WOODHEAD PUBLISHING IN MECHANICAL ENGINEERING

Modelling and simulation of integrated systems in engineering

Issues of methodology, quality, testing and application

David Murray-Smith

WP WOODHEAD PUBLISHING

Modelling and simulation of integrated systems in engineering

Modelling and simulation of integrated systems in engineering

Issues of methodology, quality, testing and application

DAVID J. MURRAY-SMITH

WP
WOODHEAD PUBLISHING

Oxford Cambridge Philadelphia New Delhi

Published by Woodhead Publishing Limited, 80 High Street, Sawston,
Cambridge CB22 3HJ, UK
www.woodheadpublishing.com
www.woodheadpublishingonline.com

Woodhead Publishing, 1518 Walnut Street, Suite 1100, Philadelphia, PA 19102–3406, USA

Woodhead Publishing India Private Limited, G-2, Vardaan House, 7/28 Ansari Road,
Daryaganj, New Delhi – 110002, India
www.woodheadpublishingindia.com

First published in 2012, Woodhead Publishing Limited

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library.

Library of Congress Control Number 2012933912

Woodhead Publishing ISBN 978-0-85709-078-2 (print)
ISBN 978-0-85709-605-0 (online)

Typeset by RefineCatch Limited, Bungay, Suffolk
Printed in the UK and USA.

Contents

List of figures

List of tables

List of abbreviations

BDF	backward difference formula
CAD	computer aided design
CAMP-G	Computer Aided Modelling Program with Graphical Input
CSCWD	Conference on Computer Supported Cooperative Work in Design
CSP	communicating sequential processes
DAE	differential algebraic equation
DE	differential evolution
DEVS	Discrete-Event System Specification
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DSP	digital signal processor
ESA	European Space Agency
ESL	European Simulation Language
FPGA	field programmable gate array
GA	genetic algorithm
GP	genetic programming
IEEE	Institute of Electrical and Electronics Engineers
INRIA	Institut Nationale de Récherche en Informatique et en Automatique
MATLAB®	MATrix LABoratory program
MD	molecular dynamics
MEMS	micro-electromechanical systems
MIMO	multi-input multi-output
MIT	Massachusetts Institute of Technology
NEMS	nano-electromechanical systems

NM	Nelder-Mead
NPS	US Naval Postgraduate School
NR	Newton-Raphson
NSHEB	North of Scotland Hydro-Electric Board
ODEs	ordinary differential equations
ONR	US Office of Naval Research
PI	proportional plus integral
PTB	Project Test Bed
PWM	pulse-width modulated
QSS	quantised state system
RAE	Royal Academy of Engineering
RCM	resistive companion method
SA	simulated annealing
SCS	The Society for Computer Simulation
SISO	single-input single-output
SSA	segmented simulated annealing
TIC	Theil's Inequality Coefficient
UML	Unified Modelling Language
UUV	unmanned underwater vehicle
V&V	verification and validation
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VTB	Virtual Test Bed
VV&A	verification, validation and accreditation

Acknowledgements

Copyright acknowledgements

The following organisations are gratefully acknowledged for granting permission to use previously published material:

North Atlantic Treaty Organisation (NATO), Research and Technology Organisation (RTO) for permission to reproduce diagrams which are included in this volume as Figures 6.1, 6.2, 6.3 and 7.2. The original version of this material was published by the Advisory Group for Aerospace Research and Development, North Atlantic Treaty Organisation (AGARD/NATO) in *AGARD Advisory Report 280* in September 1991.

SAGE Publications for permission to reproduce diagrams previously published in the *Transactions of the Institute of Measurement and Control* (Figures 7.3, 7.4 and 7.6).

Elsevier Ltd for permission to reproduce diagrams previously published in *Control Engineering Practice* and *Automatica* (Figures 6.8, 6.9 and 7.40).

Product and trademark acknowledgements

acslX® is a product of AEgis Technologies Group Inc, USA.
Autodesk® Inventor® is a product of Autodesk Inc, USA.
CAMP-G is a product of Cadsim Engineering, USA.

SolidWorks® is a product of Dassault Systèmes SolidWorks Corp, France.
Stateflow® is a product of MathWorks Inc, USA.
System Identification Toolbox is a product of MathWorks Inc, USA.
Virtual Test Bed (VTB) is a product developed at the University of South Carolina, USA.
Visual Basic® is a product of the Microsoft Corp, USA.
Windows® is a product of the Microsoft Corp, USA.
20-Sim is a product of ControlLab Products BV, Enschede, the Netherlands.

Preface

My interests in modelling and simulation began during my period as an undergraduate and Master's degree student at the University of Aberdeen, and developed further in industry at Ferranti Ltd (Edinburgh) where I worked on electronic and mechanical elements of inertial navigation and related aircraft systems. This was my first experience of working on highly integrated and multidisciplinary systems, although terms such as those were not in common usage then. For such design applications, issues of model accuracy were of critical importance and simulation techniques also had a central role. Interests in the application of modelling and simulation continued throughout my period of PhD study at the University of Glasgow and were further extended as a result of many years of teaching control engineering topics within the Department of Electronics and Electrical Engineering in that same university. My research since the 1970s has involved modelling and control systems applications in a variety of different areas, including biomedical engineering, electrical power generation, system identification applied to helicopter flight mechanics modelling and flight control, ship control systems and other applications involving underwater vehicles.

Topics such as experimental modelling methods, issues of model quality and model testing, parameter sensitivity analysis and inverse simulation are treated in some detail within this book, as they have proved very important to me in much of the work that I have been engaged in over the years. This, I believe, forms an important and timely contribution in the context of the complex problems that can arise in design applications involving integrated systems. Issues of model management, such as documentation and the development of model libraries and generic descriptions, are also emphasised. Although mentioned as examples of what is available, specific simulation tools are not given special emphasis in the book, as information of that kind inevitably becomes out of date very quickly. The emphasis is on principles of modelling and simulation and methods of approach that are, in my opinion, especially relevant to the problems of integrated systems. I hope that others will find some of the content useful in the context of their own work.

It is impossible to mention everyone who, over a period of many years, has had a direct influence on the ideas and material presented in this book but I owe many people my deepest thanks. I am especially grateful to Dr Euan McGookin (University of Glasgow) for collaborative work involving ship and underwater applications and to Professor Gareth Padfield (now of the University of Liverpool), with whom I collaborated closely on helicopter flight mechanics and flight control applications over a long period of time while he was at the Royal Aircraft Establishment (Bedford) and later the UK Defence Evaluation and Research Agency. Others who must also be mentioned include Dr Douglas Thomson (University of Glasgow), who has been responsible for my interest in inverse simulation techniques and Professor Roy Crosbie (California State University, Chico) for valuable discussions, especially on real-time applications. Many others have influenced the work and, in particular, I must mention that I have benefitted from useful general discussions about modelling principles, methods and applications with Professor Peter Gawthrop (University of Glasgow), Dr Moira

Smith (Waterfall Solutions Ltd), Professor Jay Rosenberg (University of Glasgow) and Mr Terry Ericsen (US Office of Naval Research).

I must also acknowledge the assistance provided by Professor George Moore (University of Southern California) and Professor Felix Breitenecker (Technical University of Vienna) in the arrangement of sabbatical visits to their universities. Through these periods in other universities I was provided with important new opportunities that were of long term benefit to me in terms of my research.

Many of my students have undoubtedly contributed to the work presented here and their contributions are reflected individually through references to their published work. I have also had very helpful feedback from experienced engineers during postgraduate short courses that I have presented in the UK, other parts of Europe, the USA, China and Brazil and at various pre-conference tutorial sessions that I have been involved in from time to time.

David Murray-Smith, September 2011.
Emeritus Professor,
School of Engineering,
Rankine Building,
The University of Glasgow,
Glasgow G12 8LT.

About the author

Professor David Murray-Smith was educated at Aberdeen Grammar School and at the University of Aberdeen where he graduated BSc(Eng) with First Class Honours in Electrical Engineering in 1963. He continued his studies at Aberdeen University and gained the degree of MSc by research towards the end of 1964. He was then employed at Ferranti Ltd (Edinburgh), working within a group engaged on the design and development of aircraft inertial navigation systems. After a couple of years in Edinburgh, he moved to the University of Glasgow for a three-year appointment as an Assistant, a position that also allowed him to undertake a PhD degree, which he gained in 1970. Subsequent appointments in the Department of Electronics and Electrical Engineering at the University of Glasgow over the period 1968 to 2005 were as Lecturer, Senior Lecturer, Reader and then as Professor of Engineering Systems and Control. He was appointed as Head of the Engineering Planning Unit and Dean of the Faculty of Engineering for the period 1997–2001. Sabbatical periods were spent at the University of Southern California and at the Technische Universität Wien.

Professor Murray-Smith's teaching and research has been concerned mainly with system modelling, system identification, simulation methods and engineering control systems. The emphasis in research has been mainly on interdisciplinary applications, both in the biomedical field and in aeronautics. Much of the work has involved close collaboration with colleagues in other departments at the

University of Glasgow, in other UK universities and UK research establishments, and also in universities and research institutes in other parts of Europe, North America, South America and Asia. The main areas of application have involved respiratory medicine, neurophysiology, electrical power generation and control of power systems, helicopter system identification, helicopter flight control systems, and control applications involving ships and underwater vehicles. Inverse simulation and model validation methods have been areas of particular interest in recent years.

External support for his research came mainly from the UK Engineering and Physical Sciences Research Council (EPSRC), the UK Biotechnology and Biological Sciences Research Council (BBSRC), the Wellcome Trust, the US Office of Naval Research (ONR) and last, but not least, the UK Ministry of Defence (mainly involving the Royal Aircraft Establishment (Bedford) – subsequently the Royal Aerospace Establishment (Bedford) and then the Defence Evaluation and Research Agency (Bedford)). Some industrially funded research has also been undertaken from time to time.

Professor Murray-Smith's present position at the University of Glasgow is as Emeritus Professor and Honorary Senior Research Fellow. He also holds an honorary position as Adjunct Research Professor in the Department of Electrical and Computer Engineering at California State University, Chico.

Professor Murray-Smith can be contacted at
David.Murray–Smith@glasgow.ac.uk

1

The principles of system modelling

Abstract: This chapter provides an introductory review of processes involved in developing models for engineering applications. Current trends in the design of integrated, multidisciplinary and embedded systems, and the associated consequences for modelling and simulation, are emphasised. Ways of classifying models are presented and the experimental approach to model development through system identification and optimisation is introduced. Issues of model quality are emphasised, together with model reuse, the development of libraries of sub-models, the potential of generic descriptions and use of modelling within procurement. These topics are all considered in more detail in later chapters.

Key words: modelling objectives, classes of model, quality, reuse.

1.1 General issues in the development and application of models

For scientific applications, the purpose of a model is usually to explain a complex set of behaviours or to help in the design of experiments as part of the process of hypothesis testing. In such fields, modelling is a central element of the scientific method. Similarly, in some engineering applications a model may be used to describe, analyse or explain the behaviour of a highly complex system, but it is more common to find models being used to support design, to assist in decision-making processes in the context of a specific application, or as a basis for simulators to be used in training or in further engineering developments. A properly tested and well-proven model can reduce development times and costs for many applications.

Integrated systems arise in many application areas including fly-by-wire aircraft, ships, land-based vehicles, energy conversion systems including electrical power generation and distribution systems, chemical plants and even in some household appliances. They typically involve a number of different aspects, disciplines or 'domains' (e.g. mechanical, electrical, electronic, control and software) which, ideally, are considered concurrently. For some applications, special forms of integrated system involve digital processors and software in addition to other physical hardware. The fields of aeronautical engineering, automatic control, road and rail vehicle engineering, marine engineering and robotics can provide many examples of such embedded systems.

The importance of integrated systems has been emphasised by the publication by the UK Royal Academy of Engineering (RAE) of a guide entitled 'Creating Systems that Work – Principles of Engineering Systems for the 21st Century' [1]. In the press release marking the publication of this guide in 2007, it is stated that the '. . . aim is to demystify the design of large integrated systems, and to give educators, students and practitioners alike six guiding principles that will help them to understand how large projects can be better conceived, designed and delivered'. These six principles can be summarised as:

1. debate, define, revise and pursue the purpose;
2. think holistic;
3. follow a disciplined procedure;
4. be creative;
5. take account of the people; and
6. manage the project and the relationships.

All sections of the RAE report make direct or indirect reference to the importance of appropriate tools for analysis, design and optimisation and the section dealing with the fourth of these principles ('be creative') puts special emphasis on modelling and simulation tools and methods. Simulation tools are vital for systems engineers in tackling the trade-offs within the design process, starting from the basic requirements in terms of performance, cost and timescale.

Models can have many benefits in the integrated systems approach to system design, including early assessment of performance, both within and beyond the normal operating envelope. Understanding of parameter inter-dependencies and knowledge of key sensitivities can also be very valuable for design optimisation. The use of simulation models is particularly important and leads to the concept of a virtual prototype which is a software-based implementation of the design, developed prior to any hardware prototype.

The success of virtual prototyping depends on the model quality. A successful model usually results from trade-offs involving several aspects of model performance such as the trade-off between the level of detail included in a model and the speed of solution in the corresponding computer simulation.

The level of detail is linked to model performance and as models are made more detailed, they inevitably become more complex. However, model complexity should never be confused with model quality and a simple description can often be better, in terms of quality measures, than a more complex one. It is also important to separate the processes of modelling from simulation. The development of a computer simulation is one common outcome of a modelling exercise but there are other potential applications for a model, some involving analysis carried out independently of any computer.

Whatever the use made of a model, it is important that its development should build upon previous experience. Attention must be given both to tools available for the development of computer-based modelling and simulation programs, and also to support systems for model management. Reuse of model components is important and some commercial modelling and simulation systems provide libraries of reusable models. Model management is also very important for applications involving large teams of developers, especially when these include multidisciplinary groups and geographically distributed teams. New developments in cloud computing are likely to have a significant influence on how simulation models are used within many organisations in future, but this is an area in which rapid changes are taking place and it is not possible, at the time of writing, to make more detailed predictions.

Developing a model requires careful examination of information about the real system. From this, inconsistencies or gaps in the available knowledge can be found which may result in further testing of the real system or a prototype, or some reconsideration of requirements. Donald Rumsfeld's much-quoted statement, made during a US Department of Defense news briefing in February 2002 [2], has direct relevance to issues of model quality and uncertainties:

> . . . as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns – the ones we don't know we don't know.

His statement was much ridiculed at the time but those words certainly apply to the processes of developing models. The 'unknown unknowns' in modelling are vitally important and have to be exposed by whatever means possible, including experimentation and testing.

Since a model is only an abstraction of the system it represents, perfect accuracy is impossible. The key issue is one of determining the model quality levels needed for the application in question. This implies reducing errors to defined levels for specified regions of the operating envelope of the system and balancing appropriate measures of accuracy against other measures of performance, such as solution speed.

In applications involving design, it is usual to base the structure of models on prior physical, chemical or biological knowledge. However, some sub-models may be based purely on input–output descriptions derived from tests on the corresponding elements of the real system (i.e. 'black box' models). Models thus range from completely 'transparent' descriptions, based on the application of recognised and accepted scientific or engineering principles, to purely empirical 'black box' forms, which are opaque. Between these extremes there is an important group, sometimes referred to as 'grey box' models, involving some empirical information found experimentally but with the structure of the model based on well-established physical laws and principles.

In summary, therefore, it can be said that mathematical modelling is an important tool for decision making and for engineering design. If models are developed correctly, they can then be applied over the range of conditions for which the descriptions are held to be accurate representations of the real system. An extensive programme of system testing and the creation of a solid base of experimental data are important steps in establishing how a given model may be used. When used within the concurrent approach to design, well-proven and tested models can lead to useful virtual prototype systems based on simulation software or to hardware-in-the-loop simulation involving a combination of simulation models and real system hardware.

1.2 Classes of model for engineering applications

Models used in science and engineering often involve variables that are continuous functions of time, such as position, velocity, acceleration, temperature or pressure. These are *continuous-variable models* and involve ordinary or partial differential equations or differential-algebraic equations. This is the main class of model considered in this book.

A second type of model that can be important in engineering involves what are known as *discrete-event* descriptions. In discrete-event models, all the variables remain constant between events that mark changes in the model. These changes take place at discrete time instants, either periodically or in a random fashion. A digital processor or computer used for real-time control is a good engineering example of a discrete system involving periodic changes. In this case, a continuous variable is sampled periodically through an analogue-to-digital converter and calculations are carried out using the discrete values obtained from the converter. Output from the processor may be converted back to continuous variable form using a digital-to-analogue converter. In modelling this type of component within some larger engineering system, we cannot use differential equations because of the discrete nature of the events within the processor and the associated converters and a difference-equation based approach is necessary.

Problems in which events occur in a more random fashion, as in road traffic flow or communications network modelling, lead to another approach known as *discrete-event simulation*. This is an important area, especially in engineering manufacturing and production, but this book is more concerned with modelling and simulation of continuous systems. Hybrid systems involving representations that are mainly continuous but do involve some discrete-event elements are discussed. Further details of discrete-event modelling and simulation techniques and their applications may be found in texts that deal with this area (see e.g. [3], [4]).

1.2.1 Conventional continuous-variable models

Within the class of continuous variable models we can distinguish between *models of data* and *physically based models of systems*.

A model of data involves a description fitted to measured responses, usually from a real physical system, leading to a model that expresses an observed relationship between two or more variables. It consists of mathematical functions that may have no direct link to recognisable elements of the real system. Such models are important in fields such as control engineering where input-output descriptions such as transfer functions or artificial neural net models may be used and can be derived from measurements. They can provide a starting point for design but give little information about internal processes. If such models are derived exclusively from experiments, their validity is restricted to the conditions for those experiments.

Physically based models, on the other hand, are developed using established scientific principles, such as the laws of physics and chemistry. Hypotheses about the structure and function of the system are used and appropriate simplifying assumptions are applied. Such models are more explanatory than the experimentally derived 'black box' descriptions within models of data. They include all relevant knowledge about the structure and parameters of the system, and involve internal variables that have measurable counterparts within the real hardware. The models and sub-models being considered in this book thus range from completely transparent descriptions based on physical principles, through intermediate 'grey-box' descriptions, to the entirely empirical black-box form of experimentally derived model.

Another important distinction is between *linear* and *nonlinear* models. Linear models are attractive because they are open to analysis and can be incorporated conveniently into design procedures. However, linear descriptions may be incapable of capturing aspects of the behaviour of the real physical system and issues of nonlinearity should be considered at an early stage in modelling. Assumptions of linearity should not be made without justification and the range of linear operation of the system always needs to be evaluated when a linear description is used.

As with questions of linearity, *time invariance* needs to be proved rather than assumed. A time-invariant description is one in which the performance of the system being modelled is independent of the times at which observations are made.

Models that are linear and time invariant receive particular attention in introductory engineering textbooks on topics such as electrical circuit theory, automatic control, dynamics and signal processing. Many systems have properties that allow them to be described by linear time-invariant models for some operating conditions and are attractive because they can be analysed using simple linear methods. Although nonlinear and time-varying dynamic models are more general, they are harder to deal with using classical mathematical methods, and numerical and computer simulation techniques are therefore very important for such cases. Simulation can provide solutions for cases for which no analytical approach can be used and this approach thus offers valuable insight for problems that would otherwise be intractable.

1.2.2 Inverse models and inverse simulation techniques

Although dynamic models or computer simulations are conventionally used to predict an 'output' time response from an 'input' time history, it can be very helpful to make use of a model in the opposite direction so that the user can specify the desired behaviour and the model or simulation provides an inverse solution. An example of an engineering design situation where this is important involves the design of actuators, where the inverse solution can reduce the need for an iterative approach. In such a case the *inverse model* provides information about the time history of the input needed to obtain that desired system output in a very straightforward fashion. Inverse simulation techniques of this kind are discussed in Chapter 4.

