(z|\Theta)}{\partial \Theta} dz = 0 \quad (D.2)$$

Equation (D.2) can be rewritten as

$$\int \left[\frac{1}{p(z|\Theta)} \frac{\partial p(z|\Theta)}{\partial \Theta} \right] p(z|\Theta) dz = \int \frac{\partial \ln p(z|\Theta)}{\partial \Theta} p(z|\Theta) dz = 0 \quad (D.3)$$

or equivalently,

$$E \left\{ \frac{\partial \ln p(z|\Theta)}{\partial \Theta} \right\} = 0 \quad (D.4)$$

Differentiating Eq. (D.3) and rewriting yields

$$\int \left\{ \left(\frac{\partial \ln p(z|\Theta)}{\partial \Theta} \right) \left(\frac{\partial \ln p(z|\Theta)}{\partial \Theta} \right)^T + \frac{\partial^2 \ln p(z|\Theta)}{\partial \Theta^2} \right\} p(z|\Theta) dz = 0 \quad (D.5)$$

or

$$J = E \left\{ \left[\frac{\partial \ln p(z|\Theta)}{\partial \Theta} \right] \left[\frac{\partial \ln p(z|\Theta)}{\partial \Theta} \right]^T \right\} = E \left\{ - \frac{\partial^2 \ln p(z|\Theta)}{\partial \Theta^2} \right\} \quad (\text{D.6})$$

The Fisher information matrix J , defined in Eq. (D.6) is generally positive definite. However, if the observations are independent of Θ , that is, $p(z|\Theta)$ is not a function of Θ as assumed, then J in this case will reduce to zero. In practice, this implies that it would not be possible to estimate Θ from sample observations which do not contain information about Θ .

The Taylor series expansion of the term $[\partial \ln p(z|\Theta)/\partial \Theta]$ in the above equation about the true values Θ evaluated at $\hat{\Theta}_{\text{ML}}$, leads to

$$\frac{\partial \ln p(z|\hat{\Theta}_{\text{ML}})}{\partial \Theta} = \frac{\partial \ln p(z|\Theta)}{\partial \Theta} + \frac{\partial^2 \ln p(z|\Theta^*)}{\partial \Theta^2} (\hat{\Theta}_{\text{ML}} - \Theta) \quad (\text{D.7})$$

where $\Theta^* = \lambda \Theta + (1 - \lambda) \hat{\Theta}_{\text{ML}}$; $0 \leq \lambda \leq 1$.

Since $\hat{\Theta}_{\text{ML}}$ is the solution of the likelihood equation, equating Eq. (D.7) to zero yields

$$\frac{\partial \ln p(z|\Theta)}{\partial \Theta} = - \frac{\partial^2 \ln p(z|\Theta^*)}{\partial \Theta^2} (\hat{\Theta}_{\text{ML}} - \Theta) \quad (\text{D.8})$$

We recall our assumption that the measurements at different time points are assumed to be statistically independent. This provides

$$\begin{aligned} \frac{\partial \ln p(z|\Theta)}{\partial \Theta} &= \frac{\partial}{\partial \Theta} \left[\ln \prod_{k=1}^N p(z_k|\Theta) \right] \\ &= \frac{\partial}{\partial \Theta} \left[\sum_{k=1}^N \ln p(z_k|\Theta) \right] \\ &= \sum_{k=1}^N \frac{\partial}{\partial \Theta} \ln p(z_k|\Theta) \end{aligned} \quad (\text{D.9})$$

Similarly,

$$\frac{\partial^2 \ln p(z|\Theta)}{\partial \Theta^2} = \sum_{k=1}^N \frac{\partial^2 \ln p(z_k|\Theta)}{\partial \Theta^2} \quad (\text{D.10})$$

Now, from Eqs. (D.9) and (D.10), the strong law of large numbers, which indicates that the sample average converges in probability to the ensemble

average, yields

$$\frac{1}{N} \sum_{k=1}^N \frac{\partial \ln p(z_k | \Theta)}{\partial \Theta} \rightarrow E \left\{ \frac{\partial \ln p(z_k | \Theta)}{\partial \Theta} \right\} \quad (\text{D.11})$$

and

$$\frac{1}{N} \sum_{k=1}^N \frac{\partial^2 \ln p(z_k | \Theta)}{\partial \Theta^2} \rightarrow E \left\{ \frac{\partial^2 \ln p(z_k | \Theta)}{\partial \Theta^2} \right\} \quad (\text{D.12})$$

Further it can be shown that the likelihood function is concave in Θ .^{2,3} Hence, it is appropriate to assume that the matrix $E\{\partial^2 \ln p(z_k | \Theta) / \partial \Theta^2\}$ is positive definite. From Eqs. (D.8), (D.11), and (D.12) it directly follows that

$$\hat{\Theta}_{\text{ML}} \rightarrow \Theta \quad (\text{D.13})$$

with probability of one. Thus, $\hat{\Theta}_{\text{ML}}$ is consistent.

II. Asymptotic Normality

The asymptotic normality implies that for large number of data points N , the maximum likelihood estimates $\hat{\Theta}_{\text{ML}}$ converge to normal distribution, that is,

$$\sqrt{N}(\hat{\Theta}_{\text{ML}} - \Theta) \rightarrow r_1 \sim \mathcal{N}(0, \bar{J}^{-1}) \quad (\text{D.14})$$

where \bar{J} is the average Fisher information matrix per sample and Θ the true parameter values. In Eq. (A.4.14) we have used the standard notation used in statistics which implies that the term on the left-hand side tends to random variables denoted by r_1 , having $\mathcal{N}(0, \bar{J}^{-1})$ distribution, that is, normal (Gaussian) distribution with zero mean and variance \bar{J}^{-1} . In the following we investigate this property.

The Taylor series expansion of $[\partial \ln p(z | \hat{\Theta}_{\text{ML}}) / \partial \Theta]$ about the true parameter value Θ yields

$$\begin{aligned} \frac{\partial \ln p(z | \hat{\Theta}_{\text{ML}})}{\partial \Theta} &= \frac{\partial \ln p(z | \Theta)}{\partial \Theta} + \frac{\partial^2 \ln p(z | \Theta)}{\partial \Theta^2} (\hat{\Theta}_{\text{ML}} - \Theta) \\ &\quad + \text{higher order terms} \end{aligned} \quad (\text{D.15})$$

Using the property that $\hat{\Theta}_{\text{ML}}$ are asymptotically consistent, higher-order terms can be neglected. Since $\hat{\Theta}_{\text{ML}}$ satisfies the likelihood equation, equating Eq. (D.15) to zero gives

$$\frac{1}{\sqrt{N}} \frac{\partial \ln p(z | \Theta)}{\partial \Theta} = \sqrt{N} \left\{ -\frac{1}{N} \frac{\partial^2 \ln p(z | \Theta)}{\partial \Theta^2} \right\} (\hat{\Theta}_{\text{ML}} - \Theta) \quad (\text{D.16})$$

Once again, using Eqs. (D.6) and (D.12), the strong law of large numbers leads to

$$-\frac{1}{N} \frac{\partial^2 \ln p(z|\Theta)}{\partial \Theta^2} \rightarrow E\left\{-\frac{\partial^2 \ln p(z_k|\Theta)}{\partial \Theta^2}\right\} = \bar{J} \quad (\text{D.17})$$

where $\bar{J} = J/N$ represents the average information matrix per sample, and J the Fisher information matrix.

The assumption that the samples z_k are independent allows the left-hand side of Eq. (D.16) to be written as

$$\frac{1}{\sqrt{N}} \frac{\partial \ln p(z|\Theta)}{\partial \Theta} = \frac{1}{\sqrt{N}} \sum_{k=1}^N \frac{\partial \ln p(z_k|\Theta)}{\partial \Theta} \quad (\text{D.18})$$

Now, it is already known from Eqs. (D.4) and (D.6) that

$$E\left\{\frac{\partial \ln p(z_k|\Theta)}{\partial \Theta}\right\} = 0 \quad (\text{D.19})$$

and

$$E\left\{\left[\frac{\partial \ln p(z_k|\Theta)}{\partial \Theta}\right] \left[\frac{\partial \ln p(z_k|\Theta)}{\partial \Theta}\right]^T\right\} = \bar{J} \quad (\text{D.20})$$

Using Eqs. (D.19) and (D.20), and the central limit theorem, it follows from Eq. (D.18) that

$$\frac{1}{\sqrt{N}} \frac{\partial \ln p(z|\Theta)}{\partial \Theta} \rightarrow r_2 \sim \mathcal{N}(0, \bar{J}) \quad (\text{D.21})$$

Now, Eqs. (D.16) and (D.17) yield

$$\sqrt{N}(\hat{\Theta}_{\text{ML}} - \Theta) \rightarrow r_2 \sim \mathcal{N}(0, \bar{J}) \quad (\text{D.22})$$

which can be equivalently expressed as:

$$\sqrt{N}(\hat{\Theta}_{\text{ML}} - \Theta) \rightarrow r_1 \sim \mathcal{N}(0, \bar{J}^{-1}) \quad (\text{D.23})$$

The property of asymptotic normality implies that the estimates $\hat{\Theta}$ obtained from different sets of data samples corresponding to different experiments are clustered around Θ with a normal distribution.

III. Asymptotic Efficiency

The maximum likelihood estimates $\hat{\Theta}_{ML}$ are asymptotically efficient in the sense that they attain Cramér–Rao lower bounds. In the following we study the property of asymptotic efficiency.

To establish the asymptotic efficiency of $\hat{\Theta}_{ML}$, consider a general estimator $\hat{\Theta}(z)$, not necessarily the maximum likelihood. This gives

$$E\{\hat{\Theta}\} = \int \hat{\Theta} p(z|\Theta) dz = \Theta + b(\Theta) \quad (D.24)$$

where Θ represents the true values of the parameters and $b(\Theta)$ the bias that may result in the estimates.