1.3 Questions of model quality

As pointed out in Section 1.1, the quality of a model has a direct influence on any design or strategy based on that model. Although models may be developed using physical, chemical or biological principles in the initial stages, the use of experimentation can be important for estimating appropriate sets of parameters if these are not known *a priori*.

This experimental approach to modelling is also of central importance in establishing the suitability (or otherwise) of a given model for an intended application (the so-called ‘model validation’ problem) and discussion of this topic (in Chapter 7) forms an important part of this book. Use of the word ‘validation’ may give a false impression of model capabilities and terms such as ‘model testing’ or ‘model evaluation’ may be more appropriate. Theories can be proved to be wrong but cannot ever be proved to be right, and there is always a risk of false confidence in model-based predictions if the model involved has been subjected to some form of ‘validation’. Models that can be shown to provide accurate predictions of reality in some circumstances cannot be assumed to be capable of giving good predictions in all cases. The ‘unknown unknowns’ mean that there can never be a simple conclusion in the processes that we conventionally call ‘model validation’.

Model testing, verification and validation can also be regarded as part of the process of defining boundaries within which a model and the related computer simulations should operate. As has been pointed out by Sargent (e.g. [5]), Balci (e.g. [6], [7] and [8]), Ören ([9]), Brade ([10] and [11]) and many others, validation is an integral part of the iterative process of model building. If testing is applied appropriately at each stage, confidence in the model should increase steadily during the model development process.

Early in an engineering design project, simplified models allow 'what if' situations to be examined and permit design trade-offs. At that stage little formal model validation is possible and the error bounds on model predictions can be large. Assessment of quality and fitness for purpose is thus likely to be based on previous design experience and on comparisons with models of earlier systems. As the work progresses, more detailed models may be integrated into the design process and more data should become available for testing of the model. Such testing is likely to be based on components to start with, then data resulting from tests on larger blocks and, much later on, data from the testing of complete prototype systems.

Thus, as test results become available, information begins to flow from the real system to the model. This contrasts with the start of the design process where the flow is almost entirely from the model to the system being designed. Bidirectional information transfer is a highly desirable characteristic of the later stages of the design process and ensures that the model is updated as knowledge about the real system is accumulated.

An example of this type of iterative model development may be found in a recently published account [12] of work on the design of a pumped heat electricity storage system by engineers employed at Isentropic, a company which is based in Cambridge in the UK. The concept, which is being developed for renewable energy schemes, uses a reversible heat engine to pump heat between two storage vessels containing a mineral in particulate form. Gas circulates through the system and is compressed to store energy, thus raising its temperature before passing through one of the vessels and heating the particles. This process results in cooling of the gas which then expands, producing further cooling. The gas is next passed to the second storage vessel where the mineral is cooled and the gas is brought back to a temperature close to the original value. The energy for this charging process comes from an electrical source. The discharge, which is the reverse of the charging process outlined above, releases energy which drives an electrical generator. High efficiency, of the order of 72 per cent to 85 per cent overall, is achieved by innovative design which minimises losses. These efficiency values are competitive with other methods for large-scale energy storage, such as pumped-hydro schemes. To prove the concept and validate performance predictions, engineers at Isentropic have designed small-scale demonstrator systems and this has involved an iterative process of modelling, simulation and testing. In this particular case, test results from the first prototype were not quite what were hoped for, so the model had to be refined and a second prototype built. Further testing led to further improvements in the model and a third prototype system [12].

The example outlined above is typical of modelling and simulation for innovative design situations where there are no systems of the same kind in existence. The system is an integrated one and the model, in this particular case, involves a mix of mechanics, thermodynamics and electro-mechanical energy conversion. Through careful application of modelling and simulation techniques and the cautious application of experimentally validated models, design engineers can identify strong and weak points of a proposal and refine the design in a stepwise fashion to an eventual successful realisation.

1.4 Methods of experimental modelling

System identification, which is the term used to describe experimental modelling, is generally considered to be a mature field and classical methods of identification involve linear discrete-time models within a stochastic framework. The aspects of identification, parameter estimation and optimisation techniques discussed in this book (mainly in Chapter 6) relate to the development of models for engineering applications. Objectives include real-time simulator development as well as models for engineering system design.

In some forms of model there can be a direct physical interpretation of the structure and parameters, with important benefits. Together with issues involving experimental design and the choice of test signals for the estimation of parameters, the selection of the model structure can contribute in an important way to the overall robustness of models that are established experimentally. This aspect of modelling and related issues of structural and parameter sensitivity and identifiability receive attention within Chapter 6. The process of extracting data from system and sub-system tests is not a trivial task, and the whole iterative process of development in the presence of uncertainties raises many important issues and emphasises the fact that there are no generally accepted standard approaches to model validation.

The use of *optimisation* techniques within the model development process has much in common with the use of optimisation in design and it is therefore helpful to apply experience gained in design applications for modelling situations. Although gradient-based optimisation methods remain important, the complexity of many practical problems means that it is impossible to establish a global optimum using gradient methods alone and more general techniques such as simulated annealing, genetic algorithms and genetic programming can provide benefits. These global optimisation tools are likely to become even more important as large integrated systems become more commonplace. The relevant methodologies are reviewed in Chapter 6.

Another important topic, closely related to optimisation, is *parameter sensitivity analysis*, which was the subject of much research in Eastern Europe in the 1960s and 1970s but has been rather neglected elsewhere. It has been found that insight gained from parameter sensitivity information can be of considerable value in the development and refinement of system models through investigation of model robustness and the design of appropriate test inputs. This topic is considered in detail in Chapter 5.

1.5 Model reuse and generic models

In industrial applications of modelling and simulation there is much interest in *modularisation* and *component reuse*, as these are key productivity factors in software development. In both industry and the academic world, until recently at least, simulation models were often started from scratch for each new project. This is clearly time-consuming and wasteful, and recent advances in object-oriented design and programming methods allow for repositories of reusable objects that can help to reduce the problems associated with the generation of new simulation models for new objectives. Modularity and component reuse are therefore concepts that are important in modelling. These ideas mean that, if we wish to build a new model for new given objectives, we can select established and proven sub-models from a model base to serve as elements of the new model [13].

Another significant development in recent years has been the development of a more *generic* approach to modelling in several industrial areas, including power electronic systems (e.g. [14] and [15]) and gas turbine systems (e.g. [16]). Here the word ‘generic’ is taken to mean ‘general’ or ‘not specific’ and implies adoption of a standard structure and standard building blocks within a model. These ideas are likely to become more widely used in future and this topic is explored in greater detail in Chapter 9, along with ideas relating to modularisation and libraries of sub-models.

1.6 Modelling within the procurement process

The ideas of ‘verification, validation and accreditation’ methodologies (VV&A), ‘smart procurement’ methods and the concept of ‘the model as a specification’ are currently being emphasised in the defence procurement area on both sides of the Atlantic. Requirements ideally emerge from an iterative process which involves all the stakeholders coming together to state what is wanted and design engineers then assessing possible ways of doing this using the available technology and the broader implications of different approaches in terms of lifetime costs.

Books and technical reports on modelling and simulation applied to very large and complex systems (e.g. [13]) are appearing in ever-increasing numbers from government laboratories and agencies due – in part at least – to concerns about excessive cost and time overruns in major projects. There appears to be a growing understanding that, in many cases, project failures can be traced back to failure to use modelling and simulation in appropriate ways at relevant phases of system development. Such interest in model testing and model quality for the design and development of very large and complex systems is to be welcomed but, even in cases of relatively simple models, there are many aspects of model validation, model optimisation and model tuning that require very careful consideration. Inadequate attention to model quality at an early stage, however simple the application, can lead to inappropriate design decisions that are difficult and probably expensive to correct at a later stage. The importance of these issues is not confined to large projects and careful consideration of simulation model quality issues can pay dividends whatever the application. In general, the more integrated the system being considered, the greater these benefits are likely to be.

1.7 References

[1] Elliott, C. and Deasley, P. (eds) (2007) *Creating Systems that Work: Principles of Engineering Systems for the 21st Century*. The Royal Academy of Engineering, London, UK.

[2] US Department of Defense, News Transcript, 12 February 2002 (online): *www.defense.gov/transcripts/transcript.aspx?transcriptid=2636* (accessed 13 August 2011).

[3] Banks, J., Carson II, J.S. and Nelson, B.L. (1996) *Discrete-Event System Simulation*, Second Edition, Prentice Hall, Upper Saddle River NJ, USA.

[4] Zeigler, B.P., Praehofer, H. and Lim, T.G. (2000) *Theory of Modeling and Simulation*, Second Edition, Academic Press, San Diego CA, USA.

[5] Sargent, R.G. (1979) 'Validation of simulation models', in Highland, H.J. (ed.), *Proceedings of the 1979 Winter Simulation Conference*, Vol. 2, pp. 497–503, IEEE Press, Piscataway NJ, USA.

[6] Balci, O. (1997) 'Principles of simulation model validation', *Transactions of the Society for Computer Simulation International*, Vol. 14, No. 1, pp. 3–12.

[7] Balci, O. (1997) 'Verification, validation and accreditation of simulation models', in Andradóttir, S., Healy, K.J., Withers, D.H. and Nelson, B.L. (eds), *Proceedings of the 1997 Winter Simulation Conference*, pp. 135–47, IEEE Computer Society, Washington DC, USA.

[8] Balci, O. (2004) 'Quality assessment, verification and validation of modeling and simulation applications', in Ingailis, R.G., Rossetti, M.D., Smith, J.S. and Peters, B.A. (eds), *Proceedings of the 2004 Winter Simulation Conference*, pp. 122–9, Winter Simulation Conference, USA.

[9] Ören, T.I. (1981) 'Concepts and criteria to assess acceptability of simulation studies: a frame of reference', in Adam, N., *Simulation Modeling and Statistical Computing* (guest editor), Communications of the ACM, Vol. 24, No. 4, pp. 180–8.

[10] Brade, D. (2000) 'Enhancing modeling and simulation accreditation by structuring verification and validation results', in Joines, J.A., Barton, R.R., Kang, K. and Fishwick, P.A. (eds), *Proceedings of the 2000 Winter Simulation Conference*, pp. 840–8, Society for Modelling and Computer Simulation International, La Jolla CA, USA.

[11] Brade, D. (2003) *A Generalized Process for the Verification and Validation of Models and Simulation Results*, Dissertation submitted for award of the degree Dr rer. nat., Fakultät für Informatik, Universität der Bundeswehr München, Germany.

[12] Davis, S. (2010) 'Saving for a windless day', *Engineering and Technology*, Vol. 5, No. 9, pp. 44–5.

[13] Cloud, D.J and Rainey, L.B. (eds) (1998) *Applied Modeling and Simulation*, AIAA, USA.

[14] Ericsen, T. (2000) 'Power electronic building blocks – a systematic approach to power electronics', in

Proceedings IEEE Power Engineering Summer Meeting, Vol. 2, pp. 1216–18.

[15] Ericsen, T.S. (2005) 'Physics based design, the future of modeling and simulation', *Acta Polytechnica*, Vol. 45, No. 4, pp. 59–64.

[16] Visser, W.P.J. and Broomhead, M.J. (2000) *GSP: A Generic Object-oriented Gas Turbine Simulation Environment*, National Aerospace Laboratory, NLR, Flight Division, Technical Report NLR-TP-2000-267, Amsterdam, the Netherlands.

2

Integrated systems and their significance for system modelling

Abstract: As engineering systems become more complex and include major elements of software as well as hardware, the approach to design has changed. Sequential design methods are being replaced by concurrent design processes. This has been apparent for some time in engineering applications where 'control-configured' solutions involving system integration, embedded controllers and multidisciplinary design issues are important, such as in the aircraft industry. Similar developments can also be found in other fields, such as automotive engineering and robotics. The introduction of integrated systems and the multidisciplinary processes of concurrent design have important implications for system modelling and simulation. Design processes of this kind are usually strongly model-dependent and involve optimisation at a system level. Models used as a basis for design must be of proven quality as model limitations and errors have direct implications in terms of the performance of the resulting system.

Key words: multidisciplinary, complexity, system integration, concurrent design.

2.1 An introduction to integrated systems

The widespread introduction of embedded systems and other forms of computer-based control in recent years has led to a rapid increase in the complexity of engineering systems. For example, digital 'fly-by-wire' control systems are now commonplace, both in civil and military aircraft, and this leads to new levels of interaction within the on-board systems of the aircraft, between the pilot and the vehicle, and between different vehicles. Multidisciplinary issues involving the elastic airframe, the flight control system, the propulsive control system and physiological 'biodynamic' factors involving the pilot are combining to an extent not previously encountered. For example, low-frequency modes of structural vibration may necessitate use of active structural mode control systems that are fully integrated with the primary flight control system, since the frequency ranges of these two systems are likely to overlap. Novel design features for aircraft, such as 'carefree manoeuvring', can assist aircrew in avoiding potentially hazardous situations and thus help to improve safety margins or avoid potential hazards. In extreme cases, for military aircraft applications, the integration of aircraft systems has led to designs which, without the stability augmentation inherent in the flight control systems, are essentially impossible to fly. In such cases, flight control issues have to be addressed during the design process from the earliest stages and similar issues can arise with other applications (see e.g. [1], [2]).

Although many good examples of integrated system design can be found in aeronautical engineering, similar situations involving multidisciplinary design, system integration, 'control-configured' solutions and embedded controllers can be found in other fields, such as automotive engineering, robotics, wind turbine generators and biomedical engineering. Satellite and space vehicle design is another area where integration of systems is essential due to the importance of satisfying overall design requirements in terms of the energy usage and total mass.

Taking modern road vehicles as an example, it is clear that a car or truck involves many component parts and these cannot be designed in isolation. The engine and transmission characteristics have to be chosen to take account of the mass and other physical dimensions of the vehicle, along with the performance requirements in terms of quantities such as maximum acceleration and fuel consumption (both of which are largely market-driven). The suspension is designed to suit the mass of the vehicle and the engine characteristics and, once again, must satisfy the demands of the market. Any change in one element of the system leads, inevitably, to changes elsewhere so the whole vehicle must be looked at as a single integrated system.

In a similar way, modern wind turbine systems exhibit complex dynamics involving interactions between the turbine and generator elements (involving the rotor, drive train, generator, converter and electrical load), and other dynamic elements such as the tower, substructure and foundations. As wind turbines become larger and involve rotor systems with lower natural frequencies, the interactions between these different dynamic elements become more significant, leading to a need for better models, the use of analysis and design methods that take full account of the integrated nature of the system and the application of more advanced methods of control (see e.g. [3]).

Another rapidly developing application area where system integration is very important involves micro-electromechanical systems (MEMS) and nano-electromechanical systems (NEMS). The potential of very small (micro), nano and, more recently, molecular-sized machines has been recognised for some time and specialised sensor devices such as miniature accelerometers and tuning fork rate gyroscopes with electrostatic actuation are now widely available. Many MEMS and NEMS developments involve medical applications such as blood cell separation and biochemical analysis. Dynamic models have an important role in new developments of this new technology and modelling and simulation methods can help not only in the design process but also in providing a better understanding of performance limitations. These systems are very highly integrated and frequently involve physics, chemistry and biology as well as electronics, control and mechanical engineering. A recent review of modelling, simulation and control aspects of micro- and nano-electromechanical systems has been provided by Ferreira and Aphale [4].

There are many ways in which *complexity* can be defined and one useful way to approach this is to consider complexity in the context of engineering systems using a number of *levels* (see e.g. [5]). The lowest level of complexity involves components or sub-systems from a single engineering discipline and usually involving one organisation. A simple example at this level is an electric motor or generator. At the second level, more than one engineering discipline is involved and there is likely to be more than one organisation concerned with the processes of design, operation or maintenance. A complete electrical power station is an example of a system at this second level of complexity. At the third (and top) level, we are dealing with a system of systems and this can involve many different disciplines and has an impact on non-technical factors involving social, economic and environmental issues. An example at this level of complexity is a complete electrical supply network, involving electrical power generation and distribution.

Whatever the field of application, unless appropriate design methods are applied, the development of hybrid systems, involving mechanical elements, electrical elements, electronics, control and software inevitably introduces additional complexity that can lead to delays, unexpected extra development costs or products that do not meet their specification. These design methods are usually model-driven and involve optimisation at the level of the overall system.

In the design and development of industrial processes, such as those found in chemical engineering and electrical power generation and distribution, an integrated approach supported by efficient modelling and simulation methods and tools is also recognised now as being important (see e.g. [6]). Although the adoption of modelling and simulation in the field of engineering processes has been slower than in other application areas, this may be because modelling and simulation activities in that field are often separated from other aspects of engineering. Simulation should be seen as being of central importance throughout the whole life cycle of an industrial process. In a large process, models may be developed many years before the construction begins. The construction may take years and the expected life of the plant may be several decades, once commissioning has been completed. Simulation models should therefore be recognised as being of central importance for a wide range of activities including the specification stage and investigation of options before detailed design is started, for overall optimisation of the system and for controller development. Once the commissioning stage has been reached, models and associated real-time simulators may be of great value for operator training and also for investigation of possible problems encountered during commissioning. When the plant becomes operational, models and simulations continue to be of value for investigation of operational issues and sustainability, for accident investigation and for investigation of possible plant modifications and upgrading.

In engineering products and processes, the introduction of closely integrated systems involving several distinct engineering disciplines is leading to new trends in design and development. In the past, different aspects of a system were usually designed in a number of distinct and separate phases – involving mechanical elements initially and ending with the electronics, control and software aspects. Problems encountered at a late stage could not be corrected without significant additional cost and delays, and it became common to attempt to deal with such problems through modifications of the software, adding to its complexity and often failing to address underlying issues. *Model-driven design* attempts to overcome some of the problems of traditional sequential design procedures through the introduction of a more concurrent approach.

2.2 Sequential and concurrent design procedures

Figure 2.1 shows the traditional *sequential approach* to design. Here the system specification is the starting point for the process and this usually involves mechanical design as a first stage, typically using computer aided design (CAD) tools. Once that part of the design has been optimised as a separate sub-system, the mechanical engineers pass the design on to their electrical engineering colleagues, who start work on the electrical aspects of the project; this may typically involve selection of motors and drive components, other types of actuators and sensors. Again, use is made of design tools that are specific to the domain in question and the sub-system under consideration is optimised in isolation.

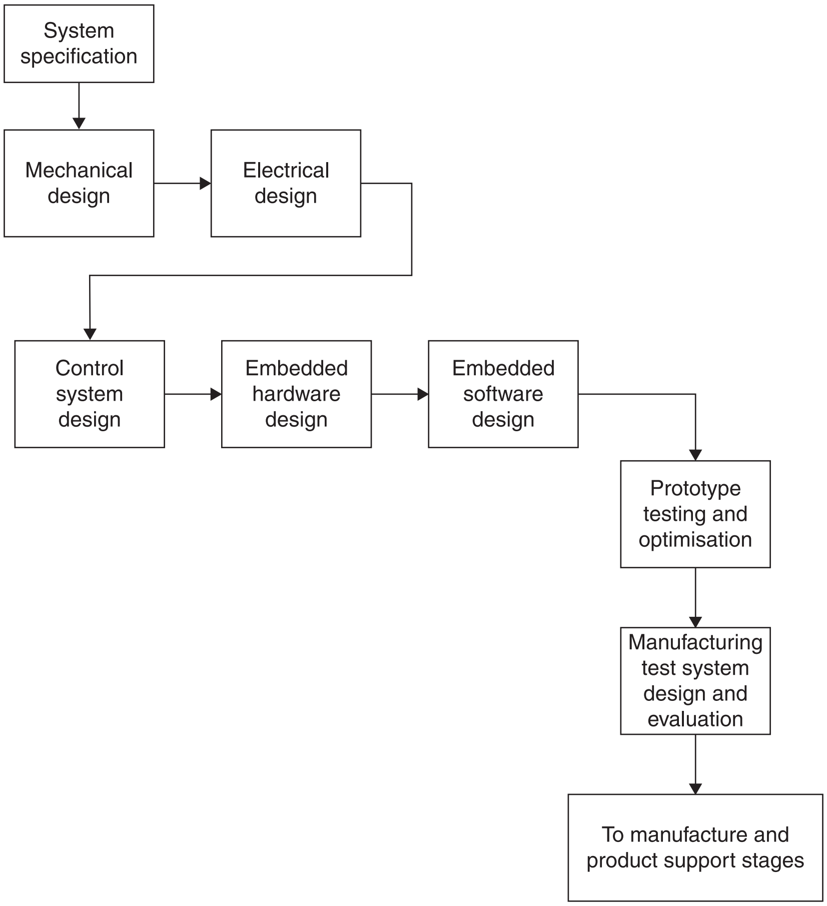


Figure 2.1 **Block diagram of the conventional design process involving sequential design stages. Note that between the five blocks that involve design, there may be many hidden communication links and iterative cycles of design modification. There is also likely to be feedback from the prototype testing stage to each of the earlier design stages in order to optimise the overall system**

At this stage, those involved with the control systems and any embedded hardware and software begin to play a part. Generally, the last aspect of design that is finalised in this sequential approach involves control laws and tuning of the control algorithm. This leads to construction of a prototype system and to testing. Once the testing has been completed successfully, the work progresses to the manufacturing stage but, if the testing process throws up problems, it is clear that the engineers responsible for those aspects must reassess their designs and make changes. These changes are then evaluated in a new or modified prototype system.

As work progresses in the sequential approach of Figure 2.1, the more difficult and expensive it becomes to make changes in the stages represented by the first blocks in this diagram. Rectifying problems can produce long delays and significant additional costs, or lead to acceptance of features of the design that do not fully satisfy the specification.

In the model-driven integrated systems approach, a more *concurrent* type of design procedure is adopted. This is illustrated in Figure 2.2, which shows that this is a *parallel* design process in which the mechanical, electrical, electronic and control aspects of the design are considered together. A virtual prototype is produced using a computer simulation model. This simulation may involve both continuous and discrete-event techniques and such combined simulation models are discussed in Chapter 3. As confidence is built up about some aspects of the design, the virtual prototype may move increasingly towards a 'hardware-in-the-loop' simulation where available prototype hardware operates in conjunction with a simulation model to represent parts of the system that are not available at that stage. After successful testing of the virtual prototype and completion of any further design iterations necessary, the work progresses to completion

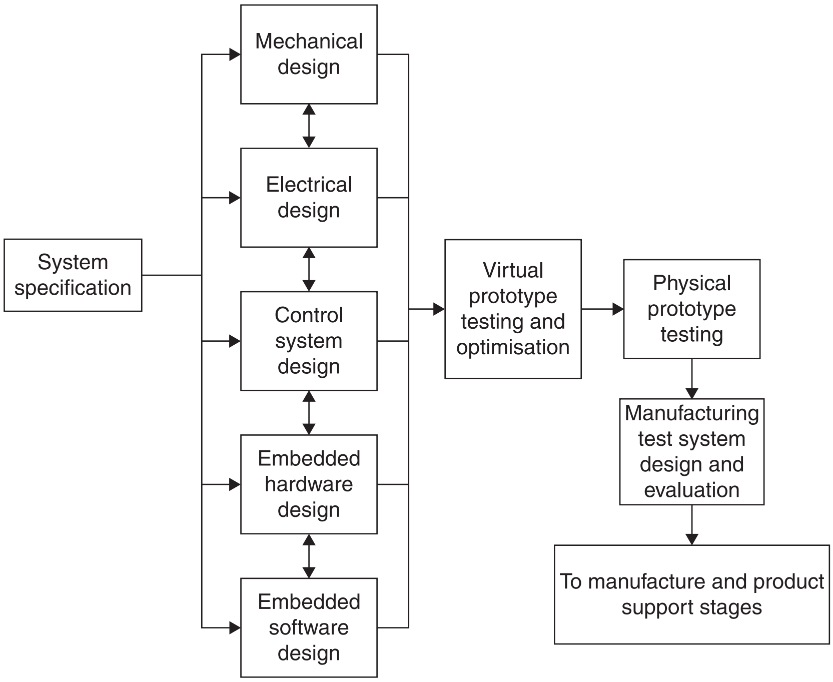

Mechanical design
Electrical design
System specification
Control system design
Embedded hardware design
Embedded software design
Virtual prototype testing and optimisation
Physical prototype testing
Manufacturing test system design and evaluation
To manufacture and product support stages

Figure 2.2 **Block diagram for the concurrent design approach with a virtual prototyping stage. Here communication links between the different design processes are shown explicitly since the design processes for all the sub-systems progress more or less in parallel. There are also important communication links, which are not shown, between the virtual prototype and the design blocks**

of a physical prototype and further evaluation of performance. By making use of multidisciplinary models, engineers are able to reason about system-level properties from a very early stage in a project. The effect of each design decision becomes immediately apparent to all involved, thus lessening the risk of misunderstandings between engineers within different specialist teams.

One of the most important features of the integrated systems approach involving concurrent design is that the success of the design depends on the quality of the models being used by the different design teams. Another issue is the need to integrate the design tools used by the teams, allowing them to work together and share information even if the tools are from different vendors and are intended for specific disciplines within engineering.

The concurrent approach should allow engineers to interact continuously and to discuss how each part of the overall system design is affected by others. Not only does this improve communication links between the designers, but it also helps ensure that there is good communication with the customer. There is increased confidence at an early stage in design that the requirements are fully understood and this helps to reduce the risk of serious errors or oversights and can therefore reduce the development time.

It is clear that, with integrated design, the systems that have to be brought together do not exist in hardware terms when initial design decisions are being made. This contrasts strongly with traditional approaches in some fields, such as control systems engineering. Traditional control system design procedures have usually involved the development of a controller for a 'plant' that already exists, or a plant that has been designed in detail prior to control being considered. In that traditional approach, direct comparisons between the plant model and the system are often possible. In contrast, within the integrated approach to design, control is no longer a second stage in the design process and the design of the control systems cannot be separated from other aspects of the design.

Although this requires a new approach, it need not produce insuperable difficulties, because it is normal to start the design process for an integrated system using some form of highly simplified initial model that only includes essential features. This initial description is intended to provide a basis for the evaluation of major design options and for making preliminary decisions. As soon as more detailed and tested models become available, they are used in place of this initial description and, inevitably, this introduces further complexities.

Within multidisciplinary design teams, the concurrent consideration of critical constraints is central to this integrated design process and this implies a need for models of the highest possible quality for each stage of the development. There is also a need for software tools that allow each team to work within a familiar environment but still produce a complete model that can be understood by everyone concerned and that is accessible to all. Full design integration also requires design teams that are organised so that technical and economic factors may be traded. This, in turn, allows the overall performance to be more fully optimised and design cycle times to be reduced.

The steady increase in complexity of models being used for these new and demanding applications and the computational speed regarded as necessary are introducing new demands on those responsible for the development of modelling and simulation tools. Among the consequences of adopting a concurrent approach to design and the growing importance of embedded systems is that virtual prototyping and hardware-in-the-loop simulation techniques are now commonplace. This means that the final stage of the development process involving a complete prototype system can be delayed. Provided the models involved in the virtual prototype are accurate, it can be used to identify features that could be regarded as weak points within the overall design and steps can be taken to improve the design before the more costly physical prototype is built. Such a process of stepwise refinement from the virtual prototype stage reduces overall costs and helps to reduce the risk of major problems at later stages.

Within multidisciplinary design teams, the concurrent consideration of critical constraints is central to this integrated design process and this implies a need for models of the highest possible quality for each stage of the development. There is also a need for software tools for dynamic modelling and simulation that can be integrated with other design software.

As an illustration, methods of computational fluid dynamics and finite element modelling are widely used in many areas of engineering. When such tools are used for the modelling of elements within a larger system involving a number of sub-systems, it may be essential to derive reduced-order descriptions to help avoid the effects of major computational overheads when sub-models are being combined to provide a more integrated description of the larger system. This model reduction process, inevitably, introduces approximations and simplifications which must be assessed carefully.

Often the need for reduced models arises because of the inevitable difficulties produced by computational timescales that are much greater than timescales convenient for human analysis and decision-making. Sub-system models should also be capable of running in time-scales that are well matched to the thought processes of human designers. Models are never unique and, at each stage of a project, whether it involves engineering design or open-ended scientific investigation, it is important that models are being properly matched to the intended application, not only in terms of quality but also in terms of computational issues such as speed.

The model-centred approach, concurrent methods of design and the development of virtual prototypes by domain-specific design teams that may be geographically distributed across the world all represent aspects of a trend towards what has been termed *computer-supported cooperative work*. This is concerned with issues of collaborative design, coordination methods and virtual prototyping, and prototype to product transition. One example of an annual conference relating to computer supported cooperative work is the IEEE (Institute of Electrical and Electronics Engineers) International Conference on Computer Supported Cooperative Work in Design (CSCWD) [7]. A special issue of the IEEE Transactions on Systems, Man and Cybernetics has been devoted recently to the theme of systems integration and collaboration in design, manufacturing and services [8]. In some fields, such as the design of integrated electronic circuits and systems and digital signal processor systems, tools already exist that are being used for virtual prototyping, such as VHDL (Very High Speed Integrated Circuit Hardware Description Language) (see e.g. [9]) and UML (Unified Modelling Language) (see e.g. [10]). For systems involving a wider range of engineering disciplines, no single tool appears to provide a solution at present.

A distinction also has to be made between the design of high-volume products which are of relatively low cost but which may require short development times and the design of low-volume, higher-cost products, for which much longer development times may be acceptable. For the first group, the requirements may include fast modelling tools and access to good libraries of well-documented and fully validated sub-models or even existing and properly documented generic models. In the second case, there may well be time to embark on more extensive simulation studies from first principles, provided the potential gains are seen to justify the inevitable high costs of such a strategy.

2.3 References

[1] Hoh, R.H. and Mitchell, D.G. (1996) 'Handling-qualities specification – a functional requirement for the flight control system', in Tischler, M.B. (ed.), *Advances in Flight Control Systems*, Taylor and Francis, London, UK.

[2] Barry, J.M. (1998) 'A conceptual approach to developing models and simulations', in Cloud, D.J. and Rainey, L (eds.) *Applied Modeling and Simulation: An Integrated Approach to Development and Operation*, McGraw-Hill.

[3] Pao, L.Y. and Johnson, K.E. (2011) 'Control of wind turbines: approaches, challenges and recent developments', *IEEE Control Systems Magazine*, Vol. 31, No. 2, pp. 44–62, 2011.

[4] Ferreira, A. and Aphale, S.S. (2011) 'A survey of modelling and control techniques for micro- and nanoelectromechanical systems', *IEEE Trans. on Systems, Man and Cybernetics, – Part C: Applications and Reviews*, Vol. 41, No. 3, pp. 350–64.

[5] Elliott, C. and Deasley, P. (eds) (2007) *Creating Systems that Work: Principles of Engineering Systems for the 21st Century*, The Royal Academy of Engineering, London, UK.

[6] Juslin, K. (2007) 'Experiences and trends in modelling and simulation of integrated industrial systems', in Zupančič, B., Karba, R. and Blažić, S., *Proceedings of the 6th EUROSIM Congress on Modelling and Simulation, EUROSIM 2007, 9–13 September, 2007, Ljubljana, Slovenia*, Vol. 2 (Full papers), ARGESIM – ARGE Simulation News, Vienna, Austria (ISBN 978–3–901608–32–2).

[7] International Working Group on Computer Supported Cooperative Work in Design (CSCWinDesign) (online): *www.cscwid.org* (accessed 3 July 2011).

[8] Shen, W., Borges, M.R.S., Bathès, J.P. and Lao, J. (2011) 'Forward to the special issue on systems integration and collaboration in design, manufacturing and services', *IEEE Trans. Systems, Man and Cybernetics, Part C: Applications and Reviews*, Vol. 41, No. 3, pp. 281–3.

[9] Christen E., Bakalar K., VHDL-AMS-a hardware description language for analog and mixed-signal applications, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 46, No. 10, pp. 1263–72, 1999.

[10] Holt, J., *UML for Systems Engineering: Watching the Wheels*, Second Edition, IET, Stevenage, UK, 2004.

3

Problem organisation

Abstract: Models used in engineering design are refined as the design moves forward and as uncertainties are eliminated. This modelling process can involve tiers, starting with descriptions that are of relatively low fidelity and leading to a model at the final tier that is a detailed description of the system in service. At every stage within this process of stepwise refinement, models may be developed using a layered structure, starting with a description of the physical components and leading to a network-based representation and then to mathematical and computer-based descriptions. This chapter introduces methods and tools for model organisation and development. Aspects covered include bond graph descriptions, differential equations and differential algebraic equation descriptions, state-variable models, transfer function models, models involving distributed parameter elements and an introduction to some commonly used modelling and simulation software tools.

Key words: tier, layers, bond graph, state-space, transfer function, discrete-event, discontinuity, software.

3.1 Model organisation for engineering systems design

Issues of problem organisation for system modelling are difficult to separate from issues relating to the choice of software tools. Some tools allow users to consider the overall problem before becoming involved in equations and numerical solutions, while others are aimed at efficient simulation but lack facilities for analysis of the physical problem. The modelling and analysis stage is particularly important for multi-domain systems and tools that provide a unified environment for modelling and simulation of multidisciplinary systems are particularly valuable.

Models used for design, like the design process itself, can be described in terms of tiers within a procedure that involves *stepwise refinement*. Tier 1 involves *concept design* and, at this stage, models are of relatively low fidelity but capture the essential features of the systems. Such models provide approximate estimates of performance, but are nevertheless useful for comparing initial design options. Tier 2 involves *preliminary design* and uses intermediate-level models based on design information from the first phase. Tier 3 involves *detailed design*. Thus, in most design projects, models are developed with different levels of complexity for each of these tiers.

The modelling of an engineering system usually starts with a schematic diagram showing the system components and their interconnections. This can be linked to text discussing assumptions made and factors that have been deliberately ignored or regarded as being of minor importance. This is also a starting point for the development of a layered structure for a model for each of the design tiers. This layered structure provides a valuable framework around which documentation can be prepared.

The first layer (Layer 1) involves separately identifiable physical components such as coils, amplifiers, electric motors, gear trains, etc. Each component or sub-system usually involves one engineering discipline and one organisation. The second layer (Layer 2) involves a network based on the elements of the first layer but viewed in terms of physically based idealisations involving concepts such as resistance, inductance, moment of inertia or friction. The third and final layer (Layer 3) involves a representation in mathematical terms where the idealised elements of the second layer are translated into a set of linked mathematical equations. It is important to note that the real-world objects of Layer 1 have no properties assigned *a priori* and these are defined in Layer 2, where account is taken of the intended application of the model. For example, for any application, an electrical coil is likely to have properties that include inductance and resistance, but if it is to be used in a high-frequency application, it may also have capacitance.

Models for use at the Tier 1 of design are usually simple in form and, although there may be a recognisable layered structure, the three layers at this initial stage contain far less detail than at the Tier 3 stage of design. Initially, the component representations and associated sub-models in each layer may involve many uncertainties and include only essential features of the real-world elements. However, the models provide enough information to allow preliminary design decisions to be made. As the iterative process of design moves through Tier 2 to Tier 3, the developers' attention becomes focused on specific options and on refinement of the features of the models that are most relevant for the design objective. Modelling uncertainties are identified and strategies are established for model reduction using data from component manufacturers or from tests on the hardware. These steps are considered in more detail in

Chapters 6 and 7 and, at Tier 3 in the design process, lead eventually to a complete description in each layer.

One topic that is not considered in detail in this book is *qualitative* modelling. Traditional models are quantitative and involve variables that take values that can be represented numerically. Qualitative models lack the precision of quantitative models and variables are discretised using broad categories such as very large, large, medium, small and very small to produce a form of *fuzzy* description. Further details of qualitative modelling techniques may be found elsewhere (see e.g. [1]).

3.2 The physical component layer

The first of these layers (Layer 1) involves a description of the system as a set of separately identifiable real-world objects such as resistors, motors, gear trains, etc., along with the information about how we observe or obtain data from that system and the environment within which it operates.

3.3 The physical concept layer

For each component of the system, we need to establish physical mechanisms that are relevant. For example, for the modelling of an electrical coil, as discussed in the introductory section of this chapter, it is necessary to establish whether or not capacitance should be included. We need, however, to retain flexibility in developing physically based descriptions of the real-world elements and this is particularly important in considering models that involve more than one physical domain (such as the electrical and mechanical domains). A system involving an electric motor, an associated gear train and a mechanical load is an example of a system involving multiple domains. One way of dealing with such cases is to use physical analogies to gain insight about overall behaviour, together with a network approach involving energy.

Ideally, when a decision is made to use a model, questions arise about the level of detail necessary. For example, in an electrical circuit the choice of a model element for a physical resistor should raise questions about the application and the frequency range over which the resistor is used. Essentially we need to decide whether resistance is the only property of importance or, because of the frequency range of interest, whether we should add additional physical effects such as capacitance or inductance. A specific physical resistor may thus have different forms of model to represent it, depending on the application. Similar issues arise with other physical components, mechanical as well as electrical. For example, the choice of model for a shaft involves decisions about whether or not internal damping and torsional stiffness effects are included.

3.3.1 Bond-graph representations

In the preliminary stages of modelling, details of interactions between physical components are not available. Various different levels of detail can be provided at interfaces between elements and refinement of sub-models inevitably leads to refinement of the interfaces. For the physical process layer, a natural way of handling sub-model interactions is provided by *bond graphs*. The interface often involves *ports* at which there is exchange of energy or information and these ports can be regarded as a refinement of the interface elements discussed in connection with the physical component layer.

Ports also define the link between the physical process layer and the mathematical layer using what are termed *port variables*.

The bond-graph approach is widely used and has origins in the mathematical theory of linear graphs and the nineteenth-century work of Kirchhoff on electrical networks. Bond graphs are based primarily on energy and power, and are particularly useful for interdisciplinary projects. Energy ports for components of a model are connected using bonds and these specify energy transfer. The system structure is kept separate from the equations through the graphical representation used in bond-graph models and this assists in establishing a qualitative physical understanding of a model (see e.g. [2]).

Bond graph ideas were devised by Professor Henry Paynter at the Massachusetts Institute of Technology (MIT) in the late 1950s [3]. Former PhD students of Professor Paynter developed the approach further, including D.C. Karnopp, D.L. Margolis and R.C. Rosenberg [4], and many research papers have since appeared. Special issues on aspects of bond-graph modelling have appeared in many journals (see e.g. [5], [6] and [7]) and an extended tutorial on bond-graph methods by P.J. Gawthrop and G.P. Bevan was published in the *IEEE Control Systems Magazine* in 2007 [8]. A useful report providing an introduction to bond graphs has been published by J.F. Broenink of the University of Twente in the Netherlands [9]. Links between bond graphs, linear graph theory and graph-theoretic modelling methods are described in a paper by Birkett *et al.* [2].

The relationships between elements of a model in bond-graph form are highlighted in a very direct fashion at a very early stage in the modelling process rather than through equations or a simulation program on a computer. Indeed, the initial stages of model development using a bond-graph approach can be carried out entirely with pencil and paper, and from this starting point information can be found about algebraic loops, constrained variables, the effects of possible simplifications of the model and other qualitative insights. In the case of complex models, computer-based modelling tools can be useful for generating bond graphs and some of these are discussed later in this section.