Differentiation of Eq. (D.24) yields

$$\int \hat{\Theta} \left[\frac{\partial p(z|\Theta)}{\partial \Theta} \right]^T dz = I + \frac{\partial b(\Theta)}{\partial \Theta} \quad (D.25)$$

which can be rewritten as

$$\int \hat{\Theta} \left[\frac{\partial \ln p(z|\Theta)}{\partial \Theta} \right]^T p(z|\Theta) dz = I + \frac{\partial b(\Theta)}{\partial \Theta} \quad (D.26)$$

Once again starting from the basic relation,

$$\int p(z|\Theta) dz = 1 \quad (D.27)$$

differentiation of Eq. (D.27) yields

$$\int \frac{\partial p(z|\Theta)}{\partial \Theta} dz = 0 \quad (D.28)$$

which can be rewritten as

$$\int \left[\frac{1}{p(z|\Theta)} \frac{\partial p(z|\Theta)}{\partial \Theta} \right] p(z|\Theta) dz = \int \frac{\partial \ln p(z|\Theta)}{\partial \Theta} p(z|\Theta) dz = 0 \quad (D.29)$$

Multiplying Eq. (D.29) with $\bar{\Theta}$, the expected value of $\hat{\Theta}$, that is, $\bar{\Theta} = E\{\hat{\Theta}\}$, and subtracting from Eq. (D.26) gives

$$\int (\hat{\Theta} - \bar{\Theta}) \left[\frac{\partial \ln p(z|\Theta)}{\partial \Theta} \right]^T p(z|\Theta) dz = I + \frac{\partial b(\Theta)}{\partial \Theta} \quad (D.30)$$

Using the Cauchy–Schwarz inequality for two integrable quadratic functions $f(x)$ and $g(x)$, namely

$$\left[\int f(x)g(x) dx \right]^2 - \int f^2(x) dx \int g^2(x) dx \leq 0 \quad (\text{D.31})$$

it follows from Eq. (D30) that

$$\begin{aligned} & \int (\hat{\Theta} - \bar{\Theta})(\hat{\Theta} - \bar{\Theta})^T p(z|\Theta) dz \left\{ \int \left[\frac{\partial \ln p(z|\Theta)}{\partial \Theta} \right] \left[\frac{\partial \ln p(z|\Theta)}{\partial \Theta} \right]^T p(z|\Theta) dz \right\} \\ & \geq \left[I + \frac{\partial b(\Theta)}{\partial \Theta} \right]^2 \end{aligned} \quad (\text{D.32})$$

However, the second term on the left-hand side of the inequality is defined as the information matrix J .

Using Eqs. (D.6) and (D.24), Eq. (D.32) leads to

$$\text{cov}(\bar{\Theta}) = E\{(\hat{\Theta} - \bar{\Theta})(\hat{\Theta} - \bar{\Theta})^T\} \geq \left(I + \frac{\partial b(\Theta)}{\partial \Theta} \right) J^{-1} \quad (\text{D.33})$$

The expression in Eq. (D.33) is called the Cramér–Rao inequality.

Since the maximum likelihood estimates are asymptotically bias free, it follows that

$$E\{\hat{\Theta}_{\text{ML}}\} = \Theta \quad \text{and} \quad b(\hat{\Theta}_{\text{ML}}) = 0 \quad (\text{D.34})$$

Therefore, Eq. (D.33) yields

$$\text{cov}(\hat{\Theta}_{\text{ML}}) \geq J^{-1} \quad (\text{D.35})$$

It has already been shown in Secs. I and II that $\hat{\Theta}_{\text{ML}}$ are asymptotically consistent and normal. Combining these results with Eq. (D.35), it follows that $\hat{\Theta}_{\text{ML}}$ is asymptotically efficient in the sense of achieving the Cramér–Rao lower bound.

The property of asymptotic efficiency is of practical significance. It implies that the maximum likelihood estimator makes efficient use of the available data. The Cramér–Rao lower bound indicates the theoretically maximum achievable accuracy of the estimates.

References

¹Cramér, H., *Mathematical Methods of Statistics*, Princeton University Press, Princeton, NJ, 1946.

²Kashyap, R. L., "Maximum Likelihood Identification of Stochastic Linear systems," *IEEE Transactions on Automatic Control*, Vol. AC-15, No. 1, 1970, pp. 25–34.

³Wilks, S. S., *Mathematical Statistics*, John Wiley & Sons, New York, 1962.

Appendix E

Minimization of Likelihood Function with Respect to Covariance Matrix R

IN CHAPTERS 4 and 5, parameter estimation applying the maximum likelihood method required minimization of the likelihood cost function given by

$$L(z | \Theta, R) = \frac{1}{2} \sum_{k=1}^N v^T R^{-1} v + \frac{N}{2} \ln(|R|) + \frac{N n_y}{2} \ln(2\pi) \quad (\text{E.1})$$

where $v(t_k) = z(t_k) - y(t_k)$ represents the residuals at the discrete time point t_k .

For a specified model structure and the data being analyzed, the number of observations n_y and the data points N to be evaluated are defined and fixed. Accordingly, the last term in the above equation is a constant and, hence, neglected without affecting minimization.

The first and the second terms on the right-hand side of Eq. (E.1) contain R^{-1} and $|R|$, respectively. To facilitate the derivation, we rewrite the right-hand side such that both terms contain R^{-1} . Using the matrix expression

$$|R| = \frac{1}{|R^{-1}|} \quad (\text{E.2})$$

and after minor simplifications, the above equation can be rewritten as

$$L(z | \Theta, R) = \frac{1}{2} \sum_k v^T R^{-1} v - \frac{N}{2} \ln(|R^{-1}|) \quad (\text{E.3})$$

Now, writing Eq. (E.3) in a component form, and denoting the elements of R^{-1} as a_{ij} leads to

$$L = \frac{1}{2} \sum_k \sum_i \sum_j a_{ij} v_i v_j - \frac{N}{2} \ln \left(\sum_i a_{ii} A_{ii} \right) \quad (\text{E.4})$$

where we used the matrix algebra result

$$|A| = \sum_i a_{ii} A_{ii} \quad (\text{E.5})$$

in which A_{ij} denote the cofactor of the element a_{ij} of a matrix A .

Partial differentiation of Eq. (E.4) and making use of the results

$$\frac{\partial(\ln y)}{\partial x} = \frac{1}{y} \frac{\partial y}{\partial x} \quad (\text{E.6})$$

leads to

$$\frac{\partial L}{\partial a_{ij}} = \frac{1}{2} \sum_k v_i v_j - \frac{N}{2} \frac{1}{(\sum_i a_{ij} A_{ij})} A_{ij} \quad (\text{E.7})$$

From Eq. (E.5) we know that the quantity in the denominator of the second term on the right-hand side of Eq. (E.7) is nothing but $|A|$. For the minimum of L with respect to the measurement error covariance matrix R , the partial derivative of L with respect to the elements a_{ij} of R^{-1} must be zero. Accordingly, setting the right-hand side of Eq. (E.7) to zero, we get

$$\frac{1}{2} \sum_k v_i v_j - \frac{N}{2} \frac{1}{|A|} A_{ij} = 0 \quad (\text{E.8})$$

Now making use of another matrix algebra result,

$$(A^{-1})_{ij} = \frac{A_{ji}}{|A|} \quad (\text{E.9})$$

Eq. (E.8) leads to

$$\frac{1}{2} \sum_k v_i v_j = \frac{N}{2} r_{ji} \quad (\text{E.10})$$

Thus, for each element r_{ij} , we get

$$r_{ij} = \frac{1}{N} \sum_k v_j v_i \quad (\text{E.11})$$

which can be rewritten in the matrix notation as

$$R = \frac{1}{N} \sum_{k=1}^N v(k) v^T(k) \quad (\text{E.12})$$

which is the solution that we are interested in.^{1,2} For any fixed value of the parameter vector Θ , Eq. (E.12) maximizes the likelihood function with respect to R .

It is also possible to derive the same expression using matrix differentiation results, see Ref. 3.

References

- ¹Plaetschke, E., "Maximum-Likelihood-Verfahren," Lecture Notes: "Praktische Einführung in die Systemidentifizierung," Carl-Cranz-Gesellschaft, Oberpfaffenhofen, Germany, 11–13 June 1985, Section 5 (in German).
- ²Groves, R. D., Bowles, R. L., and Mayhew, S. C., "A Procedure for Estimating Stability and Control Parameters from Flight Test Data by Using Maximum Likelihood Method Employing Real-Time Digital System," NASA TN D-6735, May 1972.
- ³Goodwin G. C. and Payne R. L., *Dynamic System Identification*, Academic Press, New York, 1977.

This page intentionally left blank

Appendix F

Derivation of Kalman Filter and Extended Kalman Filter

WHEN THE process under investigation contains stochastic nonmeasurable inputs, we cannot propagate the states by solving an initial-value problem through simple numerical integration as in the case of deterministic systems. Instead, we have to estimate the states using a suitable state estimator. The Kalman filter provides an efficient computational procedure to estimate the states of a linear system.¹ It has two main advantages: 1) it is optimal in the least mean squares sense, yielding a minimum variance estimator, and 2) it operates recursively, making it suitable for computer implementations. For a dynamic system represented in state space, the Kalman filter consists of two steps: 1) prediction, also called extrapolation, propagation or time update, and 2) correction, also called measurement update. The first step deals with propagating the states and the errors in the state prediction between the two discrete time points k to $k + 1$. The second step corrects a posteriori these predictions once the new measurements at sample time point $k + 1$ become available; see Fig. F.1, where x is the state vector, y the model outputs and z the corresponding measurements. The predicted states are denoted by the superscript \sim (tilde) and the corrected states by \hat{x} (hat). There are different notations used in the literature on estimation theory to denote the predicted and corrected states, as well as different derivations of the Kalman filter considering purely continuous-time or discrete-time or mixed continuous–discrete representations.^{2–6} In our specific case, we are faced with the last type, namely continuous–discrete representations.