Bond-graph modelling depends on the fact that analogies exist between dynamic systems of different kinds. Electrical, fluid and mechanical systems can all involve similar forms of differential equation description and bond-graph modelling is based on three specific types of analogy: signal analogies, component analogies and connection analogies.

Signal analogies

Bond graphs involve effort and flow variables, and depend on signal analogies to provide links between variables in different physical domains. Effort signals include mechanical forces and electrical voltages, whereas flow signals include electrical currents and mechanical velocities. The product of effort and flow in both of the electrical and mechanical domains is power. This is also true in other domains and, in general, it is always possible to write:

$$\text{effort} \times \text{flow} = \text{power} \tag{3.1}$$

In bond graphs, effort signals are conventionally represented by the generic symbol e and flow signals by the generic symbol f. Integrated flow signals include electrical charge and mechanical displacements, and are represented using the generic symbol q. Integrated effort signals include electrical lines of flux and mechanical momentum. Table 3.1 shows analogous signals for translational and rotational mechanical systems, electrical systems and hydraulic systems.

Table 3.1 **Analogous signals**

Bond graph	**Electrical**	**Mechanical rotation**	**Mechanical translation**	**Hydraulic**
Effort e	N Voltage V V	Torque τ N-m	Force F	Pressure P Pa
Flow f	Current I A	Angular velocity Ω rad s^{-1}	Velocity v ms^{-1}	Flow Q m^{3}s^{-1}
Integrated effort	Lines of flux λ V-s	Angular momentum h kg m^2 rad s^{-1}	Momentum p kg ms^{-1}	Momentum per unit area p kg-ms^{-1}
Integrated flow	Charge q C	Angle θ rad	Position x m	Volume V m^3

Effort and flow signal pairs are represented by a single power bond, as shown in Figure 3.1. The direction of the half arrow in this diagram shows that the positive direction of energy flow is from left to right. Power bonds such as this must be distinguished from active bonds which resemble signals in block diagrams and can carry either effort or flow. Figure 3.2 shows an active bond carrying flow and elements such as this provide a useful interface between a bond graph sub-model and a block diagram.

Component analogies

Table 3.2 shows analogous components with a single energy port from the mechanical, electrical and hydraulic domains.

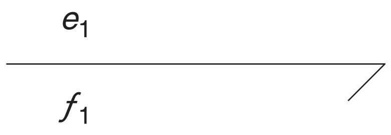


Figure 3.1 **Representation of a power bond carrying an effort and flow signal pair**

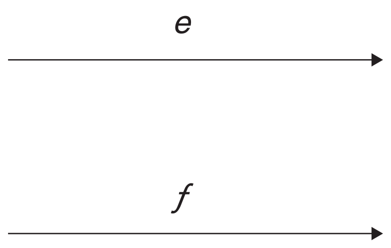


Figure 3.2 Representation of active bonds carrying either effort or flow that can act as a link between a bond-graph model and a block diagram

Table 3.2 Analogous components with one energy port, corresponding to the analogous signals of Table 3.1. The first column shows the generic bond-graph component while other columns show domain-specific components

Bond graph	**Electrical**	**Mechanical rotation**	**Mechanical translation**	**Hydraulic**
Se *De*	Applied voltage Voltmeter *V* V	Applied torque Torque sensor *T* N m	Applied force Force sensor *F* N	Applied pressure Pressure sensor *P* Pa
Sf *Df*	Applied current Ammeter *I* A	Angular velocity Tachometer Ω rads^{-1}	Applied velocity Speedometer *v* ms^{-1}	Flow Flow meter *Q* m^{3}s^{-1}
C	Capacitor *C* F	Torsional spring *K* N m rad^{-1}	Spring *K* N m^{-1}	Accumulator *K* Pa m^{-3}
I	Inductor *L*H	Moment of inertia I NMs^2 rad $^{-1}$	Mass *m* kg	Flow inertia *I Pa* m^{-3} s^2
R	Resistor *R* Ω	Torsional damper *d* N m s rad^{-1}	Damper *D N* s *m*$^{-1}$	Restrictor *K* Pa s m^{-3}

Here the generic *Se* component is a source of effort and could represent, for example, an ideal voltage source or an ideal applied force. The *Sf* component, similarly, is a generic flow source component representing an ideal current source or an applied velocity. The two generic detectors, *De* and *Df*, are detectors of effort or flow, respectively. A *De* component could therefore represent a voltmeter or force sensor while a *Df* component could represent an ammeter or a tachometer. A *C* component stores energy and corresponds to an electrical capacitor or mechanical spring. An *I* component also stores energy and could represent an electrical inductor or a mechanical mass or moment of inertia. The *R* type of component dissipates energy, as in a resistor or a mechanical damper. In addition to the components that appear in Table 3.2, there are *SS* components that can be used to model co-located sensor-actuator pairs represented as *Se-Df* or *Sf-De*.

Due to the fact that a particular type of component is likely to occur many times within a given complex model, it is essential to be able to distinguish between different instances of each component type. Thus *R:r1* is used to refer to a dissipative component *r1*.

In the case of linear components, the equations corresponding to the generic *C, I* and *R* components are:

$$\mathrm{C}\left\{\begin{matrix} e=\dfrac{q}{c} \\ \dot{q}=f \end{matrix}\right\} \tag{3.2}$$

$$\mathrm{I}\left\{\begin{matrix} f=\dfrac{p}{m} \\ \dot{p}=e \end{matrix}\right\} \tag{3.3}$$

$$\mathrm{R}\left\{e=rf\right\} \tag{3.4}$$

In these three equations, the quantities *c*, *m* and *r* are constants relating to elements of the physical system being modelled. In electrical terms the upper relationship in Equation (3.2) corresponds to Coulomb's Law or, in mechanical terms, to Hooke's Law, while the upper statement in Equation (3.3) is Newton's Second Law. The relationship in Equation (3.4) represents Ohm's Law and its equivalents for mechanical and hydraulic elements.

Connection analogies

Any two components can be connected using a power bond. Figure 3.3 shows an example involving a capacitor and a resistor, where the components have the same current (flow) and voltage (effort). The bond graph *C:c* and *R:r* components are linked using a power bond as shown in the diagram. The colon notation links the label *c* with the *C* component and the label *r* with the *R* component.

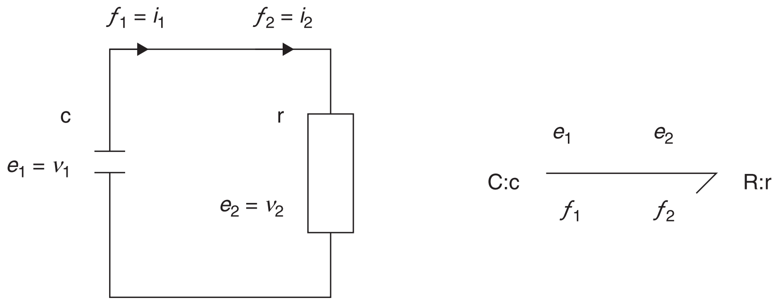
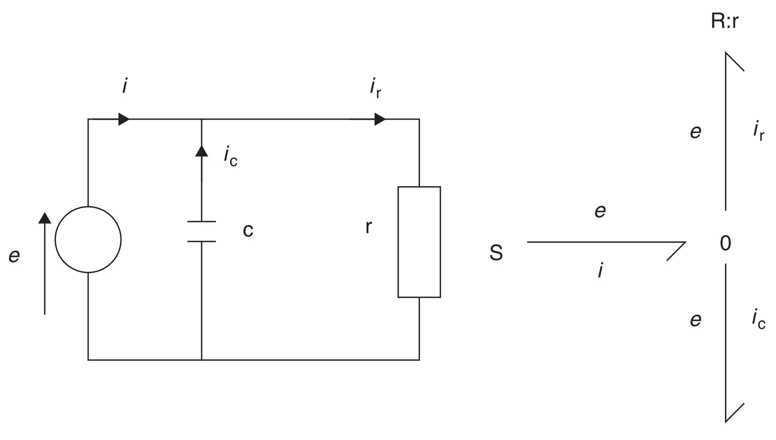
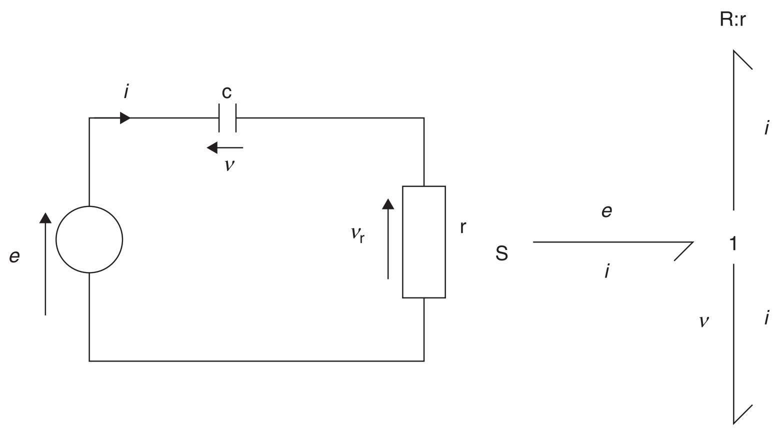


Figure 3.3 **A capacitor and a resistor connected in series so that the two components carry the same current (flow) and have the same voltage (effort). In terms of a bond-graph representation, the components are connected using the power bond shown on the right**



(a) Parallel connection



(b) Series connection

Figure 3.4 **Example illustrating connection of components (a) in parallel and (b) in series**

In general, connections are either parallel or series and Figure 3.4 provides a simple example. The parallel connection obeys Kirchhoff's voltage law while the series connection obeys Kirchhoff's current law. The corresponding bond-graph models use what is termed a *0* junction for the parallel circuit (involving common effort) and a *1* junction for the series circuit (involving common flow). The efforts for the bonds associated with a *0* junction are the same but the flows for these bonds add up to zero. Similarly, the efforts on a *1* junction sum to zero while the flows are all equal.

Power conversion in bond graphs

Effort and flow variables in the physical domains of Table 3.1 have different units and it is impossible to connect them directly. Since power is common to all physical systems, it is possible to connect different physical domains on the basis of power. This is done using the power-converting bond-graph components *TF* (a generic transformer) and *GY* (a generic gyrator). These two generic components are shown in Figure 3.5.

The ***TF*** component conserves power and represents a generalisation of an electrical transformer. In the linear case it has the property that:

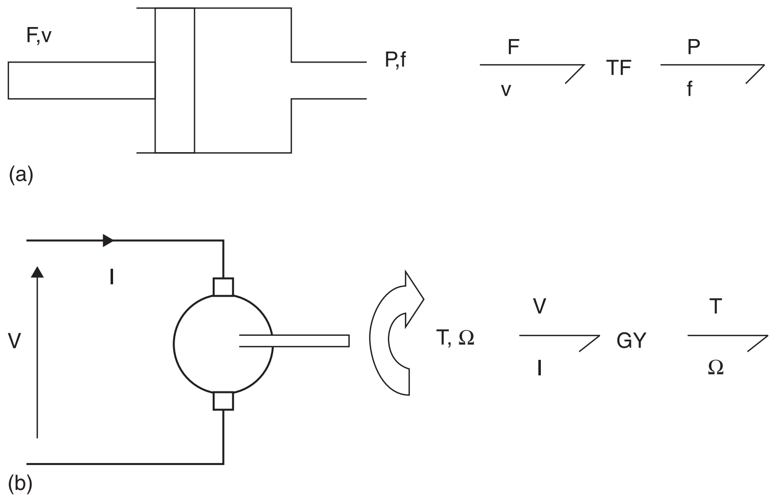


Figure 3.5 **Examples of bond-graph coupling components. (a) Bond-graph representation of a hydraulic cylinder and piston (a transformer). (b) Bond-graph representation of a dc electric motor (a gyrator)**

$$e_2 = ne_1 \text{ and } f_1 = nf_2 \tag{3.5}$$

The *GY* component also conserves power and is similar in some respects to a transformer, but the flow in a gyrator at one port depends on the effort at the other. Thus the equations for the linear case are:

$$e_2 = kf_1 \text{ and } e_1 = kf_2 \tag{3.6}$$

where n and k are non-dimensional constants. Since power is conserved for the transformer and for the gyrator, the input power and the output power must be the same in both cases, so that:

$$e_1f_1 = e_2f_2 \tag{3.7}$$

Causality in bond-graph models

Causality is not established in bond-graph models until the initial modelling is complete. This is different from block diagram models where the diagram represents a set of assignment statements instead of equations. A block diagram cannot be constructed until inputs and outputs of each component have been defined. However, in bond-graph components, inputs and outputs are determined after modelling through the assignment of computational causality.

Creation of a-causal bond graphs for a given system can provide a complete model. However, this can involve many different sets of equations depending on the purpose of the modelling exercise and the type of analysis to be carried out. The state-space type of representation so widely used in continuous system simulation and control systems analysis and design involves one particular form of causality, while the Lagrangian or Hamiltonian types of representation involve other forms of causality. A particular representation can be obtained from a bond-graph model by applying a specific pattern of causal strokes to the a-causal model.

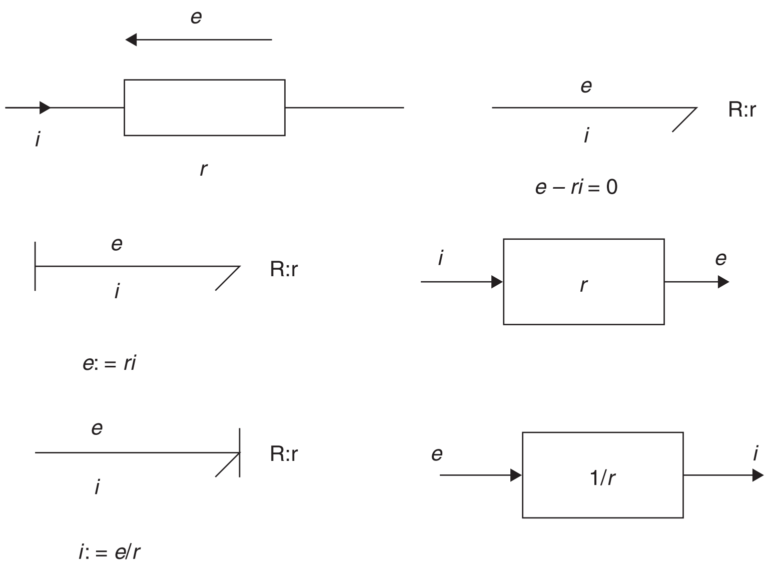


Figure 3.6 **The representation of causality in bond-graph components. The top bond graph is *a*-causal and represents the equation relating the variables *e* and *i* to the parameter *r*, which is resistance. The bond graph in the middle has a causal stroke indicating that the variable *e* is the output. The corresponding block diagram element is shown at the middle level on the right. The third bond graph has a causal stroke showing that *i* is the output and this is reflected again in the block diagram representation at the lower right**

Advantages and disadvantages of bond-graph models

One feature of bond-graph models is that they can involve several description levels and this is important for hierarchical modelling and the recognition that a single physical system can be modelled at a variety of different levels or tiers, as indicated in Section 3.1. Generation of a bond-graph model starts at the physical components and can lead directly to qualitative analysis or to a mathematical description and a simulation model. Parameter sensitivity analysis techniques based on bond-graph concepts can be applied. This topic, discussed in more detail in Chapter 5, is important for applications involving system optimisation or analysis of the effects of model uncertainties on overall performance. Inverse models can also be derived using the bi-causality concept and this is discussed further in Chapter 4.

The adaptable form of interface in bond-graph models is an important benefit of this approach. This is particularly helpful in making sub-system models reusable and when integrating system models. If all the system models are in bond-graph form, then all the interfaces transmit power and this can significantly simplify the task of bringing a set of sub-models together to represent a complete system. Another advantage is that causality of interfaces can be altered without producing new models and this reduces the number of models required in a design project, thus reducing the risk of errors and the costs of model documentation.

The idea of generic components and variables is useful for the modelling of multi-domain systems and can expose interactions that may otherwise remain hidden. This can be particularly helpful in considering possible simplifications.

Disadvantages of bond-graph models are mainly associated with the fact that they are still unfamiliar to many engineers. Although some find the terminology and notation difficult initially, the whole approach is based on concepts of energy and power which are familiar to engineers of all disciplines and this can be helpful. It should be remembered that some engineers also find signal flow diagrams and block diagrams difficult to accept, because those representations do not involve energy and power explicitly. Engineers can quickly adapt to the use of bond-graph representations and benefit significantly from their use, especially in multidisciplinary problems. Transformation of bond-graph models into other forms of representation is straightforward and software tools exist to do this.

Examples of modelling languages and simulation software for handling bond graphs

As has already been pointed out in Section 3.3.1, much of the initial model development process using a bond-graph approach can be carried out using a pencil and paper. However, there are computer-based modelling tools that allow bond-graph models and similar port-based representations to be mapped to code. Examples include Dymola [10], CAMP-G (Computer Aided Modeling Program with Graphical Input) [11] and 20-sim [12]. Some of these software tools are discussed at more length in Section 3.5.

3.4 The mathematical description layer

In physically based models, many variables are treated as *continuous variables* that are described mathematically as continuous functions of time (and possibly also as continuous functions of some other physical quantity such as position). A model expressed in terms of continuous variables leads usually to a description involving ordinary or partial differential equations.

In the case of partial differential equations, numerical techniques such as finite-element modelling or computational fluid dynamics can provide solutions very effectively, but the computer time required introduces problems when such methods are used within the engineering design process. Computational tools do exist which can provide a stepping stone between simulation models and dedicated CAD formats for distributed parameter models. One example is COMSOL Multiphysics® [13] (formerly FEMLAB), which is a general-purpose modelling and simulation tool that allows a wide range of physical phenomena involving electrical, mechanical, thermal, fluid flow and other distributed parameter problems to be considered within an integrated environment with a convenient user interface and powerful graphics for displaying results in two or three dimensions. COMSOL Multiphysics® also allows users to make connections to other software such as MATLAB® [14] and also some CAD packages including SolidWorks® [15], Autodesk® Inventor® [16] and Creo Parametric® (formerly Pro/ENGINEER®) [17].

Even with highly specialised computer hardware and software facilities, a significant mismatch may arise between the times required to perform simulation runs with complex distributed parameter models and the expectations and desires of the designer. When overall system design optimisation is necessary, the computational overheads of modelling complex systems using partial differential equations may be prohibitive.

Complex physically based descriptions involving distributed-parameter models can often be reduced to lumped parameter approximations, leading to ordinary differential equations, which offer significant computational benefits without losing physical insight. Examples of this type of approach arise in aircraft systems engineering, where aircraft models often involve elements which are approximations derived from numerical solutions of partial differential equations. Such approximations may be useful only for a limited set of operating conditions. Other examples include systems that can be described very accurately by pure or distributed time-delay models in specific types of application.

Some simulation tools allow lumped-parameter mathematical models to be formulated using port-based methods that involve flow and effort variables together with a-causal principles, as discussed in Section 3.3.1 in the context of bond-graph models. One example of this is the *Virtual Test Bed* (VTB) [18], which is discussed in more detail in Section 3.5.3. The VTB allows models to be created using *natural coupling*, which ensures that the physical laws of energy conservation apply at the model ports. This is achieved through use of the *resistive companion method* (RCM) (see e.g. [19]), as defined in the VHDL-AMS simulation standard (see e.g. [20]). The VTB also allows signal coupling between model ports when this is appropriate.

One advantage of natural coupling claimed by the developers of the VTB is that naturally coupled models can be interconnected very easily to form larger models. Thus, the use of natural coupling principles can facilitate the development of libraries of sub-models in terms of model objects that can be readily coupled together using an object-oriented type of approach.