Extending our brief discussion in Chapter 3, Sec. VIII.B and using the discrete-time theory of linear dynamic systems, it can be shown that the solution to a linear stochastic system represented by

$$\dot{x}(t) = Ax(t) + Bu(t) + Fw(t) \quad (\text{F.1})$$

$$y(t) = C(t)x(t) + Du(t) \quad (\text{F.2})$$

is given by

$$x(k + 1) = \Phi x(k) + \Psi Bu(k) + \Psi(\Delta t)^{-1/2}Fw(k) \quad (\text{F.3})$$

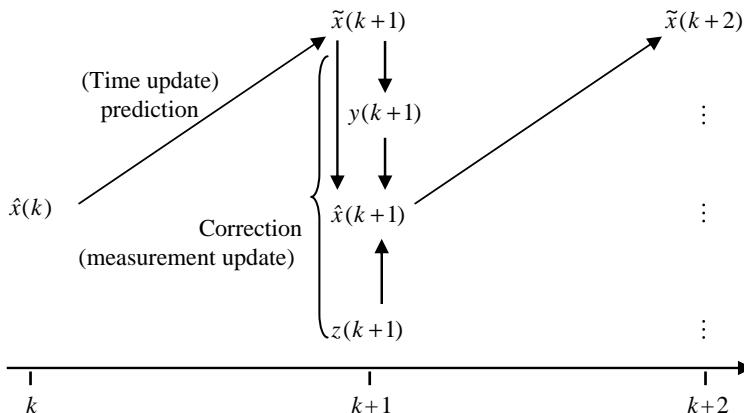


Fig. F.1 Schematic of Kalman filter computations.

where the process noise, $w(t)$, is assumed to be zero-mean white Gaussian-noise with an identity power spectral density matrix and F the process noise distribution matrix. The corresponding discrete-time white-noise process $w(k)$ has to have the covariance $1/\Delta t$ times the identity matrix. This is reflected in Eq. (F.3) through the factor $1/\sqrt{\Delta t}$ associated with F , which is necessary to account for the transition between the continuous-time and discrete-time representations.³ It is also critical to get the limit of the discrete-time system into continuous time; otherwise in the limit case the system will not react to noise, which is physically not realistic. A more detailed treatment of this subtle issue is provided by Maine and Iliff.⁶ We specifically stress that $w(t)$ and $w(k)$ are not the same, although we use the same symbol w for the two processes.

As already defined in Chapter 3, Sec. VIII.B, the state transition matrices Φ , and its integral Ψ are given by

$$\Phi = e^{A\Delta t} \approx I + A\Delta t + A^2 \frac{\Delta t^2}{2!} + \dots \quad (\text{F.4})$$

$$\Psi = \int_0^{\Delta t} e^{A\tau} d\tau \approx I\Delta t + A \frac{\Delta t^2}{2!} + A^2 \frac{\Delta t^3}{3!} + \dots \quad (\text{F.5})$$

We draw the attention of the readers to the definition of the matrix Ψ in Eq. (F.5), which differs slightly from that used in the literature on estimation theory. Specifically, we do not include the input matrix B in the definition of Ψ . This allows us to use the matrix Ψ of Eq. (F.5) in the last term of Eq. (F.3) as well, but now multiplied by the process noise distribution matrix F . Here we are mainly concerned with time-invariant systems, for which the system matrix A is not a function of time. Accordingly, the transition matrix Φ is also

time-invariant. Otherwise, in a general case, the exact notation of Eq. (F.3) should include $\Phi(k)$ and also for its integral $\Psi(k)$.

Since we cannot measure the process noise $w(t)$, it is not possible to apply directly the solution given by Eq. (F.3) to compute the true states x (the true state is denoted without \sim and without \wedge). Instead, the solution is obtained applying the Kalman filter by predicting the states and errors in their predictions across the time points and correcting them at each time point.

For the prediction, the states are propagated by considering the deterministic part of Eq. (F.1). Accordingly, the time propagation of states is given by

$$\tilde{x}(k+1) = \Phi\hat{x}(k) + \Psi Bu(k) \quad (\text{F.6})$$

where \tilde{x} is the predicted state and \hat{x} the corrected state.

Now, owing to the presence of process noise $w(t)$, the state estimates \tilde{x} are not precise; the uncertainty in their estimates is given by the covariance matrix of the state predictions $\tilde{P}(k+1)$ defined as

$$\tilde{P}(k+1) = E\{[\tilde{x}(k+1) - x(k+1)][\tilde{x}(k+1) - x(k+1)]^T\} \quad (\text{F.7})$$

where $x(k+1)$ is the true state and $\tilde{x}(k+1)$ the estimate at the $(k+1)$ th time point. In Eq. (F.7) we substitute Eq. (F.6) for $\tilde{x}(k+1)$ and Eq. (F.3) for the true state $x(k+1)$, where for notational simplicity we have used $F_d = (\Delta t)^{-1/2}F$. This yields

$$\begin{aligned} \tilde{P}(k+1) &= E\{[\Phi\hat{x}(k) - \Phi x(k) - \Psi F_d w(k)] \\ &\quad \times [\Phi\hat{x}(k) - \Phi x(k) - \Psi F_d w(k)]^T\} \end{aligned} \quad (\text{F.8})$$

where $E\{\cdot\}$ denotes the expected value; the terms $\Psi Bu(k)$ cancel out. Equation (F.8) can be rewritten in terms of the error $\hat{e}(k)$ as

$$\tilde{P}(k+1) = E\{[\Phi\hat{e}(k) - \Psi F_d w(k)][\Phi\hat{e}(k) - \Psi F_d w(k)]^T\} \quad (\text{F.9})$$

where $\hat{e}(k) = \hat{x}(k) - x(k)$, the difference between the updated estimate $\hat{x}(k)$ and the true state $x(k)$. By multiplying out the terms and using linearity of expectations, Eq. (F.9) can be simplified as follows:

$$\begin{aligned} \tilde{P}(k+1) &= E\{\Phi\hat{e}(k)\hat{e}^T(k)\Phi^T - \Phi\hat{e}(k)w^T(k)F_d^T\Psi^T \\ &\quad - \Psi F_d w(k)\hat{e}^T(k)\Phi^T + \Psi F_d w(k)w^T(k)F_d^T\Psi^T\} \\ &= \Phi E\{\hat{e}(k)\hat{e}^T(k)\}\Phi^T - \Phi E\{\hat{e}(k)w^T(k)\}F_d^T\Psi^T \\ &\quad - \Psi F_d E\{w(k)\hat{e}^T(k)\}\Phi^T + \Psi F_d E\{w(k)w^T(k)\}F_d^T\Psi^T \end{aligned} \quad (\text{F.10})$$

Now making use of the assumption that the noise $w(k)$ and errors $\hat{e}(k)$ are uncorrelated, the second and the third terms on the right-hand side in the above

equation vanish, and Eq. (F.10) simplifies to

$$\tilde{P}(k+1) = \Phi E\{\hat{e}(k)\hat{e}^T(k)\}\Phi^T + \Psi F_d E\{w(k)w^T(k)\}F_d^T\Psi^T \quad (\text{F.11})$$

The expected value of $\hat{e}(k)\hat{e}^T(k)$ is nothing but the covariance matrix $\hat{P}(k)$, and by the definition we had made at the beginning, we know that the expected value of $w(k) w^T(k)$ is the identity matrix. Thus, Eq. (F.11) can be rewritten as

$$\tilde{P}(k+1) = \Phi \hat{P}(k) \Phi^T + \Psi F_d F_d^T \Psi^T \quad (\text{F.12})$$

For small sampling time Δt , Eq. (F.5) can be approximated as $\Psi \approx I\Delta t$, and consequently Eq. (F.12) through

$$\tilde{P}(k+1) = \Phi \hat{P}(k) \Phi^T + (\Delta t)^2 F_d F_d^T \quad (\text{F.13})$$

Now back-substituting $F_d = (\Delta t)^{-1/2} F$, Eq. (F.13) reduces to a form in terms of the original process noise distribution matrix F appearing in Eq. (F.1) in which we are really interested, giving

$$\tilde{P}(k+1) = \Phi \hat{P}(k) \Phi^T + \Delta t F F^T \quad (\text{F.14})$$

Thus, Eqs. (F.6) and (F.14) provide a procedure to propagate the states and state estimation errors across the data points.

Making use of Eqs. (F.3) and (F.6), it is fairly straightforward to show that, under the assumptions of error and noise being of zero mean, the error in the state estimates is also zero:

$$\begin{aligned} E\{[\tilde{x}(k+1) - x(k+1)]\} &= E\{\Phi \hat{x}(k) - \Phi x(k) - \Psi F w(k)\} \\ &= E\{\Phi \hat{e}(k) - \Psi F w(k)\} \\ &= \Phi E\{\hat{e}(k)\} - \Psi F E\{w(k)\} \\ &= 0 \end{aligned} \quad (\text{F.15})$$

Equation (F.15) implies that the estimates of the states are unbiased.

Having derived the procedure to propagate the states and prediction errors across the time points, we now need to derive a procedure for the measurement updates at discrete time points. Such a procedure, developed by Kalman, is based on finding an equation to compute updated estimates $\hat{x}(k+1)$ as a linear combination of predicted estimates $\tilde{x}(k+1)$ and the measurements $z(k+1)$. For notational simplicity, in the following we agree to denote the discrete time point at which the measurements become available simply as k instead of $k+1$; this is without loss of generality because the computations are anyway carried out recursively, that is, from time point to time point. Accordingly, the linear correction equation is formulated as

$$\hat{x}(k) = K^*(k) \tilde{x}(k) + K(k) z(k) \quad (\text{F.16})$$

where K^* and K are weighting matrices which are to be determined such that the estimates $\hat{x}(k)$ are unbiased and efficient.

We know that the measurement equation is given by

$$z(k) = Cx(k) + v(k) \quad (\text{F.17})$$

where the control observation matrix D has been dropped from Eq. (F.2) for convenience without loss of generality. Substituting Eq. (F.17) in Eq. (F.16) yields

$$\hat{x}(k) = K^*(k)\tilde{x}(k) + K(k)[Cx(k) + v(k)] \quad (\text{F.18})$$

Now, denoting the errors in the predicted and the corrected states as $\tilde{e}(k)$ and $\hat{e}(k)$, respectively, in Eq. (F.18) we replace $\tilde{x}(k)$ and $\hat{x}(k)$ in terms of the true states $x(k)$ and corresponding errors, that is, substitute $\tilde{x}(k) = x(k) + \tilde{e}(k)$ and $\hat{x}(k) = x(k) + \hat{e}(k)$, respectively. This leads to

$$x(k) + \hat{e}(k) = K^*(k)[x(k) + \tilde{e}(k)] + K(k)[Cx(k) + v(k)] \quad (\text{F.19})$$

which can be simplified to

$$\hat{e}(k) = [K^*(k) + K(k)C - I]x(k) + K^*(k)\tilde{e}(k) + K(k)v(k) \quad (\text{F.20})$$

The estimate $\hat{x}(k)$ will be unbiased provided that the expected value $E\{\hat{e}(k)\}$ is zero. From the assumptions made, we know that the measurement noise is zero mean; and from Eq. (F.15) we also know that the errors $\tilde{e}(k)$ are zero mean. Therefore, the last two terms on the right-hand side of Eq. (F.20) vanish. For any arbitrary nonzero states x , the first term would vanish only when the term in the square bracket was zero, which leads to the necessary condition for unbiased estimates given by

$$K^*(k) = I - K(k)C \quad (\text{F.21})$$

Now substituting Eq. (F.21) in (F.16) yields

$$\begin{aligned} \hat{x}(k) &= [I - K(k)C]\tilde{x}(k) + K(k)z(k) \\ &= \tilde{x}(k) + K(k)[z(k) - C\tilde{x}(k)] \end{aligned} \quad (\text{F.22})$$

Thus, if we can determine a suitable gain matrix $K(k)$, then Eq. (F.22) provides a simple procedure to correct the predicted states using the measurements $z(k)$.