Object-oriented methods are also emphasised in the case of the Modelica® language [21]. This is a simulation tool that allows a-causal methods to be applied through use of a declarative modelling style in which models are based on equations rather than assignment statements. In assignment statements, variables on the left-hand side of an expression are always results of a calculation and variables on the right-hand side are quantities that are known at the start of the calculation. In an equation, it is not specified which quantities are inputs and which are results; causality is therefore initially unspecified and fixed only when the equation is solved. The equation-based approach is well suited to representing the physical structure of the system being modelled and is a natural approach. One consequence of the equation-based methodology is that the equations may involve a combination of ordinary differential equations and algebraic equations, and this leads to a model described by *differential algebraic equations* (DAEs) as outlined in Section 3.4.3.

In contrast with continuous mechanical, thermal or electrical variables, as discussed above, a controller implemented using digital logic, or a processor, is likely to be described using *discrete-event* modelling methods. This normally involves difference equations rather than differential equations. Other cases that involve discrete-events and discontinuities are found in hydraulic valves and limiters, on/off switches, logical overrides and many other components of practical systems. Combined or hybrid models that describe such elements introduce difficulties, since many simulation tools currently available are not well suited to describing systems involving discontinuities and where the continuous and discrete elements may be tightly coupled. One example of a hybrid system is an automatic transmission system where there are first, second, third and fourth gears. Others can be found in automatic fault detection and recovery systems where the way in which a given actuator is used may change suddenly when a fault is detected. Interest in hybrid systems has grown in recent years (see e.g. [22] and [23]) and some software tools, such as Modelica® [21], provide comprehensive facilities for simulation of discontinuous and discrete elements, even involving packet-switched communications protocols.

One interesting development involves the *state quantisation approach* proposed initially by Zeigler *et al.* [24] which involves replacing discretisation of time by discretisation of state and provides the basis of a new approach to numerical integration. One specific and powerful discrete-event simulation methodology, which is known as the *Discrete-Event System Specification* (DEVS) formalism, has been found to be well suited to implementation of state quantisation. The DEVS approach, which is discussed again in the context of discrete-event system modelling in Section 3.5, was introduced by Zeigler [25] and provides a methodology for discrete-event simulation that is firmly based on system theory. In a further development of the ideas put forward by Zeigler *et al.* [24], Kofman and Junco [26] in 2001 introduced the *quantised state system* (QSS) formalism, which allowed efficient discrete-event simulation of large and complex continuous time systems using the DEVS type of discrete approach. The significant advantage of this is that the DEVS formalism allows for asynchronous updates of model variables and this can be important in the case of models involving a wide range of time-constants or natural frequencies, leading to reduced computing costs as each variable has its own update rate. A Dymola/ Modelica® library known as ModelicaDEVS has been developed [27], which implements a number of QSS simulation algorithms.

Some other issues relating to combined discrete-event and continuous system simulation are discussed in Section 3.5 in the context of specific simulation tools. One interesting development is the availability of symbolic computing software, such as that provided within the Maplesim® [28] and Mathematica® [29] packages. Symbolic computing provides an exact solution in the form of analytical expressions and can provide insight into how parameters affect solutions. This can complement simulation methods but, although symbolic computing can be more efficient than numerical solutions for relatively simple modelling problems, this approach has clear limitations as models become more complex. Used as part of the tool-set of the system modeller, symbolic computing techniques can be very useful.

3.4.1 Equation-based and reduced form representations for lumped-parameter models

Once an appropriate model structure has been obtained in lumped parameter form, equations can be written down to provide an equivalent mathematical description. This process can be illustrated using an example involving the simple electrical circuit of Figure 3.7. This circuit involves an Inductance L, a capacitance C and two resistors. One resistor (R) is in series with the inductance while the second resistor (r) is connected in parallel with the capacitance. The voltages $e(t)$ and $v(t)$ and the current $i(t)$ are then related according to Kirchhoff's voltage and current laws by the following equations:

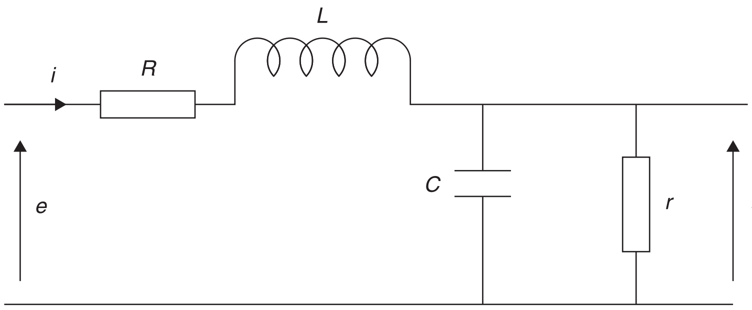


Figure 3.7 **Electrical circuit example involving inductance *L*, capacitance *C* and two resistors (*R* and *r*)**

$$e(t) = Ri(t) + L\frac{di(t)}{dt} + v(t) \tag{3.8}$$

$$i(t) = C\frac{dv(t)}{dt} + \frac{1}{r}v(t) \tag{3.9}$$

Equations (3.8) and (3.9) provide a basis for a complete description of the behaviour of the circuit, in terms (for example) of its response to changes of the voltage $e(t)$ and may be combined to form a single second-order ordinary differential equation:

$$\frac{di(t)}{dt} = C\frac{d^2v(t)}{dt^2} + \frac{1}{r}\frac{dv(t)}{dt} \tag{3.10}$$

Substituting this expression for $\frac{di(t)}{dt}$ in Equation (3.8) then gives:

$$LC\frac{d^2v(t)}{dt^2} + \left(\frac{L}{r} + RC\right)\frac{dv(t)}{dt} + \left(\frac{R}{r} + 1\right)v(t) = e(t) \tag{3.11}$$

This is an example of a *reduced form* model. Equivalent second-order equations could be derived for simple mechanical, hydraulic, thermal or process systems. Although such linear ordinary differential equations may be solved analytically, a reduced form model which is nonlinear can, in general, only be solved using numerical methods or simulation software.

If we return to the basic equations describing this model (Equations (3.8) and (3.9)), it can be seen that these two equations could be combined with the algebraic equation (3.12) to form a different type of model in which the variables $i(t)$ and $e(t)$ are unknowns.

$$v(t) = kf(t) \tag{3.12}$$

Here, the function $f(t)$ could represent any function of time, such as a unit ramp (t) or a sinusoid $(\sin \omega t)$, to describe the specific form of the voltage $v(t)$.

$$L\frac{di(t)}{dt} = e(t) - Ri(t) - v(t) \tag{3.13}$$

$$C\frac{dv(t)}{dt} = i(t) - \frac{1}{r}v(t) \tag{3.14}$$

$$v(t) = kf(t) \tag{3.15}$$

The three equations above constitute a differential algebraic equation (DAE) which cannot be solved using the standard approaches for ordinary differential equations. Here the voltage $v(t)$ is being constrained to follow a specific waveform, such as a ramp or sinusoid, and solution of the model equations should allow us to find the form of the voltage $e(t)$ and current $i(t)$ to make this possible. The properties of differential algebraic equations and methods for their solution are discussed further in Section 3.4.3.

3.4.2 Models in state-space form

Although reduced-form representations are often convenient when a linear model is being studied analytically, a mathematical representation that is often convenient for simulation purposes involves a set of simultaneous first-order ordinary differential equations. The basic principles of the numerical solution of sets of first-order ordinary differential equations – a topic of fundamental importance in continuous system simulation – are presented briefly in Appendix A2. Further details of these numerical integration techniques may be found in many textbooks on modelling and simulation (see e.g. [25], [30], [31] and [32]).

For an *n*th order model in reduced form, there must be *n* first-order equations and this set of *n* equations forms a *state-space* description. Each of the variables of a state-space model is a *state-variable* and, in general, the number of state variables must be sufficient to allow the behaviour of the system to be predicted, given information about the initial conditions and the input forcing to be applied (the function $f(t)$ in the mechanical example above). The general form of an nth order linear state-space model is:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{3.16}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \tag{3.17}$$

where $\boldsymbol{x}$ is the column vector of n state variables, $\boldsymbol{u}$ is a column vector of m input variables, $\boldsymbol{y}$ is a column vector of p output variables, $\boldsymbol{A}$ is an $n \times n$ square matrix of model parameters, $\boldsymbol{B}$ is a matrix of parameters involving n rows and m columns ($n \times m$), while $\boldsymbol{C}$ and $\boldsymbol{D}$ are $p \times n$ and $p \times m$ matrices of parameters respectively. Equation (3.16) relates the rate of change of the state to the present state and the input. It has a form that is especially convenient for simulation, since the numerical solution can be obtained for each equation within the state-space model simply by integration. The quantity on the right-hand side of each of the equations within the matrix-vector representation of Equation (3.16) is the derivative of a state variable and, for the nth order case, n integration operations would be required in the corresponding simulation program.

For the more general case of a nonlinear model, the state-space representation would take the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\{\boldsymbol{x}(t), \boldsymbol{u}(t), t\} \tag{3.18}$$

$$\boldsymbol{y} = \boldsymbol{g}\{\boldsymbol{x}(t), \boldsymbol{u}(t), t\} \tag{3.19}$$

where $\boldsymbol{f}$ and $\boldsymbol{g}$ denote nonlinear functions.

For the electrical circuit model of Figure 3.7, an equivalent state-space representation involves the use of two first-order equations in place of the second-order reduced form representation (Equation (3.11)). These equations are based upon the *state variables* x_1 and x_2, where x_1 is the current $i(t)$ and x_2 is the voltage $v(t)$, and an input variable u which represents the input voltage $e(t)$ may be derived by rearranging Equations (3.8) and (3.9) to give:

$$\frac{di(t)}{dt} = \frac{1}{L}e(t) - \frac{R}{L}i(t) - \frac{1}{L}v(t) \tag{3.20}$$

$$\frac{dv(t)}{dt} = \frac{1}{C}i(t) - \frac{1}{Cr}v(t) \tag{3.21}$$

That is:

$$\dot{x}_1 = -\frac{R}{L}x_1 - \frac{1}{L}x_2 + \frac{1}{L}u(t) \tag{3.22}$$

$$\dot{x}_2 = \frac{1}{C}x_1 - \frac{1}{CR}x_2 \tag{3.23}$$

A further equation must be used to define the output variable of interest in terms of the state variables. In this case, the output is the voltage $v(t)$ so the output equation in this state-variable representation has the form:

$$y = x_2 \tag{3.24}$$

In matrix form, this state-variable model may be written as:

$$\begin{bmatrix} \dfrac{di(t)}{dt} \\ \dfrac{dv(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\dfrac{R}{L} & -\dfrac{1}{L} \\ \dfrac{1}{C} & -\dfrac{1}{CR} \end{bmatrix} \begin{bmatrix} i(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} \dfrac{1}{L} \\ 0 \end{bmatrix} e(t) \tag{3.25}$$

This is not the only possible choice of state variables for this problem, but it has the advantage of being physically meaningful and involves quantities that could be measured in the real physical system. This is an important consideration for model validation, as discussed in Chapter 7.

3.4.3 Differential algebraic equation (DAE) models

Differential algebraic equations have been mentioned briefly in Sections 3.4 and 3.4.1. The example discussed in Section 3.4.1 involving a simple second-order electrical circuit model (Figure 3.7) illustrates a situation in which a quantity which would normally be regarded as a model output is constrained to have a particular form and the model may be used to determine other variables of the model. This example of a differential algebraic equation is a special type of application which is closely related to inverse simulation, and situations of this kind are discussed further in Chapter 4. A more common situation in which differential algebraic equations are found in engineering applications arises when a dynamic system is subjected to physical constraints which are described through algebraic relationships.

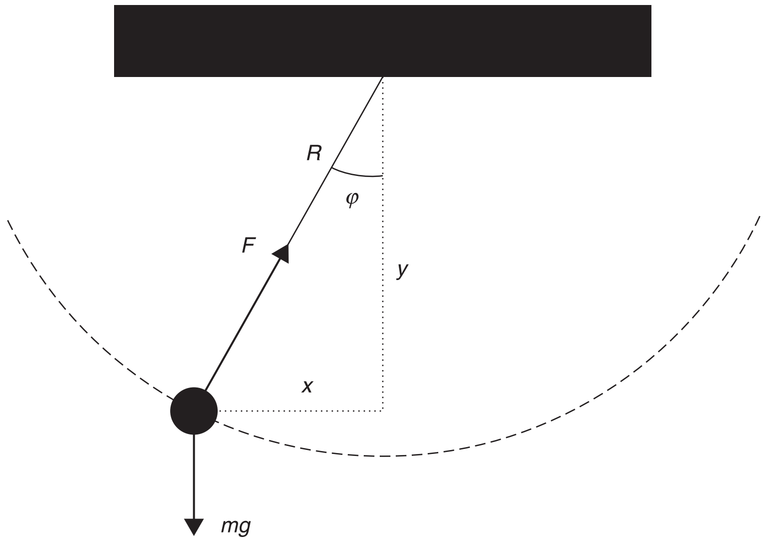


Figure 3.8 Schematic diagram of pendulum system

A very commonly used example (see e.g. [21]) involves the simple pendulum system shown in Figure 3.8. This can be modelled by applying Newton's Second Law to give the following set of equations:

$$M\frac{d^2x}{dt^2} = -F\sin\varphi = -F\frac{x}{R} \tag{3.26}$$

$$M\frac{d^2y}{dt^2} = -F\cos\varphi - mg = -F\frac{y}{R} - mg \tag{3.27}$$

$$x^2 + y^2 = R^2 \tag{3.28}$$

Equation (3.18) represents a geometric constraint ensuring that the position of the centre of gravity of the mass *M* lies on the circumference of a circle of radius *R*. This is an algebraic equation and involves no derivatives.

Equations (3.26) and (3.27) may each be rearranged as a pair of first-order equations to give:

$$M\dot{v}_x = -F\sin\varphi = -F\frac{x}{R} \tag{3.29}$$

$$M\dot{v}_y = -F\cos\varphi - mg = -F\frac{y}{R} - mg \tag{3.30}$$

$$\dot{x} = v_x \tag{3.31}$$

$$\dot{y} = v_y \tag{3.32}$$

Taken with Equation (3.28), these equations define the nonlinear model for the pendulum and a DAE model, such as this, can be simulated directly using any simulation tools incorporating facilities for handling DAEs (e.g. Modelica® [21]).

If a simulation model is developed using a block diagram, situations frequently arise involving 'implicit' or 'algebraic' loops that are characterised by closed pathways that do not include any integrator or pure delay blocks. This situation is clearly another form of DAE model and can be dealt with either by using specialist tools for the solution of DAEs or by eliminating the algebraic equation. That may be done by changing the DAE into an equivalent differential equation with a very short pure delay or by introducing a first-order lag element with a time constant that is very small compared with time constants of the system being modelled. Advocates of bond-graph methods claim that the improved understanding of causality that comes from that approach allows potential difficulties with algebraic loops to be identified at an early stage in the modelling process.

Simulation of a model in the form of a set of DAEs is based on implicit methods involving an iterative procedure that includes all the algebraic equations as well as ordinary differential equations of the model. Solution of a DAE may require differentiation as well as integration, since the algebraic equations may involve constraints between state variables but not necessarily directly between derivatives of those state variables (although they may be constrained implicitly). In order to make all the constraints explicit, some algebraic equations of a given model may have to be differentiated. The *differentiation index* of a DAE model is defined as the number of times that certain given equations of the model have to be differentiated in order to reduce the model to a set of ODEs (ordinary differential equations) that can be solved by conventional methods (i.e. through transformation into ODE explicit state-variable form). One widely used DAE solver is DASSL [33] which is used in Modelica$^{®}$ [21] and in other general-purpose simulation tools. DASSL uses a *backward difference formula* (BDF) approach and a number of variations of the basic solver have been produced.

One benefit of working with DAEs rather than ODEs is that there is no need to manipulate equations of the model into state-space form or to make use of sorting techniques to ensure that statements in a simulation model occur in the correct sequence. Also, of course, the problems with algebraic loops that occur with ODE-based methods cannot arise with DAE-based numerical solutions.

The Modelica® language makes comprehensive provision for discrete and hybrid modelling together with continuous system simulation, and discrete variables and discrete state transitions are allowed within the basic state-space type of model. This means that continuous DAE representations are not sufficient and Modelica® allows for *hybrid differential and algebraic equations* (hybrid DAEs). These are specific to the Modelica® language and are therefore not discussed further in this book, but interested readers can find detailed information in the text by Fritzson [21].

3.4.4 Transfer function descriptions

A linear model in reduced or state-space form can also be described using a *transfer function* representation. This is defined as the ratio of the Laplace transform of the chosen model output variable to the Laplace transform of the input for the case where all initial conditions are zero. This process (details of which may be found in introductory texts on system modelling, continuous system simulation or control engineering – see e.g. [30] and [34]) transforms the differential equation, expressed as a function of time, into an algebraic equation in the Laplace variable *s* and subsequent analysis is entirely algebraic.

For the electrical network example discussed in Section 3.4.1, Equation (3.11) transforms to:

$$LCs^2V(s)+\left(\frac{L}{r}+RC\right)sV(s)+\left(\frac{R}{r}+1\right)V(s)=E(s) \tag{3.33}$$

so that:

$$\frac{V(s)}{E(s)} = \frac{1}{LCs^2 + \left(\frac{L}{r} + RC\right)s + \left(\frac{R}{r} + 1\right)} \tag{3.34}$$

In the general case, a transfer function $G(s)$ relating an output $Y(s)$ to an input $U(s)$ may be written as:

$$\frac{Y(s)}{U(s)} = G(s) = \frac{A(s)}{B(s)} \tag{3.35}$$

where $A(s)$ and $B(s)$ are polynomials in s. Many important properties of the model depend upon the transfer function denominator, $B(s)$. Roots of the *characteristic equation* $B(s) = 0$ largely determine initial transients in the model response to any given input. These roots are the *poles* of the transfer function. Roots of the equation $A(s) = 0$ determine the *zeros* of the transfer function which also have an influence on the form of the transient behaviour.

Complex models can be built up using cascaded combinations of transfer functions of individual sub-models, provided appropriate care is taken in terms of possible loading and other interactions between adjacent blocks, which may not be fully considered in the sub-model development. Thus, with transfer function models, it is still important to consider all the implicit assumptions. Viewing a transfer function as an idealised mathematical block that can be manipulated without regard to the underlying physics is dangerous, especially when using modern block-oriented simulation tools.

A number of methods are available that allow transfer function descriptions to be expressed as a block diagram involving cascaded integrator blocks. These techniques also allow state-space representations to be developed from transfer function models. Details may be found in many elementary textbooks on system modelling, continuous system simulation and automatic control (see e.g. [30]).

3.4.5 Problems of stiffness

Whatever form of mathematical description is used, problems can arise when a model involves some responses that are very fast compared with others, so that the numerical methods being employed must take small steps to obtain satisfactory results but the overall behaviour of the model is relatively slow. For linear models this corresponds to a situation in which the roots of the characteristic equation cover a wide range but the phenomenon arises with nonlinear models where such analysis in terms of system poles is not applicable. Models of this kind are termed 'stiff' and the difficulty in simulating a stiff model is that the integration step must be small enough to suit the most rapidly changing component. This step size may, however, be unacceptably small in terms of the total time needed to include the slowest components. Stiffness is really an issue of computational efficiency and arises because of the need, in most applications, to keep the computation time to a minimum. Widely used variable-step integration algorithms may not be capable of reducing the computing time sufficiently for design applications and either the form of model being used has to be reconsidered or specialised algorithms may have to be selected.

Redefining the model using physical understanding of the problem and the intended application may well help to eliminate unnecessary features that are contributing to the overall stiffness. For example, some fast dynamic components (such as sensors in which dynamic effects are included initially) might be replaced by a static element involving simply a gain constant in the simulation model. Equally some long time constants might involve timescales outside the range of interest for the intended application and the model could be adapted so that these time constants could be considered as infinite. Situations in which problems of stiffness cannot be avoided through adaptations to the model may be handled effectively using specialised numerical integration algorithms such as the one developed by Gear, which is widely available in continuous simulation software tools [35].

3.4.6 Sub-models involving discrete events

Within the context of system modelling, an *event* is something that happens in an instantaneous fashion and is the result of an event *condition* that changes from *false* to *true*. At an event, a set of associated variables may be changed in some way and conditional equations may become active or may be deactivated by the occurrence of the event. By linking an event to an instant in time, we can order events to form an event history and we can define discrete-time variables that only change values at discrete points in time and keep their values constant between events.

Event-based stochastic system modelling is a separate area of study which is important in applications involving the operation of servers and queues, and for this a specialised modelling formalism has been developed which, as already mentioned, is known as the Discrete-Event System Specification (DEVS) [36]. Together with other tools, such as the Petri Net (see e.g. [21]), this DEVS approach (already mentioned in the context of quantised state systems) is particularly important in areas such as manufacturing or for systems involving packet-switched communications networks. Many good textbooks may be found that deal with these issues of discrete-event modelling (see e.g. [36]). Some modern system simulation tools, such as Modelica® [21], allow for both the DEVS and Petri Net formalisms.

Considering events that can arise in the modelling and simulation of continuous-time systems, examples include a continuous variable encountering some physical limit (such as a maximum mechanical displacement) or the sampling of a continuous variable through an analogue-to-digital converter. The first example is often associated with a discontinuity within an otherwise continuous simulation model, while the second arises in the modelling of sampled systems where digital processors are linked to external hardware.

Discontinuities

Discontinuities within continuous system simulation models can arise in two ways. The first of these is where there is more than one set of derivative functions for the model under consideration. Switching can occur instantaneously from one derivative function value to another when a particular variable reaches a threshold value. One very simple example of this is a simulation in which the input is a discontinuous function such as a square wave. The timing of this switching action has to be very precisely represented in the simulation, otherwise significant errors may arise. The second type of discontinuity arises when an instantaneous change in a state variable occurs. This could be, for example, due to the action of a valve within a simulation of a hydraulic system or a diode within an electrical circuit simulation. There are clear trade-offs between the treatment of discontinuities through specialised numerical methods that accurately determine the timing of a discontinuity and approaches involving more detailed physical modelling of the underlying processes. One disadvantage of the latter approach is that this might lead to stiffer models. Which approach is adopted depends very much upon the intended application of the model and the level of interest in the detailed mechanisms associated with the switching action.

Conditions for switching from one derivative function to another or for an instantaneous change in a state variable may be determined from a switch function, the zeros of which define the events of interest. This involves establishing first whether or not an event has occurred within the most recent integration interval and then, if it has occurred, determining the exact time of that event. In a continuous system simulation context, discontinuities can be handled using numerical tools devised specifically for the solution of this two-stage process involving the 'detection problem' and the subsequent 'location problem' (see e.g. [37], [38] and [39]).

Models of sampled systems

The growing importance of systems that involve both hardware and software elements means that simulation models must be capable of including both continuous variables and discrete-event dynamic elements. The modelling of digital processors and the associated analogue-to-digital and digital-to-analogue converters that allow discrete elements of this kind to communicate with the continuous elements within the system is important and is an essential part of an overall system model.

A very common example arises in the use of digital processors in automatic control system applications. This is illustrated in the block diagram of Figure 3.9 where the discrete elements involve a digital processor, interface units and the software used to implement the digital control algorithm for the closed-loop system. In modelling this system, the analogue-to-digital converter within the interface may be replaced by idealised elements in which the sampling

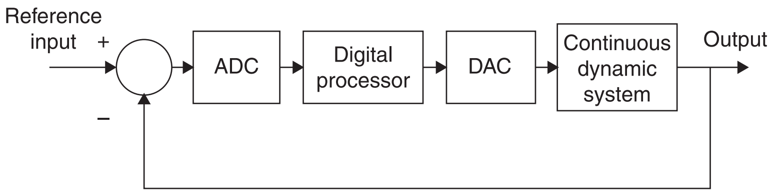


Figure 3.9 **Block diagram of sampled data system involving digital control of a continuous dynamic system**

process is represented by an ideal switch which closes for a very short period of time at regular intervals. The sampled value provides the discrete input to the control algorithm which is implemented on the digital processor using a software program. The numerical output from the digital processor is converted back to the form of a continuous variable by the digital-to-analogue converter and this may be modelled in an idealised way using a zero-order hold. This element gives an output which changes periodically but maintains a constant value between each output event.

The analogue-to-digital and digital-to-analogue converters are normally assumed to operate in a synchronous manner so that the input and output operations at the associated processor occur at the same time. Details of this approach to the modelling of digital processors and their interface units and the inherent assumptions may be found in many textbooks on simulation or automatic control (see e.g. [30] and [40]).

The algorithm within the digital processor of Figure 3.9 is normally modelled using a difference equation which allows the output at time $t = kT$ to be expressed in terms of the input at that time instant and by the input and output at previous sample times $t = (k\text{-}1)T$, $t = (k\text{-}2)T$, . . . etc. Here, T represents the sample period and the index k defines the sample number being considered. For example, a very simple algorithm involving proportional control would be modelled using a difference equation:

$$O(kT)=GE(k\mathrm{T}) \tag{3.36}$$

where $O(kT)$ is the signal at the digital-to-analogue converter and $E(kT)$ is the signal obtained from the analogue-to-digital converter at time $t = kT$. If, instead of proportional control, the control algorithm involved integral control so that the error signal was being integrated numerically within the control processor, the difference equation would have a more complex form and would involve values of the discrete variables O and E at previous sample times as well as the current values. Further discussion of the simulation of sampled-data systems and the more general issues that arise in the modelling and simulation of hybrid systems involving combinations of continuous and discrete dynamic elements may be found in recent textbooks and papers (see e.g. [22]).

3.5 Software for modelling and simulation

In discussing software for modelling and simulation, it is important to distinguish between software that is concerned primarily with simulation of lumped element dynamic systems and software tools that provide general tools for system modelling and analysis, and include some facilities for simulation. Since the CSSL specification was published in 1967 [41], many tools used for modelling and continuous system simulation have adhered, in part at least, to that standard (see e.g. [30] and [42]). In that approach, models are defined through assignment statements for variables and derivatives, and physically based equations have to be changed to a specific form that is suitable for calculations. In many packages the process has to be carried out largely by pencil and paper manipulation, although in many cases there is automatic sorting of the statements involved in calculating derivatives. One well-known and widely used example of a simulation 'language' of this kind is acslX [43] from AEgis Technologies.

In some cases simulation languages have facilities for models to be defined initially in terms of a block diagram or other convenient graphical representation. The complete model is translated into the CSSL form prior to execution.

Another widely used tool, which includes facilities for analysis as well as modelling and simulation of continuous systems, is MATLAB®/Simulink® [14]. Although it does not conform to the 1967 CSSL specification, it has facilities for defining models in terms of simple assignment statements and also for defining models in block diagram form.

One important development in recent years has been the introduction of equation-based object-oriented modelling languages such as Modelica® [21]. In contrast with other simulation tools, these use expressions that are essentially the same as the mathematical equations of the underlying model. Hierarchical decomposition and model reuse are also central to these developments. In addition, some languages have provision for the description of physical connections and can associate a set of variables with a port, thus allowing easy interconnection of sub-models.

Issues of combining discrete-event and continuous system simulation models were mentioned briefly in Section 3.3. In some cases the approach adopted has involved bringing together the necessary features to create a properly integrated tool that provides for discrete and continuous representation. Modelica, together with the associated simulation tools such as Dymola [10] or Opensource Modelica® [21], typifies this approach. The Stateflow™ toolbox [44] from MathWorks represents a different approach and allows the facilities of Simulink® to be extended to allow logic elements to be described, and this relates directly to previous discussion of events and the simulation of discrete-event systems in Section 3.4. MapleSim® [28] is another versatile physical modelling and simulation tool that includes many similar features, in this case developed on a foundation of symbolic computation technology.

Many simulation tools now commercially available have their origins in engineering research groups in universities. Examples include the simulation language ESL, developed initially at the University of Salford in England under contract to the European Space Agency but now marketed and supported commercially by ISIM International Simulation Ltd in the UK [45]. One important feature of ESL is the provision of powerful numerical tools for handling discontinuities. Another widely used simulation package which has origins in academia is 20-sim, maintained and marketed by the company Controllab Products BV, which has roots within Twente University in the Netherlands [12].

Another software tool that has its origin within an academic research environment is the Virtual Test Bed (VTB) which is being developed at the University of South Carolina in the USA [18]. This is a particularly interesting development in the context of integrated system applications and model-driven design involving multidisciplinary systems. The VTB can accept sub-models created using different simulation tools and this is important in creating efficiently a complete system model from elements that come from different teams. Each team is likely to have its preferred software tools and may be reluctant to move to some unfamiliar tool that would be common to all the teams involved in a multidisciplinary project. Adoption of tools such as the VTB permits an element of independence to be maintained while allowing an overall system model to be created quickly and efficiently. Model reuse may also be facilitated in that existing models based on the previously used tools may be integrated into new and more complex descriptions of the overall system.

Simulation tools such as acslX [43], MATLAB®/Simulink® [14], Modelica® [21], 20-sim [12], Maplesim® [28], ESL [45] and the VTB [18] are mentioned in this book in the context of specific applications. It should be noted that all these tools allow for graphical input, with models described by block diagrams or bond graphs. They also all support sub-models and thus allow some form of model reuse. They allow output visualisation through two-dimensional graphs and some, such as 20-sim, the VTB, Simulink®, Modelica® and ESL, provide access to three-dimensional animation tools. Examples that are presented in this book have mainly involved use of MATLAB®/Simulink® (or the open-source equivalent, SCilab/Xcos [46]), Modelica® and the VTB, and a brief introduction to the main features of these software packages is provided in the next two sub-sections. Readers who are unfamiliar with these tools may wish to consult some of the many textbooks available on the use of these packages.

3.5.1 A brief introduction to MATLAB®/Simulink®

MATLAB® (the MATrix LABoratory program) was developed to provide an easily accessible and interactive version of the powerful LINPACK [47] and EISPACK [48] routines which had been developed for solution of linear equations and eigenvalue problems. MATLAB® is now a general-purpose commercial matrix package that provides an interactive programming environment with graphical output. Every data object in MATLAB® is an array.

Although MATLAB® incorporates standard functions for the solution of ordinary differential equations, or can be used as a general-purpose programming language for simulation applications involving user-written algorithms, much present-day simulation work involving MATLAB® is based upon use of the Simulink® package. This provides a graphical input to allow models to be defined in terms of block diagrams and through this feature, models can be created using a hierarchical structure. This is helpful not only in the programming of complex simulation problems, but also in the efficient documentation of such models and to other users who need to understand a simulation in detail before applying it.

Data analysis and visualisation are both very straightforward using the facilities available in MATLAB®/ Simulink® and there are many specialist add-on tools, such as those for system identification, optimisation and signal processing. Symbolic computing results can be generated using one of the toolboxes and symbolic results can be integrated with MATLAB® and Simulink® results. Facilities for the generation of reports are made available through the MATLAB® Report Generator.

With the additional facilities provided by the Simscape™ package [49], the MATLAB®/Simulink® user has access to powerful facilities for the integrated modelling of systems involving a number of physical domains. Simulink® itself includes some standard library sub-models and through Simscape™ there are additional standard libraries involving sub-models for components in specialised fields, including mechanics, electronics and hydraulics. There are also libraries for mechanical transmission systems and for electrical power systems. Using the Simscape™ language, which is based on MATLAB®, sub-models can be created together with equivalent Simulink® blocks for new physical components that do not appear in the standard libraries. Similarly, it is relatively straightforward to create entirely new libraries using the facilities of Simscape™ and to extend existing libraries so that specialist models can be deployed across an organisation or to sub-contractors in large projects. Signals and parameters have units within the models in Simscape™ libraries and there are also facilities for the automatic conversion of units. The use of the Simscape™ environment is discussed further in Chapter 9.

The additional Stateflow® [44] tool available from the MathWorks extends the Simulink® environment to allow discrete-event situations to be included. Control, supervisory and mode logic can be incorporated into simulations in a natural fashion using this extension to Simulink®. Stateflow® charts can be created in a drag-and-drop fashion like Simulink® diagrams and facilities are included for hierarchical structures involving sub-charts.

Although MATLAB®/Simulink® is very widely used in industry, there is an open-source alternative to MATLAB®/ Simulink® known as Scilab/Xcos [46], which is now quite commonly found in industrial organisations and in universities. This was developed at the Institut Nationale de Récherche en Informatique et en Automatique (INRIA) in France. Scilab is similar in many respects to MATLAB® and the simulation tool Xcos provides simulation facilities similar to those of Simulink®. Versions of Scilab/Xcos are readily available by download for Linux and Windows® operating systems.

3.5.2 A brief introduction to Modelica®

The Modelica® standard has been under development continuously since 1997 [21] by an international non-profit association and this is a modern, object-oriented modelling language that is well suited to the solution of problems involving a number of different engineering domains. Modelica® models may be simulated in a number of ways, using the Dymola environment [10] or the OpenModelica tool.

It is important to note that, compared with many other modelling and simulation software products, Modelica® is a modelling language. It has been developed relatively recently and incorporates object-oriented features based on the principles of *encapsulation, inheritance* and *hierarchy*. It also builds upon the concept of non-causal modelling, with sub-model interfaces being defined through pairs of variables which are not assigned to be inputs or outputs at the outset.

The inheritance principle means that Modelica® is particularly well suited to supporting the reuse of models, since a generic sub-model can be defined in a broad fashion and then refined into a number of sub-models of which the common elements need be defined only once. Modelica® also has multi-domain modelling capabilities so that components from several different domains (e.g. mechanical, electrical, thermal, control systems) can be readily described in a unified fashion and combined easily.

Modelica® is based primarily on equations rather than the type of assignment statements that are commonly used in traditional programming languages and it is this feature that allows a-causal modelling with Modelica® and facilitates model reuse, since equations do not in themselves define a specific direction of data flow.

3.5.3 A brief introduction to the Virtual Test Bed (VTB)

The Virtual Test Bed (VTB) is a suite of software tools intended for use in the development of large-scale dynamic systems, usually by multidisciplined teams [18]. The facilities within the VTB are intended to assist in the process of developing and testing new designs prior to implementation as a hardware prototype or in a production system. The application area for which the VTB was initially developed was electrical power systems in the context of 'more electric' vehicles for use on land, in the air or in a marine environment. It has, however, been used for a variety of other types of application such as robotics.

Electric and hybrid vehicles of all types have provided an excellent application area for evaluation of new modelling and simulation software such as the VTB. Although involving electrical power systems, this application area differs significantly from conventional terrestrial electrical power generation and distribution systems since vehicle systems can involve distributed energy generation and storage, unconventional power sources such as fuel cells and gas turbines, much use of power electronics and distribution systems that may have to be reconfigured rapidly. The development of new vehicle systems of this kind is likely to involve sub-systems that cross conventional disciplinary lines to an even greater extent than in other types of projects and will certainly involve major elements of mechanical, electrical, power electronic, control and software engineering that have to be integrated very carefully within the final system. Inevitably, a number of different specialist design teams are likely to be involved and members of these teams must contribute fully and efficiently to the development of an interdisciplinary virtual prototype system. In many cases these teams will have preferred software tools and the VTB allows dynamic models from many environments to be brought together through the application of coupling laws involving signal coupling, data coupling or natural coupling principles. The VTB thus strives to provide a pathway to a fully inclusive teaming environment. These features of the VTB make it particularly interesting as a software environment for projects involving integrated systems and are considered further in Chapter 9.

The VTB software suite contains several tools. The first of these is the *Schematic Designer*, which is the central tool that is used in designing and simulating systems. It allows component models to be assembled into system models. The *Entity Designer* provides the means for developing VTB components, known as 'entities'. An entity is the simplest form of component and each entity has its own simulation engine. The Entity Designer allows use of a proprietary VTB modelling language and also permits the development of components using languages such as C++, C#, Visual Basic® and J#. The *Module Designer* is used to define and assemble new reusable models, known as modules, which are based on one or more existing components. A module is treated as a single component but does provide a simulation engine, as it makes use of the engines of the components within it.

It is important to note that the VTB includes an extensive library of simulation components. This has considerable value but the fact that the VTB allows import of sub-models developed in other simulation environments to be used in VTB simulations means that the range of model libraries and established sub-models available to VTB users is very large. Another feature of the VTB that should be mentioned is the fact that it provides a highly interactive environment in which the user can change the model topology or parameter values while the simulation is executing. High-level visualisation facilities can also be linked to live simulation data to allow animation of simulation results.

3.6 New developments in the modelling and simulation of micro- and nano-mechanical systems

As mentioned in Chapter 2, micro- and nano-electromechanical systems (MEMS and NEMS) technology is one field in which modelling and simulation methods are of central importance for the design of highly integrated systems. These can involve many types of model, including classical physical models, continuum mechanics models (e.g. for investigation of elasto-dynamic effects), thermal models, magnetic models and electrical models. Computational studies may involve not only three-dimensional finite element analysis, but also atomic-scale descriptions of materials using *molecular dynamics (MD)* simulation methods [50]. For many practical applications, there is therefore a need to reduce the computational complexity by reducing the number of variables and parameters in the model. This is especially important in applications involving the application of feedback principles where control design techniques have to be applied. In such cases, the use of model reduction methods becomes essential.

Issues such as the effects of thermal fluctuations on NEMS device performance leads to situations where physically based models are again essential and there is therefore a need to be able to move efficiently from highly detailed, physically based models to reduced models for control system design and then back to physically based models for further analysis. The capability to model effectively at a number of different resolutions is therefore vitally important and two categories of multi-scale simulation model are recognised in this application area [50]. The first of these involves *sequential multi-scale modelling* methods, in which large-scale models (such as those used for control system design) use low-resolution representations derived from more detailed, physically based, higher-resolution descriptions. The simulations at these different levels run independently of each other. The second approach is termed *concurrent multi-scale modelling* and attempts to link methods together in a combined model in which the different scales are considered concurrently. Developments of this kind are clearly applicable in other fields and have considerable relevance for all involved in the use of simulation techniques in designing highly integrated systems, whatever the area of application.

3.7 References

[1] Kuipers, B. (1989) ‘Qualitative reasoning, modelling and simulation with incomplete knowledge’, *Automatica*, Vol. 25, pp. 571–85.

[2] Birkett, S., Thoma, J. and Roe, P. (2006) ‘A pedagogical analysis of bond graph and linear graph physical system models’, *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 12, No. 2–3, pp. 107–25.

[3] Paynter, H.M. (1961) *Analysis and Design of Engineering Systems*, MIT Press, Cambridge MA, USA.

[4] Karnopp, D.C., Margolis, D.L. and Rosenberg, R.C. (2000) *System Dynamics – Modeling and Simulation of Mechatonic Systems*, Third Edition, Wiley, New York NY, USA.

[5] Breedveld, P. (ed.) (1991) *J. Franklin Institute*, Vol. 328 (special issue on current topics in bond-graph related research).

[6] Gawthrop, P. and Scavarda, S. (eds) (2002) *Proc. Inst. Mechanical Engineers, Journal of Systems and Control Engineering, Proceedings Part 1*, Vol. 216 (special issue on bond graphs).

[7] Borutzky, W. and Gawthrop, P. (2006) 'Bond graph modelling', *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 12, No. 2–3 (special issue on bond graph modelling).

[8] Gawthrop, P.J. and Bevan, G.P. (2007) 'Bond-graph modeling', *IEEE Control Systems* Magazine, Vol. 27, No. 2, pp. 24–45.