The Kalman gain matrix $K(k)$ is now determined such that the estimates $\hat{x}(k)$ are efficient. The estimates are efficient when the variance of the estimation errors, $\hat{e}(k)$, which is the same as the trace of the estimation error covariance matrix $\hat{P}(k)$, is minimized. Substitution of Eq. (F.21) into Eq. (F.20) leads to

$$\hat{e}(k) = [I - K(k)C]\tilde{e}(k) + K(k)v(k) \quad (\text{F.23})$$

Thus, the covariance matrix $\hat{P}(k)$ is given by

$$\begin{aligned}
 \hat{P}(k) &= E\{\hat{e}(k)\hat{e}^T(k)\} \\
 &= E\{([I - K(k)C]\tilde{e}(k) + K(k)v(k)) \\
 &\quad \times ([I - K(k)C]\tilde{e}(k) + K(k)v(k))^T\} \\
 &= E\{[I - K(k)C]\tilde{e}(k)\tilde{e}^T(k)[I - K(k)C]^T \\
 &\quad + [I - K(k)C]\tilde{e}(k)v^T(k)K^T(k) \\
 &\quad + K(k)v(k)\tilde{e}^T(k)[I - K(k)C]^T \\
 &\quad + K(k)v(k)v^T(k)K^T(k)\}
 \end{aligned} \tag{F.24}$$

The second and the third term on the right-hand of Eq. (F.24) vanish, because the measurement noise $v(k)$ and the state errors $\tilde{e}(k)$ are assumed to be uncorrelated. Furthermore, by definition we know that $E\{v(k)v^T(k)\} = R(k)$ and $E\{\tilde{e}(k)\tilde{e}^T(k)\} = \tilde{P}(k)$. Thus, Eq. (F.24) leads to

$$\hat{P}(k) = [I - K(k)C]\tilde{P}(k)[I - K(k)C]^T + K(k)R(k)K^T(k) \tag{F.25}$$

The cost function to be minimized is then given by

$$J(k) = \text{trace}[\hat{P}(k)] \tag{F.26}$$

Here, we use the following result from matrix algebra for the partial differentiation of a trace:

$$\frac{\partial}{\partial A} \{\text{trace}(ABA^T)\} = 2AB \tag{F.27}$$

where A and B are square and square symmetric matrices, respectively. Substituting Eq. (F.25) in Eq. (F.26), and partial differentiating with respect to $K(k)$, applying the above rule on the two terms on the right-hand side leads to

$$\frac{\partial J(k)}{\partial K(k)} = -2[I - K(k)C]\tilde{P}(k)C^T + 2K(k)R(k) \tag{F.28}$$

Equating Eq. (F.28) to zero for a minimum, and solving for $K(k)$ yields the Kalman gain matrix given by

$$K(k) = \tilde{P}(k)C^T[C\tilde{P}(k)C^T + R(k)]^{-1} \tag{F.29}$$

Let us take a look at the physical significance of the Kalman gain matrix K . The gain matrix given by Eq. (F.29) optimally weighs the influence due to the measurement noise covariance matrix $R(k)$ and that due to the a priori estimate

of the error covariance matrix $\tilde{P}(k)$. It is inversely proportional to $R(k)$, the measurement noise and proportional to $\tilde{P}(k)$, the uncertainty in the estimates. The two limiting cases of Eq. (F.29) are interesting. Although the following discussion is valid for a general case, to better appreciate the influence of the two terms, for convenience we assume that only states are measured, which implies that C is an identity square matrix. For $R(k) \rightarrow 0$, Eq. (F.29) leads to $K(k) = C^{-1}(k)$, implying that the residuals are more heavily weighted. In other words, actual measurements are trusted more than the predicted outputs. In the other case, as $\tilde{P}(k) \rightarrow 0$ it leads to $K(k) = 0$, implying that residuals are less weighted, which is to say that predicted states and as consequence predicted outputs are associated with very low uncertainty. The estimates are trusted more than the actual measurements in such a case.

Having determined the optimal Kalman gain matrix $K(k)$, it now becomes possible to derive an alternative form for the covariance matrix given by Eq. (F.25):

$$\begin{aligned}
 \hat{P}(k) &= [I - K(k)C] \tilde{P}(k) [I - K(k)C]^T + K(k) R(k) K^T(k) \\
 &= [\tilde{P}(k) - K(k)C\tilde{P}(k)][I - K(k)C]^T + K(k)R(k)K^T(k) \\
 &= \tilde{P}(k) - K(k)C\tilde{P}(k) - \tilde{P}(k)C^T K^T(k) \\
 &\quad + K(k)C\tilde{P}(k)C^T K^T(k) + K(k)R(k)K^T(k) \\
 &= [I - K(k)C]\tilde{P}(k) - \tilde{P}(k)C^T K^T(k) \\
 &\quad + \underbrace{K(k)\{C\tilde{P}(k)C^T + R(k)\}K^T(k)}_{\tilde{P}(k)C^T} \\
 &= [I - K(k)C]\tilde{P}(k)
 \end{aligned} \tag{F.30}$$

where Eq. (F.29) was used in the second last step of the above equation. The two forms given by Eqs. (F.25) and (F.30) are equivalent. The form given by Eq. (F.30) is less complex and computationally faster. Hence, it is often used in practice. However, the differencing in Eq. (F.30) and round-off errors may lead to non-positive diagonal elements, which is physically unrealistic. On the other hand, the longer form given by Eq. (F.25) involves on the right-hand side the summation of two symmetric matrices. The first is positive definite and the second is non-negative definite. Thus, the resulting matrix is always positive definite. Hence, we recommend the longer form of Eq. (F.25), which is numerically more stable. In the literature this longer form is called the “Joseph” form.⁷

In some specific cases, owing to computer roundoff errors and ill-conditioning of the intermediate result $C\tilde{P}(k)C^T + R(k)$ for inversion in the gain formula, the Kalman filter may fail.⁵ This may happen more often on machines with smaller word length, for example 16-bit computers used in the past. In such cases the factorization methods will perform better. The Kalman filter implementations using factorization methods are, in general, numerically most stable. Details of such techniques are found in books exclusively addressing the subject of Kalman filtering, for example Ref. 5. For the purpose of offline parameter

Table F.1 Computational steps of Kalman filter for linear systems

Prediction (extrapolation/time update)

$$\begin{aligned}\tilde{x}(k+1) &= \Phi\hat{x}(k) + \Psi B u(k) \\ \tilde{P}(k+1) &\approx \Phi\hat{P}(k)\Phi^T + \Delta t F F^T\end{aligned}$$

where

$$\begin{aligned}\Phi &= e^{A\Delta t} \approx I + A\Delta t + A^2 \frac{\Delta t^2}{2!} + \dots \\ \Psi &= \int_0^{\Delta t} e^{A\tau} d\tau \approx I\Delta t + A \frac{\Delta t^2}{2!} + A^2 \frac{\Delta t^3}{3!} + \dots\end{aligned}$$

Correction (measurement update)

$$\begin{aligned}\tilde{y}(k) &= C\tilde{x}(k) + D u(k) \\ K(k) &= \tilde{P}(k)C^T[\tilde{P}(k)C^T + R(k)]^{-1} \\ \hat{x}(k) &= \tilde{x}(k) + K(k)[z(k) - \tilde{y}(k)] \\ \hat{P}(k) &= [I - K(k)C]\tilde{P}(k)[I - K(k)C]^T + K(k)R(k)K^T(k)\end{aligned}$$

estimation from flight data, performed on a 32-bit machine, the Kalman filter using the Joseph form for the covariance matrix update has proved sufficient in the cases analyzed.

The computational steps of the Kalman filter for a linear dynamic system represented in Eqs. (F.1)–(F.3) are summarized in Table F.1.

Extended Kalman Filter

In contrast to the linear filtering technique based on an optimal Kalman filter, there is no unique solution to nonlinear filtering, that is, state estimation based on a nonlinear model. Optimal nonlinear state estimators involve propagation of non-Gaussian functions. They are significantly more complex than the linear estimators, and in many cases computationally not realizable. A nonoptimal approach using the framework of linear filters leads to nonlinear filters that are useful in practice. Different approximation techniques lead to different forms, which can be broadly classified into three groups: 1) linearized Kalman filter, 2) extended Kalman filter, and 3) iterated Kalman filter.⁸ The extended Kalman filter (EKF) based on linearization of a nonlinear model about the most recent state estimate and covariance is widely used, particularly in applications related to aircraft. Linearization about the most recent estimates helps to minimize the errors. EKF being an approximation using a linear state estimator, it basically represents a procedural approach. We have already covered the linear state estimator in necessary details and summarized it in Table F.1. These computational steps will be adapted in the EKF to nonlinear models.

Let us consider a nonlinear system represented by the following state and observation equations:

$$\dot{x}(t) = f[x(t), u(t), \beta] + Fw(t), \quad x(t_0) = x_0 \quad (\text{F.31})$$

$$y(t) = g[x(t), u(t), \beta] \quad (\text{F.32})$$

where β denotes the system parameters appearing in f and g . As in the preceding case of linear systems, w represents the process noise which is assumed to be additive.