[9] Broenink, J.F. (1999) 'Introduction to physical systems modelling with bond graphs', Technical Report, University of Twente, Enschede, the Netherlands (online): *www.ce.utwente.nl/bnk/papers/BondGraphs V2.pdf* (accessed 3 July 2011).

[10] Elmquist, H., Bruck, D. and Otter, M. (1996) *Dymola – User's Manual*, Dynamsim AB, Ideon SE-223 70, Lund, Sweden.

[11] Granda, J. (1990) *CAMP-G Users Manual*, Mitchell and Gauthier Associates, Concord MA, USA.

[12] 20-sim, modelling and simulation package, Control-Lab Products BV, Enschede, the Netherlands (online): *www.20sim.com* (accessed 30 June 2011).

[13] COMSOL Multiphysics® software, COMSOL Group, Sweden (online): *www.uk.comsol.com* (accessed 30 August 2011).

[14] MATLAB®/Simulink®, modelling and simulation software, Mathworks Inc, USA (online): *www. mathworks.com/products* (accessed 30 June 2011).

[15] SolidWorks®, CAD software, Solid Solutions Management Ltd, UK (online): *www.solidsolutions.co.uk* (accessed 15 September 2011).

[16] Autodesk®Inventor®, CAD software, Autodesk Inc, USA (online): *http://usa.autodesk.com/autodesk-inventor* (accessed 15 September 2011).

[17] Creo Parametric® (formerly ProENGINEER®), CAD software, Parametric Technology Corp, USA (online): *www.ptc.com/products/index.htm* (accessed 15 September 2011).

[18] VTB modelling and simulation package, the University of South Carolina, *The Virtual Test Bed* (online): *http://vtb.engr.sc.edu/vtbwebsite* (accessed 30 June 2011).

[19] Wu, B., Dougal, R. and White, R.E. (2001) 'Resistive companion battery modelling for electric circuit simulation', *Journal of Power Sources*, Vol. 93, pp. 186–200.

[20] Christen, E. and Bakalar, K. (1999) 'VHDL-AMS – a hardware description language for analog and mixed-signal applications', *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 46, No. 10, pp. 1263–72.

[21] Fritzson, P. (2004) *Principles of Object-oriented Modeling and Simulation with Modelica 2.1*, IEEE Press, Piscataway NJ, USA.

[22] van der Schaft, A. and Schumacher, H. (1999) 'An introduction to hybrid dynamical systems', *Lecture Notes in Control and Information Sciences*, Vol. 251, pp. v–vii.

[23] Bemporad, A. and Morari, M. (1999) 'Control of systems incorporating logic, dynamics and constraints', *Automatica*, Vol. 35, No. 3, pp. 407–27.

[24] Zeigler, B.P. and Lee, J.S. (1998) 'Theory of quantized systems: formal basis for DEVS/HLA distributed

simulation environment', *SPIE Proceedings*, Vol. 3369, pp. 49–58.

[25] Zeigler, B.P. (1976) *Theory of Modeling and Simulation*, Wiley, New York NY, USA.

[26] Kofman, E. and Junco, S. (2001) 'Quantised state systems: A DEVS approach for continuous system simulation', *Transactions of Society for Computer Simulation*, Vol. 18, No. 3 pp. 123–32.

[27] ModelicaDEVS library, Modelica Association, Sweden (online): *www.modelica.org/libraries/ModelicaDEVS* (accessed 15 September 2011).

[28] Maple™ software; Maplesim™ modelling and simulation software: Maplesoft, a division of Waterloo Maple Inc (online): *www.maplesoft.com/products/maple* and *www.maplesoft.com/products/maplesim* (accessed 3 July 2011).

[29] *Mathematica*® software, Wolfram Research, USA (online): *www.wolfram.com/mathematica* (accessed 3 July 2011).

[30] Murray-Smith, D.J. (1995) *Continuous System Simulation*, Chapman and Hall, London, UK.

[31] Bennett, B.S. (1995) *Simulation Fundamentals*, Prentice-Hall, Hemel Hempstead, UK.

[32] Matko, D., Karba, R. and Zupančič, B. (1992) *Simulation and Modelling of Continuous Systems*, Prentice-Hall, Hemel Hempstead, UK.

[33] Breman, K.E., Campbell, S.L. and Petzold, L.R. (1989) *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Elsevier, New York NY, USA.

[34] Close, C.M. and Frederick, D.K. (1995) *Modeling and Analysis of Dynamic Systems*, Wiley, New York NY, USA.

[35] Gear, C.W. (1971) *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice Hall, Englewood Cliffs NJ, USA.

[36] Zeigler, B.P., Praehofer, H. and Lim, T.G. (2000) *Theory of Modeling and Simulation*, Second Edition, Academic Press, San Diego CA, USA.

[37] Carver, M.B. (1978) 'Efficient integration over discontinuities in ordinary differential equation simulations', *Mathematics and Computers in Simulation*, Vol. XX, pp. 190–6.

[38] Gear, C.W. (1984) 'Efficient step size control for output and discontinuities', *Trans. Society for Computer Simulation*, Vol. 1, pp. 27–31.

[39] Birta, L.G., Oren, T.I. and Kettinis, D.L. (1985) 'A robust procedure for discontinuity handling', *Transactions of the Society for Computer Simulation*, Vol. 2, No. 3, pp. 185–205.

[40] Leigh, J.R. (1985) *Applied Digital Control*, Prentice-Hall, Englewood Cliffs NJ, USA.

[41] SCi Software Committee (1967) 'The SCi Continuous-System Simulation Language', *Simulation*, Vol. 9, No. 6, pp. 281–3.

[42] Cellier, F.E. (1991) *Continuous System Modelling*, Springer, Berlin, Germany.

[43] acslX modelling and simulation software, AEgis Technologies Inc, USA (online): *www.acslx.com* (accessed 3 July 2011).

[44] Stateflow® software, Mathworks Inc, USA (online): *www.mathworks.com/products* (accessed 30 June 2011).

[45] ESL modelling and simulation package, ISIM International Simulation Ltd, UK (online): *www.isimsimulation.com* (accessed 30 June 2011).

[46] Campbell, S.L., Chancelier, J.-P. and Nikoukhah, R. (2000) *Modeling and Simlation in Scilab/Scicos*, Springer, NewYork NY, USA.

[47] Dongarra, J.J., Bunch, J., Moler, C.B. and Stewart, G. (1979) *LINPACK User's Guide*, SIAM, Philadelphia PA, USA.

[48] Smith, B.T., Boyle, J.M., Dongarra, J.J., Garbow, B.S., Ikebe, Y. *et al.* (1976) *Matrix Eigensystem Routines: EISPACK Guide*, Lecture Notes in Computer Science, Vol. 6, Second Edition, Springer, Berlin, Germany.

[49] Simscape^{TM} software, Mathworks Inc, USA (online): *www.mathworks.com/products* (accessed 30 June 2011).

[50] Ferreira, A. and Aphale, S.S. (2011) 'A survey of modelling and control techniques for micro- and nanoelectromechanical systems', *IEEE Trans. on Systems, Man and Cybernetics, – Part C: Applications and Reviews*, Vol. 41, No. 3, pp. 350–64.

4

Inverse simulation for system modelling and design

Abstract: Inverse simulation is a tool for finding inputs such that model outputs match predetermined time histories. This could, for example, be an aircraft manoeuvre or the movement of a robotic arm. Through the inverse solution model, behaviour can be investigated in a way that is different from conventional simulation methods which provide model outputs for given initial conditions and input time histories. Several techniques are available for inverse simulation and methods described include a so-called 'differentiation' approach, an 'integration' approach and an approach based on feedback system principles. Case studies are used in discussing applications of inverse simulation and these involve a process system with two coupled tanks of liquid, a dynamic model of an unmanned underwater vehicle and an aircraft model. Emphasis is given to the feedback system approach in these case studies.

Key words: inverse, differentiation, integration, feedback.

4.1 An introduction to inverse modelling and inverse simulation

An *inverse dynamic model* allows time histories of input variables to be found that give model output responses that match given time histories. In fields such as environmental science, 'inverse modelling' describes the process of fitting a model to measurements or to field observations (essentially the process of system identification and parameter estimation), but that is not the meaning used here.

The significance of inverse modelling can be understood from an example. If a manoeuvre is defined for an underwater vehicle as a series of positions in three dimensions (in an earth-based axis system) and times, inverse modelling techniques provide the required time history of thrust values and control surface movements to allow that manoeuvre to be performed. If, for given maximum thrust values and given actuator characteristics, the requirements of this manoeuvre are not met, the inverse model may also provide information to facilitate design changes.

Although inverse models have relevance for any problem, the advantages have been found to be particularly significant for systems involving a human operator, especially if the operator and the system being controlled interact closely. Examples include piloting of fixed-wing aircraft and helicopters, crane operation, ship steering and other similar man-machine control tasks. Military fixed-wing aircraft and helicopter applications stimulated much early research since inverse methods can provide additional insight concerning vehicle-handling qualities, which is an important topic.

Although inversion techniques for linear models are of limited importance for practical applications, the structure of linear inverse models can provide useful insight regarding limitations of inverse solutions. For example, for single-input single-output (SISO) models, a transfer function can be inverted directly, provided we ensure that the inverse model is realisable or 'proper'. In other words, since poles and zeros are interchanged in the inversion process, additional factors may be needed in the denominator of the inverse so that the number of poles is at least as great as the number of zeros. The additional poles must lie at points in the s-plane that are far from the poles and zeros of the given model.

For multi-input multi-output (MIMO) linear models, inversion is also possible through simple analytical techniques, but practical difficulties arise for some model structures. Details of methods of inversion for linear MIMO models may be found in the published work of Brockett [1], Dorato [2] and Hirschorn [3].

Analytical techniques of model inversion have been developed further by Isidori and his co-workers and others (see e.g. [4] and [5]). Nonlinear models require transformation to linear and controllable descriptions using nonlinear state feedback (see e.g. [6], [7] and [8]) together with concepts from differential geometry that are unfamiliar to most engineers. Other mathematical techniques, such as regularisation, are also relevant. However, although successful in applications involving automatic control, these methods have not been used more widely in design.

The equations of a nonlinear dynamic system for most cases of practical importance may be written in the form:

$$\dot{x} = f(x, u) \tag{4.1}$$

$$y = g(x) \tag{4.2}$$

where x and u represent the vector of state variables and input variables, respectively, and y represents the vector of output variables. In Equation (4.2) the output variables are functions of the state variables only and there is no direct coupling of input variables to the outputs. This is representative of most real systems, since changes of input seldom produce instantaneous output changes.

The essence of the inverse modelling problem is to find the input vector u^* that will produce an output $y(t)$ that exactly matches a required output $y^*(t)$.

Differentiating Equation (4.2) with respect to time gives:

$$\dot{y} = g_x(x)\dot{x} \tag{4.3}$$

and substitution in Equation (4.1) then gives:

$$\dot{y} = g_x(x)f(x, u) \tag{4.4}$$

If this equation is solvable for u, it gives u^* from the solution of the following equation:

$$\dot{y}^* = g_x(x)f(x, u^*) \tag{4.5}$$

This may be possible for some functions $g(x)$, but further differentiation is usually needed and the additional equation:

$$\ddot{y}^* = g_{xx}(x)f(x, u^*)^2 + g_x(x)(f_x(x, u^*)f(x, u^*) + f_u(x, u^*)\dot{u}^* \tag{4.6}$$

must be solved, together with Equation (4.5), for u^* in terms of x and derivatives of y^*. If this is still impossible, Equation (4.6) must be differentiated repeatedly until a solution is obtained.

The number of differentiation operations needed is termed the *relative degree* of the system [4]. The relative degree does not exist if the equations cannot be solved, regardless of how many differentiations are performed.

Many practical system models have a form that is linear in the control inputs:

$$\dot{x} = f_0(x) + f_1(x)u \tag{4.7}$$

This has been applied widely and is appropriate for vehicle models, including aircraft. Algebraic manipulation can provide straightforward solutions, as demonstrated by Bradley for simplified helicopter models [9].

One of the simplest situations involving hard limits arises in the modelling of nonlinear actuators, where a simple and widely used actuator model involves a saturation element in association with a simple first-order lag. Whether or not a given actuator becomes saturated or reaches its rate limit for a given demanded output response can be an important design issue.

4.2 Methods of inverse simulation

Inverse simulation methods avoid analytical complexities in the inversion of nonlinear models, just as conventional simulation methods do for traditional input-output modelling in the nonlinear case [see e.g. [10], [11] and [12]).

Aeronautical engineers have been using inverse simulation since the late 1980s. The techniques are also useful for mechatronic systems and other integrated control system applications, since they provide insight about actuators, such as effects of amplitude and rate limits, that is otherwise more difficult to obtain.

Available techniques may be divided into five categories:

1. differentiation methods as developed by Kato and Saguira [13] and by Thomson and Bradley at the University of Glasgow (e.g. [14] and [15]);
2. integration methods which originated with the work of Hess and his colleagues at the University of California, Davis (e.g. [16]) and, independently, by Thomson *et al.* (e.g. [17]);
3. methods which use search-based or evolutionary optimisation algorithms in conjunction with the integration-based approach (e.g. [18]), together with methods based on traditional optimisation algorithms (e.g. [19], [20] and [21]);

4. methods based on the numerical solution of differential algebraic equations; and
5. methods based on the application of feedback system principles (see e.g. [22]).

4.2.1 The differentiation-based approach

Inverse simulation methods based on numerical differentiation of state variables were quickly adopted for a range of aeronautical applications and especially for helicopter flight mechanics and handling qualities investigations, as reported by Thomson and Bradley (e.g. [23] and [24]). Liceago-Castro [25] provided an alternative approach using symbolic computing methods.

When Thomson and Bradley began applying their differentiation method to helicopter flight mechanics ([15], [23] and [24]), Kato and Suguira [13] were using a similar approach for fixed-wing aircraft problems. In both cases the algorithm involved discretisation of the flight path in terms of time and the use of numerical differentiation to find angular rates. The equations of motion were then solved as a set of algebraic equations at each time step. In this approach, equations for the inverse simulation cannot be expressed as a set of first-order ordinary differential equations and the inverse simulation therefore cannot be separated easily from the forward simulation model. Any changes in the model that forms the forward description leads to major changes in the structure of the inverse simulation.

Although it has limitations, as discussed above, this approach was adopted by others and applied to practical problems. One result was an improved understanding of the properties of the constrained system produced by inversion. The dynamic characteristics of the inverse simulation model differ greatly from those of the forward model, since the dynamics of the inverse involve the zero dynamics of the system (see e.g. [5], [12] and [18]).

4.2.2 The integration-based approach

In 1991 Hess *et al.* [16] described an approach that avoided the inflexibility inherent in the differentiation method. This method is based on numerical integration and involves discretisation of the required manoeuvre, as in the differentiation method. In the integration-based approach an estimate is then made, at each time point, of the amplitude of the step displacement needed for each input to move the vehicle to the next point in the manoeuvre time history. The new position is calculated and the error between the actual and required position is found. An iterative procedure then minimises the error and the time history of inputs needed to move the vehicle to the required position is found.

Although computationally more demanding than the differentiation method, this approach has some advantages. There is flexibility in terms of the form of the model and reorganisation of the program is not required when small changes are made in the model structure. Thus any conventional forward model can be incorporated within the iterative loop to provide an inverse simulation. In recent years this has become the most widely used approach to inverse simulation (see e.g. [5], [11] and [12]).

The fundamental assumption of the integration-based method is that inputs are constant over a time interval T which is significantly larger than the integration step time. For the interval T (starting at time $t = 0$), the inverse simulation problem then involves finding a constant input u^* such that:

$$y^*(T) = g(x(T)) \tag{4.8}$$

where:

$$x(T) = x(0) + \int_0^T f(x, u^*)dt \tag{4.9}$$

The solution u^* may be found using Newton's method or some similar algorithm. The variation of the output with u^* at time T may be written as:

$$y_u^* (T) = g_x\, x_u\, (T) = g_x \int_0^T f_u\, dt \tag{4.10}$$

and numerical differentiation can be used to estimate y_u^*. The time step T is then repeated over the whole period under investigation, with the output $y(t)$ matching the required output $y^*(t)$ at multiples of that time step.

Other methods have involved variations of the integration-based approach. For example, *sensitivity function* ideas [26] have been applied successfully, showing advantages over the traditional integration-based method, especially in terms of improved accuracy in calculation of the Jacobian matrix. Further variations involve the use of *direct-search optimisation* algorithms [18] and other optimisation methods (see e.g. [21]), as discussed further in Section 4.2.3.

A *two time-scale approach* to inverse simulation was developed by Avanzini and de Matteis (e.g. [27] and [28]). It involves partitioning the state variables into two sub-vectors on physical grounds and reduces the model so that a relatively large time step may be used, reducing the computer time. Thomson and Bradley [12] have also described the use of this approach with a helicopter model.

4.2.3 Methods involving search-based routines and other optimisation algorithms

The established methods of inverse simulation, based on the differentiation and integration-based approaches, introduce derivative information and involve calculation of Jacobian or Hessian matrices. Being derivative-free, optimisation methods involving direct search algorithms avoid any requirement to determine elements of the Jacobian matrix or Hessian matrix and therefore can alleviate problems caused by discontinuities and input saturation effects. As mentioned in Section 4.2.2, the integration-based approach has been used successfully with search-based optimisation techniques and a number of other optimisation algorithms.

The 2008 paper by Lu *et al.* [18] provides details of a derivative-free method of inverse simulation, involving a version of the downhill simplex optimisation method of Nelder and Mead [29]. This Nelder-Mead (NM) approach involves minimisation of a scalar-valued nonlinear function of real variables using function values only and avoiding explicit or implicit use of gradient information. Examples of the successful application of a pattern-search based approach may be found in [18] where a method of inverse simulation based on the constrained Nelder-Mead algorithm is described in detail. Applications to nonlinear ship models are discussed and it is shown that problems of convergence that can arise with the gradient-based Newton-Raphson (NR) type of approach are avoided using the derivative-free constrained Nelder-Mead algorithm.

Two case studies are considered in the work of Lu *et al.* [18] and the first of these involves a relatively simple single-input single-output ship model which includes rudder amplitude and rate limiting. Although results obtained by inverse simulation using the gradient-based Newton-Raphson approach and the constrained NM method agree well for ship manoeuvres that involve rudder angles and rates that are below the limits, the NM method also achieves good convergence and physically meaningful inverse simulation results for cases where the NR algorithm fails to converge.

The second case study involves a nonlinear container ship model with two inputs and three output variables. Turning circle and pull-out manoeuvres were considered and, once again, the NM method was successful in situations involving manoeuvre discontinuities for which the NR algorithm failed to converge.

It was concluded from the results of these case studies that the derivative-free procedure based on the constrained NM algorithm could provide important benefits in terms of convergence and numerical stability compared with the more conventional approach involving the NR algorithm. These benefits were significant in cases involving input saturation or discontinuous manoeuvres.

Methods of inverse simulation involving other forms of optimisation algorithm may be found in the work of de Matteis *et al.* [19], Lee and Kim [20], and Celi [21]. Those three examples all involve aircraft and helicopter applications but the optimisation algorithms applied are general in form and could readily be applied in other areas.

4.2.4 Inverse simulation through numerical solution of differential algebraic equations

Models based on differential algebraic equations (DAEs) are discussed briefly in Section 3.4.1 where it was pointed out that using the DAE form of description, a model variable can be constrained to have a particular algebraic form, usually expressed as a function of time. In that section an example was presented using the simple RLC electrical circuit of Figure 4.1.