To apply the computational steps of Table F.1, we need to linearize the nonlinear model of Eqs. (F.31) and (F.32). Linearized system matrices are defined as

$$\begin{aligned} A &= \frac{\partial f[x, u, \beta]}{\partial x} & B &= \frac{\partial f[x, u, \beta]}{\partial u} \\ C &= \frac{\partial g[x, u, \beta]}{\partial x} & D &= \frac{\partial g[x, u, \beta]}{\partial u} \end{aligned} \quad (\text{F.33})$$

At any discrete time point k , linearized system matrices defined in Eq. (F.33) can be approximated about the current state estimate using the central difference formulas defined in Chapter 3, Sec. V.C.

Thus, the computational steps of the EKF filter for a nonlinear dynamic system represented in Eqs. (F.31)–(F.32) are the same as those in Table F.1, incorporating the system matrices A , B , C , and D defined in Eq. (F.33) and approximated through the central difference formula; as before the state transition matrix Φ and its integral Ψ are given by Eqs. (F.4) and (F.5), respectively.

The above procedure incorporates linearized matrices in the prediction (time propagation) step and correction step (measurement update) for the states as well as for errors in the states (i.e., for covariance matrices). To retain the system nonlinearities in the state estimation as far as possible, a mixed version is used

Table F.2 Computational steps of EKF for nonlinear systems

Prediction (extrapolation/time update)

$$\begin{aligned} \tilde{x}(k+1) &= \hat{x}(k) + \int_{t_k}^{t_{k+1}} f[x(t), u(t_k), \beta] dt \\ \tilde{P}(k+1) &\approx \Phi(k+1)\hat{P}(k)\Phi^T(k+1) + \Delta t FF^T \end{aligned}$$

where

$$\Phi(k+1) = e^{A(k)\Delta t} \approx I + A(k)\Delta t + A^2(k) \frac{\Delta t^2}{2!} + \dots$$

$$A(k) = \left. \frac{\partial f[x(t), u(t), \beta]}{\partial x} \right|_{x=\hat{x}(k)}$$

Correction (measurement update)

$$\tilde{y}(k) = g[\tilde{x}(k), u(k), \beta]$$

$$K(k) = \tilde{P}(k)C^T[\tilde{C}\tilde{P}(k)C^T + R(k)]^{-1}$$

$$\hat{x}(k) = \tilde{x}(k) + K(k) [z(k) - \tilde{y}(k)]$$

$$\hat{P}(k) = [I - K(k)C]\tilde{P}(k)[I - K(k)C]^T + K(k)R(k)K^T(k)$$

where

$$C(k) = \left. \frac{\partial g[x(t), u(t), \beta]}{\partial x} \right|_{x=\tilde{x}(k)}$$

that incorporates state and model output prediction with a nonlinear mode, whereas the error propagation is based on a linear approximation of the state and measurement equations. Any suitable numerical integration procedure, such as Runge–Kutta methods discussed in Chapter 3, Sec. VIII.A, can be used to propagate the states across the two discrete time points. Computational steps of such a nonlinear filter are summarized in Table F.2. The various steps of the EKF are very similar to those shown in Fig. F.1, except for the computational details.

References

- ¹Kalman, R. E., “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME, Journal of Basic Engineering*, Vol. 82, Series D, March 1960, pp. 35–45.
- ²Sorenson, H. W., “Least-Squares Estimation from Gauss to Kalman,” *IEEE Spectrum*, Vol. 7, July 1970, pp. 63–68.
- ³Gelb, A., *Applied Optimal Control*, MIT Press, Cambridge, MA, 1974.
- ⁴Sorenson, H. W., “Kalman Filtering Techniques,” *Advances in Control Systems: Theory and Applications*, edited by C. T. Leondes, Vol. 3, 1966, Academic Press, New York, pp. 219–292.
- ⁵Grewal, M. S. and Andrews, A. P., *Kalman Filtering Theory and Practice*, Prentice Hall, Upper Saddle River, NJ, 1993.
- ⁶Maine, R. E. and Iliff, K. W., “Identification of Dynamic Systems,” AGARD AG-300, Vol. 2, Jan. 1985, Chapters 6 and 7.
- ⁷Bucy, R. S. and Joseph, P. D., *Filtering for Stochastic Processes with Applications to Guidance*, John Wiley & Sons, New York, 1968.
- ⁸Schmidt, G. S., “Linear and Nonlinear Filtering Techniques,” *Control and Dynamic Systems—Advances in Theory and Applications*, edited by C. T. Leondes, Vol. 12, Academic Press, New York, 1976, pp. 63–95.

Index

- A-optimality criterion, 35
a posteriori probability density function, 109
a priori information, 109–111, 303
acceleration–deceleration maneuver, 27, 29
accelerometer sensor position (AS), 205
accelerometer triple, 49, 341
acceptance test guide (ATG), 26, 389
activation function, 268
active set strategy, 99–100
actuator models, 7
Adams–Bashforth–Moulton method, 72
additive zero-mean noise disturbances, 241
adjusted coefficient of determination, 202–203
Advisory Group for Aerospace Research and Development (AGARD), 15
aerodynamic center (AC), 207
aerodynamic data set (ADS), 327, 330
aerodynamic database, 46, 387–388, 392
aerodynamic derivatives, 10, 12, 67–68, 71
 integration methods, 74
aerodynamic effects, 49
aerodynamic encounter model validation, 452
aerodynamic force, 190, 204–209
 and moment coefficients
 comparison of wind-tunnel-
 predicted and flight-derived, 461
 identification, 462–463
 schematic of data preprocessing, 205
aerodynamic interaction models (AIM), 450, 455
aerodynamic models, 7, 9, 11
 validation and update, basics, 459–460
aerodynamic parameters
 linear model, 196, 207–216
 nonlinear model, 215–216
aeroservoelastic effects, 390
aeroservoelastic models, 6, 8
AGARDGraphs (AG), 15
AGL (above ground level), 365
AIAA Journal of Aircraft, 15
aileron-to-rudder interconnection, 31
ailerons, 30–31
 deflections, 31, 327
 effectiveness, 330
 input maneuver, 31
air data measurements, 50
aircraft state estimation, 336, 372
Akaike’s information criterion, 202
alignment error in pitch axis, 340
all-moving trimmable horizontal tail, 33
American Institute of Aeronautics and Astronautics (AIAA), 14–15
analog matching techniques, 12
analytical wake vortex model, 453
angle-of-attack, 50, 62, 64, 70, 172, 196, 314, 326–329, 331, 338, 342, 344, 354–355, 360–361, 397–398
 errors in linear calibration, 459
angle-of-sideslip, 50, 62, 120, 342, 354–355
angular accelerations, 50, 206
anti-aliasing filter, 52
approximation function, 417
array delay, 66
artificial feel system, 387
artificial neural networks (ANN), 265–293
interconnected structure, 266

- artificial stabilization, 310–311
output error method, 324
- aspect ratio, 215
- asymmetric stall behavior, time histories, 426
- asymmetric stall modeling, 425–427
schematic, 426
- asymmetric trim conditions, 32
- asymptotic approximation, 144
- asymptotic consistency, 83, 184,
501–503
- asymptotic efficiency, 83, 111,
505–506
- asymptotic normality, 83, 503–504
- atmospheric turbulence, 10, 131, 157,
169, 346
- ATTAS aircraft *see* VFW-614 ATTAS
aircraft
- attitude angles, 50, 343
- audio-cuing, 43
- augmented state vector, 234, 241–242
- autocorrelation function, 381
- autocorrelation matrix, 381
- automatic differentiation (AD)
concept of, 94
example of, 95
- automatic gradient computation,
94–95
- auxiliary state equation, 234, 353
- back-propagation (BP), 266, 270
- algorithms, 282–283
learning, 271
modified, 275–276, 283, 287
standard, 272–274
with momentum term, 274–275
schematic, 272
- backslash operator, 182
- back-transformation, 143, 308, 314
- backward differentiation, 75
- backward elimination, 198
- backward pass, 271
- behavioral models, 5
- bank-to-bank maneuver, 27, 29,
328–329
- bank-to-bank motion, 208, 354
- bank-to-bank roll, 30
- barrier function approach, 99
- barrier parameter, 101, 103
- Bayes' rule, 109
- Bayesian estimator, 109
- bias corrections, 358
- bias parameters, 120, 133, 164, 343
identifiability of, 493–495
- bias proportion, 381
- bias vector, 211
- black-box approach, input–output
matching, 288
- black-box models, 6
- BO-105 model validation in frequency
domain, 474
- BO-105 S123 flying rotorcraft test bed,
472
- Bode magnitude plot, 38
- Bode plot of measured system response,
391
- body-fixed accelerations, 205
- body-fixed coordinates, 339, 341
- body-fixed rotational rates, 337
- Boolean functions, 265
- bounded-variable Gauss–Newton
method, 96, 98–100
- Burnham–Hallock model, 453–454
- C-160 *see* Transall C-160 aircraft
- calibration factor, 338
- canard, pitching moment due to, 315
- canard deflection, 314
- Carrot formula, 491
- cause-effect relations, 177, 210, 267, 335
- center of gravity (CG), 205, 207,
326–327, 339–340
- central difference formula, 69, 93,
148–149, 153
- chain rule, 95
- Cholesky factorization, 91, 103, 161, 182
- climb/sawtooth climb, 28
- closed loop identification, 295
- closed-loop simulation, 317
- coefficient of determination, 202
- coefficient of rolling and yawing
moments, 326
- collinearity, 198 *see also* data collinearity
- colored noise, 377
- combined output error and least squares
method, 304
- combined yaw damping, 330
- complementary data, 13, 386
- computational fluid dynamics (CFD), 6,
8, 10
- computational unsteady aerodynamics
(CUA), 8
- condition index, 301
- condition number, 300

- conditional probability density function, 109
- consistency concept, 83
- constant parameters, 67–68
- constrained Gauss–Newton method, 100–103
- constrained optimization, 99, 100, 106, 161, 312
- control derivative, 173
- control dynamics, 7
- control input vector, 70
- control inputs, 45
- control problem, 3
- control research, 11
- control strategy, 104
- control surface
- deflections, 49, 315, 387
 - effectiveness, 315
 - malfuction effects, 411–413
- convergence criterion, 116, 163
- convergence plot of dimensional derivatives, 122, 124
- convergence problems, 137
- correction step, 134, 236, 308–309, 347
- correlation coefficients, 111, 200–201, 300, 376–377, 499
- correlation matrix, 105, 300
- cost function, 86, 88–89, 96, 100, 134, 136, 144, 149, 151, 177, 181, 187–188, 378
- minimization, 190
 - optimization, 86–87
- covariance matrix, 82, 84, 87, 102, 135, 140–145, 153, 184, 210, 507–509, 516
- covariance proportion, 381
- CPU time, 112–113
- Cramér–Rao bounds, 83, 111, 376
- Cramér–Rao inequality, 506
- D-optimality, 35
- damping coefficients, 103
- damping derivatives, 403
- data acquisition process, 12–13
- data analysis, 13, 451
- data channels, 52
- data collinearity, 296–297, 331–332
- detection of, 300–301
 - estimation in presence of, 301–303
 - reduction, 315
- data compatibility check, 335–374
- aim of, 335
 - example, 354–358
 - kinematic model, 336–344
 - time histories of input variables, 356
- data estimation
- in seemingly steady atmosphere, 171–174
 - parameter, 33
- data gathering, 25–58
- important aspects, 25
- data partitioning, 196–197, 331
- schematic, 197
- data processing, 64
- general flow chart, 65
- data recording, 52
- data reduction, 52
- databases, 46, 387–388, 392
- hinge moment, 387
 - preflight aerodynamic (ADB), 460–462
 - training simulators, 433–447
- degrees-of-freedom, 71, 387, 468, 472–473
- delay array, 64
- differential feedback, 299
- digital filters, 53
- dihedral effect, 330, 449
- dimensional derivatives, convergence plot of, 122, 124
- direct lift control (DLC) flaps, 33
- aerodynamic effects, 403–406
 - static characteristics, 406
- direct search methods, 105–107
- directional stability, 449
- discrete Fourier transform, 230, 232
- discrete-time theory, 75
- Dornier 328 aircraft, 419, 441–447
- stall modeling, 422–425
- doublet inputs, energy spectra of, 41
- downwash, 64, 396
- downwash lag effect, 396
- drag estimation, 51–52
- Dryden spectrum, 157, 159
- Dutch roll maneuver, 30–31, 44, 208, 255, 354, 441
- dynamic effects, 66
- dynamic pressure, 363
- dynamic stability, 11
- dynamic system, 84
- deterministic, 85
 - representation of, 2
- dynamic thrust input maneuvers, 30

- EC-135 FHS flying helicopter simulator, 467, 469
efficiency concept, 83
EFRLS (extended forgetting factor recursive least squares), 243–246, 252
eigensystem analysis, 300, 306
eigenvalue transformation method, 306–308, 321
eigenvalues, 301, 317, 324–325, 385
 S-plane plot, 384
eigenvectors, typical plot, 385
eigenvector decomposition, 139
EKF method *see* extended Kalman filter (EKF)
elevator control system, schematic, 442
elevator input and roll maneuvers, 401
energy spectra
 3-2-1-1 input, 42
 doublet inputs, 41
 pulse inputs, 40
engine parameters, 51–52
end-to-end match, 387, 445
equation decoupling method, 296, 304–306, 323
equation error covariance, 231
equation error methods, 79, 177–218
equations of motion, 8, 11, 60, 112, 219, 336
error bounds, 377
error covariance matrix, 376, 517
error feedback, Kalman filter, 370, 372
error function, 277, 391
error-in-variable (EIV) modeling, 191
estimation-before-modeling (EBM), 336
 block structure of system matrix for state estimation using, 354
 methodology, 350–353
estimation error analysis, imput design by, 33–36
estimation error criterion, 55
estimation program for linear systems, 71
Euler angles, 337
Euler approximation, 229
Euler's formula, 489
Euler method, 73–75
exogenous inputs, 84
explanatory variables, 180
extended Kalman filter (EKF), 132, 146, 151, 178, 222, 235–237, 241, 245, 299, 309–310, 321, 324, 329, 331, 344–349, 353, 372
 application, 518–520
 derivation, 511–520
 nonlinear systems, 519–520
extrapolation methods, 72
F-value, 202
fast Fourier transform (FFT), 39
Federal Aviation Administration (FAA), 26, 46
feedback gain matrix, 298
feedforward neural networks (FFNN), 266–270, 276–278
 application, 270
 architecture, 286
 convergence, 277
 examples, 281–289
 extraction of stability and control derivatives, 278–279
 generalization, 289
main advantage and strength, 279
outputs, 279
potentials, 290
prediction, 270
software, 279–280
stall hysteresis, 287
training, 270, 289
weights, 271, 277, 289
filter error method, 79, 131–176, 222, 306, 308–309, 316, 323, 328–329, 331
algorithm, 132, 138–145, 316
application to data with moderate to high level of turbulence, 167
application to data with process noise, 164–167
block schematic, 132–133
estimation of nondimensional derivatives, 171
examples, 164–174
extension to multiple experiments, 155–157
flight data in seemingly steady atmospheric conditions, 173
flow chart, 160
linear systems, 133–135, 161
nonlinear systems, 145–154, 162
software, 160–174
summary of computations for nonlinear systems, 150
filtering methods, 234–245, 348
finite-difference approximation, 93, 149
finite-element method (FEM), 7
first-order differential equation, 66
first-order lag, 64–66

- first-order linear approximation, 132
first-order state equations, 386
Fisher information matrix, 34, 81, 83,
 111, 502
fit error decomposition, 379–381
five-hole probe
 calibration, 358–362
 mounted on noseboom, 360–361
flight control laws, 327, 331
flight control systems, 390
flight controller, 295–296
flight-derived aerodynamic
 coefficients, 209
flight-derived aircraft parameters,
 past methods and techniques,
 11–12
flight-derived coefficients, 207,
 460–462
flight-estimated vortex model
 parameters, 454
flight flutter investigation, 46
flight maneuvers, 26–33, 207–208
flight mechanical parameters, 10
flight mechanics, 6, 8
flight path reconstruction (FPR),
 335–337, 339, 341
 comparison of EKF and ML
 methods, 349
 deterministic approach, 344–345
 stochastic approach, 345–350
 techniques, 344–350
flight testing, 26–33
 categories, 26
 instrumentation, 457
 performance evaluation, 27–29
 program organization, 46
 scope, 46–49, 55
 system identification, 29–33
 techniques, 13
flow angles, 338
 calibration, 358–363
 measurement, 362
flow-field characteristics,
 determination of, 452–455
flow separation point
 and quasi-steady lift
 coefficient, 416
 during dynamic stall, 418
flush air data system (FADS),
 362–363
forgetting factor, 226, 244, 276
forward difference approximation, 92
forward Kalman filter, 348
forward pass, 270–271
 schematic, 272
forward propagation, 271–272
forward selection procedure, 198
Fourier coefficients, 313
Fourier transform regression (FTR),
 221, 246–247, 252
 algorithm, 233
 method, 229–234
Fourier transformation, 37, 313
free flight including extended landing
 gear phase simulation, 464
frequency domain, 230, 299, 313–314
 comparison of standard inputs, 42
 methods, 35
 model validation in, 389–392, 474
frequency response magnitude plots,
 38, 44
frequency response matrix, 391
frequency response methods, 12
frequency sweep testing, 33, 46
fudge factor, 377
function cells, number of, 73
function evaluation, 94
gain matrix, 145, 223
Gauss–Markov process, 157–158, 351
Gauss–Newton method, 88–91,
 94–96, 98–100, 102–105,
 107, 112, 119, 121–122,
 138–149, 151, 154, 300,
 307, 312–313, 468, 476
Gaussian random variables, 134
Gaussian weighting function, 267
Gaussian white noise, 345–346
Gaussian window width, 227
Gear’s method, 75
general nonlinear function, 60–62, 190
general state variable, 351
generalized least squares (GLS)
 method, 189
geodetical positions, state
 equations for, 339
global model development, 48
goodness of fit, 376, 378–379
gray-box models, 6
ground effect, 29, 383, 431–433
 influence function, 428, 432
 modeling, 427–433
gust spectrum, explicit modeling,
 157–159

- Hamiltonian matrix, 139, 143
helicopters *see* rotorcraft
Hessian matrix, 88
Heun method, 73
HFB-320 aircraft, 167, 257
hidden layer, 269–271, 277
 activation function, 271, 273
 nodes, 278
higher-order centered formula, 69
hinge moment, 441–443
 database, 387
hyperbolic tangent function, 271–273
hypersonic flight vehicles, 55
- identifiability, 67, 493
identification *see* system identification
identity spectral density, 134
imput design by estimation error analysis, 33–36
IMU (inertial measuring unit), 49
incremental control inputs, 382
incremental model update procedure, 434–435
inequality coefficients, 380
inequality constraints, 100–101
inertial accelerations, 339
inertial navigation system (INS), 49–50
inflow dynamics, 474–477
information matrix, 91, 97, 99, 112, 151, 184, 300
 block structure, 114
initial conditions, 61, 71
 identifiability of, 493–495
 state variables, 387
initial covariance matrix, 236
initial noise covariance matrix, 154–155
initial-value problem, 72, 311
input–output behavior, 59
input–output characteristic, neuron, 268
input–output equation, 177
input–output matching, black-box
 approach, 288
instrument errors, 61–67, 221, 344–345, 352–353, 370
instrumental variable (IV) method, 194–196
 block schematic, 194–195
instrumentation and measurements, 49–55
integrated models, 387, 389, 447
integrating feedback, 299
- integration formulas, 73
integration interval, 73–74, 312
integration methods, 72–76
 aerodynamic derivatives, 74
interdisciplinary flight vehicle
 modeling, 6–11
interior-point algorithm, 100–103
internal system behavior, 59, 64
International Federation on Automatic Control (IFAC), 15
inverse simulation, 382–383
iterated Kalman filter, 237, 518
iteration index, 91, 103, 140, 151
iterative OEM program, 208
iterative procedure, 210
iterative update equation, 151
- Jacobian matrices, 347
Jacob's method, 106
Joint Aviation Authorities (JAA), 26
joint probability density function, 498
“Joseph” form, 517
- Kalman filter, 132, 134, 139–145, 178, 234, 243, 261, 275, 348
 algorithm, 244, 368, 370
 computations, 512
 derivation, 511–520
 gain matrix, 135
 linear systems, 518
 technique, 370
 see also specific types
 and applications
Kalman gain, 242, 283
Kalman gain matrix, 141, 154–156, 515–517
Kalman gain vectors, 275
Karush–Kuhn–Tucker conditions, 102
Kelly–Bryson gradient algorithm, 266
kinematic equations, 59–60, 336–344, 371–372
 motion variables, 339
kinematic relationships, 335
knowledge-based models, 5
Kronecker delta, 81
Kuhn–Tucker optimality conditions, 100
- lag effect, 64
Lagrange multipliers, 101
Lagrangian approach, 99, 101
Lamb–Oseen model, 453
landing flaps, 48

- landing gear effects, 383, 409–411, 462–465
landings and takeoffs, 28–29
lateral-directional derivatives, 30
 recursive estimates, 256
lateral-directional motion, 62, 119–122, 254–256, 326
 aerodynamics modeling, 281–283
 identification, 325–331
 observation equations, 120
 parameter estimates, 255, 258
 state equations, 119–120
lateral force derivatives, 330
least mean squares estimator (LMS), 180
least squares method, 177–179, 219, 331, 336
 and combined output error, 304
 applicability to state space
 models, 187
 block schematic, 179
 cost function, 230–231
 examples, 204–216
 overview, 178–187
 practical considerations, 185–187
 properties of, 182–185
 recursive, 222–234
 selected parameters, 213
 summary of variables for
 one-shot, 212
 unstable aircraft identification, 299–303
Levenberg–Marquardt method, 93, 103–105, 107, 112, 119, 121
lift coefficient during dynamic stall, 418
lift curve slope, 383
likelihood equation, 81
likelihood function, 80–82, 84–85
 derivation of, 497–499
 logarithm of, 80
 minimization, 88, 507–509
line search, 96
linear accelerations, 208, 339
linear combinations, aircraft responses, 298
linear constraints, 103
linear equations, scaled system, 105
linear filtering theory, 132
linear least squares method, 105, 177
linear models, 59, 67–71, 189
 aerodynamic parameters, 196, 207–216, 248–250
linear propagation, 238
linear regression analysis, 196
linear systems, 92, 245
 estimation program, 71
 filter error method, 133–135, 161
 integration, 75–76
linearity of expectation, 186
linearized Kalman filter, 518
local model network (LMN), 266–267
locally weighted regression method, 226–229
LOES (low-order equivalent system)
 mismatch, 390–392
longitudinal data with process noise, 283–285
longitudinal derivatives, 167–171
 recursive estimates, 253
longitudinal motion, 250–253
lumped bias parameters, 68, 71
Lyupanov equation, 143

Mach number, 46–48, 55
Mallow's *C_m* statistics, 202
maneuver combination to avoid scale
 factor and bias correlation, 343
mathematical models, 1, 9, 12, 84, 133
 extensions, 61–63
 general case, 60
 structure, 3
Matlab, 19, 20, 116, 126, 160, 174, 182
matrix exponential, 75
matrix-inversion, 91, 112, 182
matrix-inversion lemma, 223, 228
maximum inequality, 380
maximum likelihood (ML)
 estimates, 12, 69–70, 100, 137, 144, 299
 essentials of, 83
 parameters in dynamic systems, 84–86
 principle of, 79–82
 properties of, 83
 statistical properties, 501–506
maximum roll capability, 28
mean aerodynamic chord, 206
mean square error (MSE), 274
measured responses, 357–358
 Bode plot of, 391
measurement covariance matrix, 308
measurement equations, 342

- measurement errors, 61, 70, 81–82, 84, 90, 178, 387
measurement noise, 10, 61, 79, 84, 92, 134, 153, 186, 234, 237, 289, 306, 309, 331
covariance matrix, 135
known, 86
unknown, 86–87
distribution matrix, 61
measurement vector, 305
memory index, 244
minimization problem, 86
minimum sampling rate, 52
mixed continuous/discrete time formulation, 79
mixed estimation, basic principle, 302–303
model adequacy, 389
model building process, 3
model characterization, 5–6
model deficiencies, 376
model definition function, 118, 165, 247
model fidelity, 3, 386–389, 409, 430, 472
model identification procedure, schematic, 436
model plausibility, 383–386
model postulates, 59–78
model predictive capability, 375–376, 386–389, 392
model structure determination, 197–204
basic approaches, 198
choice of potential independent variables, 203
difficulties associated with, 203
practical aspects, 203–204
model validation, 375–394
aspects of, 375
definition, 375
frequency domain, 389–392
role of, 375
modeling errors, 10–11
modified Newton–Raphson method, 90
modular neural networks (MNN), concepts of, 287–288
moment coefficients, 190, 204–209
schematic of data preprocessing, 205
moment of inertia, 206
moment reference point, 207
momentum factor, 274
momentum term, 277–278
MSL (mean sea level), 365
multiaxis orthogonal phase-optimized sweeps, 55
multicollinearity measure, 300
multi-output regression, 189–190
multiple experiments, 155–157
multiple linear regression, 179
multiple shooting method, 296, 311–312, 316
basic concept, 311
multiplicative rule, 186
multistep input signal
1-1-2-3 input, 43
1-2-1 input, 44
3-2-1-1 input, 42–44
design, 36–46
power spectrum, 489–491
multistep methods, 72
multivariate multiple regression (MMR), 190–191, 199
Navier–Stokes flow solvers, 10
Nelder and Mead simplex method, 106
neural networks (NN), 267
basic building block, 268
performance, 269
processing basics, 268–270
techniques, 5
neuron, 268
input–output characteristic, 268
schematic, 268
Newton–Balakrishnan method, 90
Newton–Raphson method, 88, 90
Newtonian mechanics, 5, 59
nine degrees-of-freedom models, 472–473
node activation function, 272
noise covariance, 160
noise covariance matrix, 82, 100, 308
noise distribution matrix, 60–61, 134, 136
noise processes, 158, 239, 346
nondimensional longitudinal derivatives, 257–259
nonlinear activation functions, 278
nonlinear filtering, 235
nonlinear inequality constraints, 149
nonlinear least squares method, 177
nonlinear models, 70–71, 93, 219
aerodynamic parameters, 215–216
nonlinear multivariate multiple regression (MMR), 191

- nonlinear optimization, 87
nonlinear propagation, 238
nonlinear regression, 189–190
nonlinear response relation, 190
nonlinear state space model, recursive estimates, 259
nonlinear systems, 62, 84, 132, 160 equations, 341 extended Kalman filter (EKF), 519–520 filter error method, 145–154, 162 nonparametric models, 6 nonsystematic error, 381 normal equation, 182, 188 normalized index, 380 noseboom five-hole probes mounted on, 360–361 position error in static pressure measured, 368 numerical approximation initial-value problem, 72 system matrices, 69–71 numerical differentiation, 53–54 numerical divergence, 72 numerical simulation, 317 Nyquist frequency, 52
- observability, 67, 139, 493
observation, 180
observation equations, 164, 168, 191, 250, 305, 317, 344, 352
observation matrix, 135
observation variables, 308
Ockham's Razor, 1
off-CG location, 342
off-CG sensors, 340
one dimensional search, 96
open-loop identification, 298
open-loop parameter estimation, 295–296
open-loop simulation, 317
optical cuing, 43
optimal input design, 25–26, 33–46
optimal tuning parameters, 276–278
optimization algorithm, 87, 137 problem, 87, 99 procedure, 87–88, 105, 277, 313, 378 schematic, 89 theory, 97, 99
ordinary least squares (OLS), 177, 213–215 schematic, 192
orthogonal distance regression (ODR), 191
oscillatory modes, natural frequency and damping, 385
Oswald factor, 215
output error method (OEM), 79–129, 159–160, 166, 172, 207, 209, 256, 296, 304, 310, 321, 325, 328, 345, 372
algorithmic implementations, 112–115 application to data with moderate to high level of turbulence, 124–125 application to data with process noise, 122–126 artificial stabilization, 296, 310–311 block schematic, 85–86 estimation of nondimensional derivatives, 170 examples, 119–126 flight data in seemingly steady atmospheric conditions, 173 frequency domain, 313–314 implementation, 117 software, 115–119
- Padé approximation, 64
parameter error covariance matrix, 34
parameter estimation, 3–4, 68, 89, 98, 104, 314, 316, 326, 375 accuracy and reliability, 25, 49–50 error, 186 flight data, 33, 252 flight mechanics, 9–11 lateral-directional motion, 255, 258 methods, 335 modern methods, 12–14 standard deviations, 376 statistical accuracy, 111–112, 376–378 unstable aircraft identification, 296
parameter identification (PID), 4, 296, 314
parameter perturbation, 93
parameter vector, 61, 63, 80, 134, 137 update, 90, 140–145
parametric models, 6
parametric wake distortion (PWD), 476
partial correlation, 200, 202
partial *F*-values, 200

- penalty function methods, 99
period of oscillation, 41–42
perturbation
 equations, computational sequences, 114
 error-covariance matrix, 153
 gain matrix, 149
 output variables, 149
 parameter vector, 93, 149
 relative parameter, 93
 responses, 93, 153
 state equations, 93
 state prediction error, 149
 state variables, 149
 system equations, 93, 153
 system matrices, 149
phenomenological models, 5–6
Phoenix RLV demonstrator, 455–465
 final configuration and sensors, 457–459
 future applications, 465
phugoid mode, 30, 167–168
pilot inputs, 52, 298
pitch axis, alignment error, 340
pitch damping derivatives, separation of, 396–403
pitch rate
 and rate of change of angle of attack, 397–398
 response to DLC flap input, 405
pitching moment
 derivatives, 400
 due to canard, 315
 equation terms, 38
pitching motion, 51
 variables, 383
Pitt and Peters inflow dynamics, 474
plausibility of estimates, 376
position error in static pressure measured at noseboom, 368
postulated model *see* model postulates
Potter's method, 139–140
Powell's method, 106
power spectra, 51
 multistep input signal, 489–491
power spectral density, 158, 172
prediction step, 134, 235, 346, 518, 519
predictor variables, 180
preflight aerodynamic database (ADB), 460–462
principle of parsimony, 1
principle of simplicity, 1
probability density, 80
probability density function, 80, 497–499
process noise, 79, 122–126, 131, 156, 159, 164–167, 234, 306, 309, 331, 346
approaches, 132
combined natural cum innovation formulation, 138
distribution matrix, 61, 63, 72, 155, 165, 174
formulations, 135–137
innovation formulation, 136–137
longitudinal data, 283–285
natural formulation, 135–136
proof-of-match (POM), 386–387, 389
 computational procedure, 388
 critical engine failure on takeoff, 446–447
pseudo-control inputs, 69–71
 basic concept, 70
pseudo-parallel processing, 114–115
pseudo-state variables, 351
pulse inputs, energy spectra, 40
pushover–pullups, 27, 30
Quad-M basics, 13–14, 54, 485
quadratic programming, 143
quasi-linearization method, 90–91
quasi-steady lift coefficient and flow separation point, 416
quasi-steady stall
 modeling, 285–289, 418–427
 time histories, 422
radial basis function (RBF) neural network, 266
Rauch-Tung-Striebel smoother, 348
real-time onboard
 verification, 368–369
real-time simulation, 72
recurrent neural network (RNN), 266
recursive algorithms, comparative evaluation, 260–261
recursive estimates
 lateral-directional derivatives, 256
 longitudinal derivatives, 253
 nonlinear state space model, 259
recursive–iterative process, 271
recursive least squares (RLS) algorithm, 224–225, 246, 249
recursive maximum likelihood (RML) method, 222

- recursive parameter estimation (RPE),
219–264
algorithms, 256, 261
examples, 248–259
limitations, 220–221
methods, 220
- recursive weighted least squares,
225–226
- regions of identifiability, 38–39
- regression analysis, 108, 177
- regression equation, 184, 189, 192
- regression method, 299, 328
- regression startup procedure, 107–109,
296
- regressors, 180
- relative parameter perturbation, 93
- relaxation procedure, 87
- Research and Technology Organization
(RTO), 15
- residual analysis, 375, 378–382
- residuals, from simple linear regression,
188
- response curve fitting method, 12, 79
- response gradient, 87, 152
- response gradient matrix, 92
- response variable, 179–180
- retarded systems, 63–66
- reversible flight controls, 52, 387, 441
- Riccati equation, 139–140, 142,
147, 156
- rigid-body aerodynamics, 75, 441–445,
468, 470, 473
- roll maneuvers and elevator input, 401
- roll-spoiler hardover
identification, 414
simulation, 412
- roller-coaster, 27
- rolling motion, 51
equation, 470
- rotor wake modeling, 473–479
- rotorcraft
extended rigid-body and rotor model
structure, 471
flight at forward speed, measured and
model estimated, 476–478
identification, 33
improvement of simulation fidelity at
hover by parametric wake distor-
tion model, 477
modeling and simulation,
465–479
off-axis response at hover, 477
simulation models (SIM), 466
- rudder effectiveness, 173
- rudder input maneuvers, 31
- Runge–Kutta method, 72–75, 77,
236, 311
- s-plane transformation, 296
- S-WAKE project, 450
- scale factors, 343, 360
angle of attack and angle of
sideslip, 342
flight-estimated, 361
linear accelerations and angular rates,
341
- scaled system of linear equations, 105
- scaling parameters, 240
- Schroeder-phased multisine signal, 46
- SEADS (Shuttle Entry Air Data System),
362
- second order system, 386
- secondary aerodynamic effects, 383
- sensitivity coefficients, 91–94, 148, 153,
360–361
- sensitivity matrix, 140–141
- sensors
calibration, 52
dynamics, 7
errors, 10
models, 338, 341
- separate surface excitation (SSE),
314–315, 331
- serial processing, 114
- short period dynamics, 390
- sideforce due to sideslip, 330
- sideslip, sideforce due to, 330
- sideslip angle, 32–33, 328–329
- sideslip maneuvers, 28
- sigma point filters (SPF), 237
- sigma points, 242
- sigmoidal function, 269
- signal-to-noise ratio, 52
- similarity transformation, 143
- simple bounds, 99
- simplex method, 106
- simulated responses, 357–358
- simulated unstable aircraft response
data, 317–325
- simulated unstable short period
motion, 318
- simulation, 2, 71–76, 392
definition, 4, 71
- rotorcraft modeling approach,
466–473
see also specific applications

- simulator certification, 47
Simulink, 116
simultaneous perturbations, 94
singular value decomposition (SVD), 96–98, 192, 229, 303
six degrees-of-freedom models, 472–473
slope factor, 271–272
smoothing, 348, 353
software
 feedforward neural networks (FFNN), 279–280
 filter error method, 160–174
 implementation, 245–248
 output error method (OEM), 115–119
software packages, 19, 316
speed brake
 and flap configuration, 407
 effect modeling, 406–409
 influence of downwash lag cause by, 409
spring–mass–damper system, 385
stabilization matrix, 311, 324
stabilized output error method, 321
stall hysteresis
 feedforward neural networks (FFNN), 287
 time histories, 424
stall modeling
 Dornier 328 aircraft, 422–425
 VFW-614 ATTAS, 420–422
stand-alone models, 387, 389
standard deviations, 111, 376
standard output error method, 323
standard regression problem, 231
state derivative, 173
state equations, 145, 164, 168, 250, 254–256, 317, 339
state estimation, 72, 132, 309, 344–345, 350
 without knowledge of noise
 covariances, 243–245
state noise, 153
state prediction error, 135, 139, 152
state propagation error, 135, 247
state space models, 59–60, 162–163, 187, 338, 345, 466
state transition matrix, 75–76, 135
state variables, 3, 59, 153, 305, 387
state vector augmentation, 70–71, 243
static pressure field, 364
static pressure measurement, 50–51
static pressure ports, calibration, 363–368
statistical accuracy of parameter estimates, 376–378
statistical analysis of residuals, 376
statistical properties of estimates, 375
statistical test criteria, 201–203
steady flow separation point, approximation, 417
steady heading sideslip, 32
steady heading steady sideslip (SHSS), 32–33
steady-state filter, 146–149, 153–154, 308
steady-state gain matrix, 147
steepest descent method, 273
step size control, 95–98
 dominant directions, 96–98
 heuristic approach, 95–96
stepwise regression, 178, 199–200
stick force per g, 28
stiff systems, 75
stochastic equations, 146
strip method validation, 455–456
structural mechanics (SM), 6
structural modes, 8
subplex method, 106
sweep input maneuvers, 32
system identification, 3, 266, 375
 applications, 486
 art and science of, 13
 contributions to theory and applications, 14
 coordinated approach, 12–13
 definition, 3
 examples, 395–484
 flight testing, 29–33
 fundamental assumptions, 2
 future directions, 486–487
 goal-oriented task, 485
 maneuvers, 45
 modern era, 12
 need for, 8–9
 outcomes of discussions, 485–486
 overview, 2–5
 present state of maturity, 14
 process, 2–3
 rotorcraft modeling approach, 466–473
 technical definition, 3
 use of term, 1
system information matrix, 105
system matrices, numerical approximation, 69–71

- system parameters, 60, 62–63, 67, 160
system representation, 68
system response, 91–94
system theory, problems encountered, 2–3
systematic errors, 61–62, 185, 309, 335, 381
- t*-ratios, 198
takeoff, proof-of-match for critical engine failure on, 446–447
Taylor series methods, 10, 60, 72–73, 76, 88, 396, 502
Taylor's hypothesis, 396
Theil's inequality coefficient, 379–381
thrust deflection angle, 327
thrust vector deflection, 314, 326
time delays, 60, 63–66, 360
time domain methods, 35, 60, 79, 314
time domain system identification, 14
time histories, 51
 control inputs, 122–123
 independent variables, 211
 measured and estimated outputs, 212
 plots, 329
time-invariant parameters, 219
time-shift, 65–66
time-varying filters, 72, 151–154
time-vector method, 12
tolerances, 387, 391
total least squares (TLS) method, 178, 191–194, 213–215, 303
 schematic, 192
tower flyby maneuver data analysis, 365–368
tower flyby technique, 372
tower flyby tests, 364–365
trailing edge flow separation phenomenon, 415
training algorithms, 270–276
training simulators, databases, 433–447
Transall C-160 aircraft, 47, 172, 360–361, 364, 406–407, 419, 430, 437–441
 ground effect estimation, 431–433
 validation test for Dutch roll dynamics, 441
transformation techniques, 99
transit time, 64
 lag effects modeling, 396–409
translational accelerations, 205
- truncation error, 69
turbulence, 124–125, 132, 159, 167, 171–172, 186, 308, 387
two-point boundary value problem, 296
two-step nonlinear filter, 147–149
two-step relaxation strategy, 101
- UD-factorization, 245
universal gas law, 338
unscented Kalman filter (UKF), 222, 237–243, 245, 309–310, 331
unstable aircraft identification, 295–333
 basics, 297–299
 closed-loop, 295–296
 determination of aerodynamic characteristics, 295
 examples, 317–331
 least squares method, 299–303
 open-loop, 295–296
 programming considerations, 316
unstable systems, estimation method, 322
unsteady aerodynamics modeling, 414–418
- validation tests, 390
 sample, 388
VECTOR, 362
VFW-614 ATTAS aircraft, 360, 399, 403, 419, 429, 431–432
flight-identified total aerodynamic coefficients, 433
ground effect influence function, 432
stall modeling, 420–422
vortex model identification and validation, 451
- wake distortion parameters, 476
wake model parameter estimation, 452
wake propagation in pure hover and pitching motion, 474–475
wake velocity components during lateral encounter, 454
wake vortex aircraft encounter model, 450–455
weathercock stability, 172–173
weight-update rule, 273
weighted least squares (WLS) method, 79, 177–178, 188–189, 194
weighting factor, 110

- weighting functions, 226
weighting matrix, 90, 227
white-box models, 6
white zero-mean Gaussian noise, 352
whiteness test, 381–382
wind-box maneuver
 flight measured and reconstructed time histories, 371
 flight track, 369
 technique, 368–371
wind-tunnel techniques, 10
wind-up turns, 28
wings-level steady sideslip, 32–33

X-31A, 325, 425, 435
X-31A model validation and update, 435, 448–450

zero shifts, 61–62

This page intentionally left blank