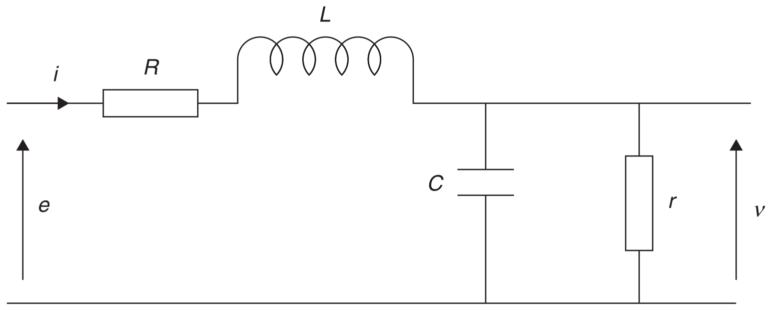


Figure 4.1 Electrical circuit example involving inductance *L*, capacitance *C* and two resistors (*R* and *r*)

The voltages $e(t)$ and $v(t)$ and the current $i(t)$ are then related according to Kirchhoff's voltage and current laws by the following equations:

$$e(t) = Ri(t) + L\frac{di(t)}{dt} + v(t) \tag{4.11}$$

$$i(t) = C\frac{dv(t)}{dt} + \frac{1}{r}v(t) \tag{4.12}$$

These equations may be combined with the algebraic equation:

$$v(t) = kf(t) \tag{4.13}$$

where $f(t)$ is some required function of time such as a ramp or sinusoid or triangular wave to form a model in which the variables $i(t)$ and $e(t)$ are unknowns, and $v(t)$ is constrained.

$$L\frac{di(t)}{dt} = e(t) - Ri(t) - v(t) \tag{4.14}$$

$$C\frac{dv(t)}{dt} = i(t) - \frac{1}{r}v(t) \tag{4.15}$$

$$v(t) = kf(t) \tag{4.16}$$

Equations (4.14), (4.15) and (4.16) constitute a DAE and solution of the model equations should allow us to find the form of the voltage $e(t)$ and current $i(t)$ to achieve the required output $v(t)$. This is an inverse problem and it is clear that simulation tools, such as Modelica® [30], that have facilities for the solution of DAEs provide an immediate basis for inverse simulation.

The use of Modelica® [30] for inverse simulation based on the solution of DAEs has produced some potentially interesting results (see e.g. [31]). Although such simulation tools can, in effect, generate inverse simulation models automatically, it has been pointed out [31] that, as with other methods for inverse simulation, the inversion process does not always lead to the expected results. Issues of stability can arise and for a general form of DAE, no stability proof is available and many simulations may have to be performed to check whether or not a nonlinear inverse simulation is stable in the region of the operating space of interest for the application. As with other methods of model inversion, additional analysis, based usually on a linearised version of the model, is essential before inverse simulation is attempted using DAE solution methods.

4.2.5 A feedback systems approach

Feedback principles have for long been used to generate inverse functions on analogue computers, as described in the simulation literature between the 1950s and 1980s. For example, feedback pathways applied to analogue multiplier hardware allow these units to carry out division. Similarly, feedback can also allow inverse functions to be found from conventional analogue function generators (see e.g. [32] and [33]). Although some problems of stability and performance are reported with these feedback methods, especially for high speeds of solution, they have been used with great success on general-purpose analogue computers. The same principles can be applied more generally for inverse simulation and provide a different and potentially very fast approach to inverse simulation of linear and nonlinear systems.

The single-input single-output case

The use of feedback to generate inverse solutions is based on properties of closed-loop systems. For a linear time-invariant SISO model with transfer function $G(s)$, the block diagram of Figure 4.2 illustrates the principles. The reference signal v is the time history to be followed by the model output and the signal w is the input to the model for that required output. Hence if v is given, the quantity w represents the

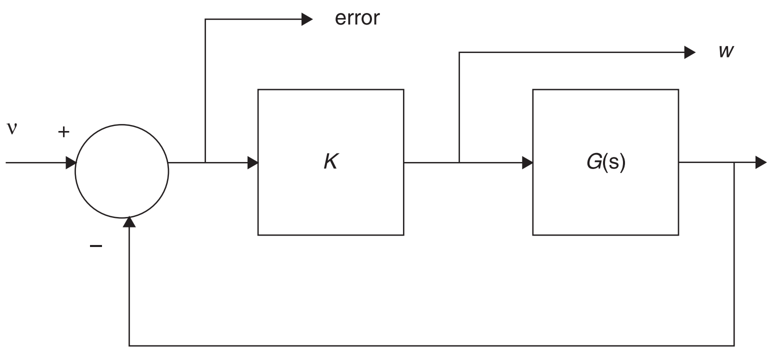


Figure 4.2 Block diagram for feedback solution to model inversion problem. Here the variable *w* obtained from simulation of the system with feedback is the input to the model *G* required to produce a model output equal to the reference signal *v*. Although the single-input case is presented in this diagram, the approach is also applicable to multi-input situations

inverse solution for the model G, provided the difference between the reference v and the output from the model is sufficiently small.

This approach can be demonstrated using simple linear analysis based on Laplace transforms, as in linear control theory. For the SISO case, with a simple gain factor K in cascade with the model $G(s)$, the transfer function relating the variable $W(s)$ to the reference input $V(s)$ in Figure 4.2 is given by:

$$\frac{W(s)}{V(s)} = \frac{K}{1 + KG(s)} \tag{4.17}$$

$$= \frac{1}{\frac{1}{K} + G(s)} \tag{4.18}$$

For large values of K, this gives:

$$\frac{W(s)}{V(s)} \approx \frac{1}{G(s)} \tag{4.19}$$

Thus the inverse model for G(s) may be found by applying high-gain feedback and taking the input variable for $G(s)$ as the output of the inverse model. The variable V is therefore the output required from the model while W is the input to the model (under open-loop conditions) that will produce that output. Note that the controller block K in Figure 4.2 may, in the general case, include dynamic elements in addition to a gain factor (see e.g. [22]). Note also that the orders of numerator and denominator of the closed-loop transfer function are the same as the order of the denominator of $G(s)$. The number of poles of the inverse model is thus always the same as the number of zeros and issues of realisability due to an excess of zeros do not arise [34]. Root locus analysis of the closed-loop system allows one to ensure that the additional poles of the inverse model introduced through feedback lie far from the other poles and zeros of the inverse.

The multi-input multi-output case

The block diagram of Figure 4.2 can be extended to cover the case of MIMO models involving standard state-space descriptions (see e.g. [34] and [35]). In this case the block ***G*** is replaced by a standard linear state-space representation of the system model as shown in Figure 4.3 and we add feedback through a controller block ***K***.

Then, the output ***w*** of the closed-loop system is given by:

$$\boldsymbol{w} = \boldsymbol{K}(\boldsymbol{v} - (\boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{w})) \tag{4.20}$$

which gives:

$$(\boldsymbol{I} + \boldsymbol{K}\boldsymbol{D})\boldsymbol{w} = \boldsymbol{K}\boldsymbol{v} - \boldsymbol{K}\boldsymbol{C}\boldsymbol{x} \tag{4.21}$$

so that:

$$\mathbf{w} = -(\mathbf{I} + \mathbf{KD})^{-1}\,\mathbf{KCx} + (\mathbf{I} + \mathbf{KD})^{-1}\,\mathbf{Kv} \tag{4.22}$$

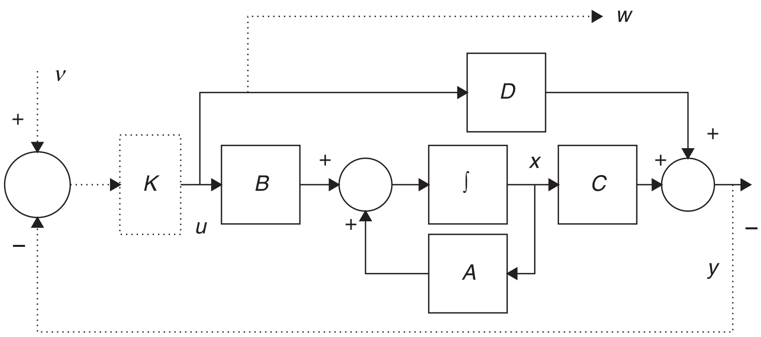


Figure 4.3 **Block diagram illustrating the use of feedback principles for a multi-input, multi-output system model in standard state-space format**

In terms of the state variables $\boldsymbol{x}$, we then have:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{w} \tag{4.23}$$

$$= \mathbf{A}\mathbf{x} + \mathbf{B}(-(\mathbf{I} + \mathbf{K}\mathbf{D})^{-1}\,\mathbf{K}\mathbf{C}\mathbf{x} + (\mathbf{I} + \mathbf{K}\mathbf{D})^{-1}\,\mathbf{K}\mathbf{v}) \tag{4.24}$$

$$= (\mathbf{A} - \mathbf{B}(\mathbf{I} + \mathbf{K}\mathbf{D})^{-1}\,\mathbf{K}\mathbf{C})\mathbf{x} + \mathbf{B}(\mathbf{I} + \mathbf{K}\mathbf{D})^{-1}\,\mathbf{K}\mathbf{v} \tag{4.25}$$

So that the inverse model has a system matrix:

$$\boldsymbol{A}^* = (\mathbf{A} - \mathbf{B}(\mathbf{I} + \mathbf{K}\mathbf{D})^{-1}\,\mathbf{K}\mathbf{C}) \tag{4.26}$$

$$\text{an input matrix } \boldsymbol{B}^* = \mathbf{B}(\mathbf{I} + \mathbf{K}\mathbf{D})^{-1}\,\mathbf{K} \tag{4.27}$$

$$\text{an output matrix } \boldsymbol{C}^* = -(\mathbf{I} + \mathbf{K}\mathbf{D})^{-1}\,\mathbf{K}\mathbf{C} \tag{4.28}$$

$$\text{and } \boldsymbol{D}^* = (\mathbf{I} + \mathbf{K}\mathbf{D})^{-1}\,\mathbf{K} \tag{4.29}$$

When the diagonal elements of ***KD*** are much greater than one in Equations (4.26) to (4.29), the elements of the identity matrix ***I*** can be neglected and it is possible to show that, as the elements of ***K*** tend to infinity:

$$\begin{aligned}\lim(\mathbf{I} + \mathbf{K}\mathbf{D})^{-1}\,\mathbf{K} &= \lim(\mathbf{K}\mathbf{D})^{-1}\,\mathbf{K} \\ &= \lim \boldsymbol{D}^{-1}\boldsymbol{K}^{-1}\,\boldsymbol{K} = \boldsymbol{D}^{-1}\end{aligned} \tag{4.30}$$

This approach can therefore be employed in cases where no inverse of the matrix ***D*** exists. It should also be noted that for ***D*** = **0**, the inverse model simplifies to:

$$\boldsymbol{A}^* = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{C}) \tag{4.31}$$

$$\boldsymbol{B}^* = \boldsymbol{B}\boldsymbol{K} \tag{4.32}$$

$$\boldsymbol{C}^* = -\boldsymbol{K}\boldsymbol{C} \tag{4.33}$$

$$\boldsymbol{D}^* = \boldsymbol{K} \tag{4.34}$$

This is important since the feedback approach is then applicable to linear models in which the block diagram has no direct pathways from the input to the output variables, which is very common in practice.

Selection of elements of the K matrix may present difficulties since the feedback system must be stable. Thinking in terms of SISO closed-loop systems, some closed-loop poles move towards the positions of open-loop zeros as the loop gain is increased, but there are additional closed-loop poles that may move into undesirable areas of the s-plane as the gain is increased. It is important to understand how these closed-loop poles behave and to limit their movement. This may require use of other techniques such as root locus analysis and pole-placement design.

A general procedure for inverse simulation through feedback system design

Inverse simulation procedures using feedback involve two distinct stages. The first of these is the design of the feedback system, while the second stage involves the implementation of the feedback system to provide the inverse simulation.

In the first stage the closed-loop system is designed, by any appropriate method, and feedback parameters are adjusted so that it meets some given requirements for simple inputs such as a step or impulse. The required output is used as reference input for this closed-loop system and the inverse solution can be found from the signal at the input to the model. This is, in control terms, a 'tracking problem' for which there are many possible solutions.

Although linear theory has been used here to introduce the problem, tracking principles apply equally to nonlinear models. If a linearised description is used in the initial design, the performance of the closed-loop system must be checked through simulation for several operating conditions, preferably using the fully nonlinear model. Limitations must be established and noted for the application stage.

In the second stage, the feedback system is used to ensure that the model output follows a given time history that may be generated by a reference model. The difference between the reference input (the time history to be followed by the model output variable) and the actual model output should be monitored continuously. A measure of this difference (such as the integral of the squared error) may be recorded along with the time history of the model input (which is the required output of the inverse simulation), as shown in Figure 4.2.

Although the feedback approach to inverse simulation is computationally efficient and potentially faster than other methods, it should be noted that closed-loop system simulations may involve longer execution times than equivalent forward simulations. This is due mainly to the fact that the integration step size needed for a model with feedback is normally smaller than the integration step size for the equivalent forward model due to closed-loop system poles that lie far from poles and zeros of the forward model. The feedback system is thus stiffer than the forward model and integration algorithms for stiff systems may introduce extra computational burdens.

One application of feedback principles to inverse simulation involved model validation tools and was developed by Gray and von Grünhagen [36]. Their approach was based on explicit model-following methods of closed-loop system design and was applied to problems of helicopter flight control. Inverse simulation techniques were applied, together with a so-called 'open-loop' simulation procedure, to identify weak features within a non-linear MIMO helicopter model. The overall conclusion was that the combination of the open-loop approach and inverse simulation provides insight about physical sources of model deficiencies. The feedback method of inverse simulation, as compared with slower techniques based on optimisation, clearly helped to match the timescales for computation to the thought processes of the investigators engaged in the model validation process.

Although Gray and von Grünhagen [36] were concerned with model validation and used one feedback structure, their work provides useful pointers to the benefits of applying feedback principles for inverse simulation. Reports by Buchholz and von Grünhagen (see [34] and [35]) provide other useful information about the use of feedback methods for model inversion.

There are similarities between the processes of model inversion based on feedback principles and procedures used for the design of closed-loop control systems, but there are also important differences. Most of these differences relate to feedback system design requirements. Designing a feedback system for model inversion does not involve all the requirements that are necessary for control system design. One key difference is that in control system design, un-modelled or partially known external disturbances must be taken into account and model errors and uncertainties must also be allowed for, as well as measurement noise. No external disturbances or issues of measurement noise apply in a feedback system for model inversion and, since the requirement is to invert a given model, there can never be any modelling error or uncertainties. Thus, methods of design that are seldom used for control applications due to issues of poor disturbance rejection, susceptibility to measurement noise, or lack of robustness to model uncertainties, can be used without difficulty for model inversion (see e.g. [22]).

4.3 Example: inverse simulation applied to a linear model

The system model considered is a two-input two-output system having the form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4.35}$$

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} \tag{4.36}$$

where:

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

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

This system has two complex eigenvalues and one real eigenvalue in the left-half plane. There is only one transmission zero and this lies in the left half of the complex plane, showing that the system is minimum phase.

The approach used for this example is the method based on feedback system principles. Tests using MATLAB® and the MATLAB function *lsim* (see [37] and [38]) show that gains of 1,000 for both feedback pathways allow a satisfactory inverse model to be established. A signal which the model must follow, such as a sinusoid or repeated ramp signal, is first defined. The input signal needed to achieve this output is then determined from the inverse simulation and this calculated input is applied to the original forward simulation to establish whether it meets the requirements. Results from such tests are shown in Figures 4.4 to 4.7.

Figures 4.4 to 4.7 show that the feedback approach can successfully generate inverse simulations of linear MIMO models. The errors depend on the closed-loop dynamics and thus on the feedback system, and the number of possible inverse simulation models is therefore infinite as the number of designs is unlimited. Analysis functions within MATLAB® (see [37] and [38]) can, however, provide useful information

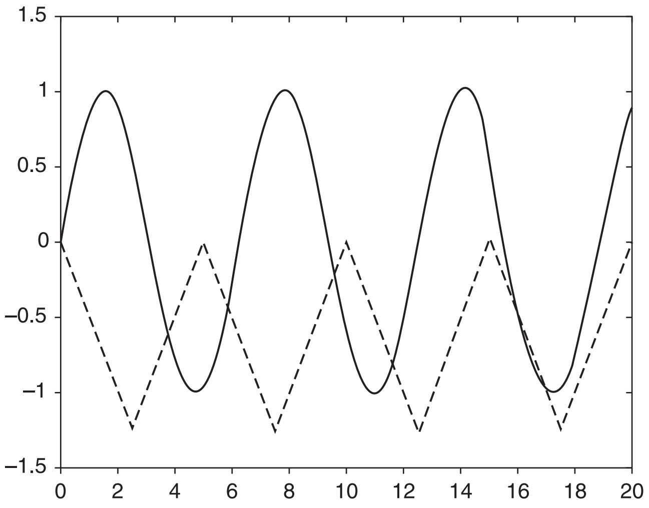


Figure 4.4 Required time histories of model outputs. The repeated ramp is required at output 1 and the sinewave is required at output 2. The x-axis scale represents time (s)

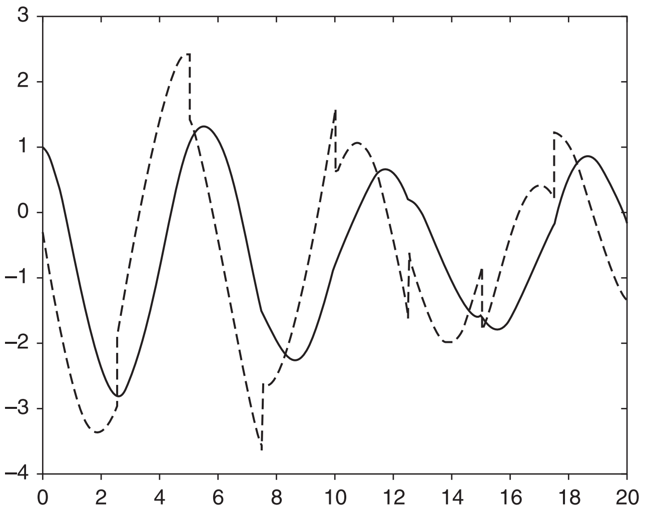


Figure 4.5 Input time histories found from inverse simulation model using gain factors of 1,000. The x-axis scale represents time (s). The discontinuous trace is for input 1 and the smoother trace is for input 2

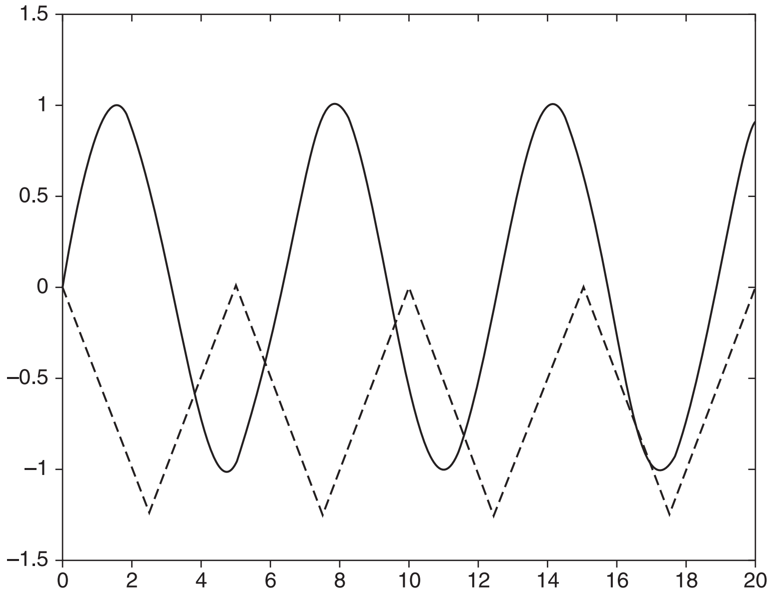


Figure 4.6 **Time histories obtained from forward simulation using the given model when subjected to inputs of Figure 4.5 found from inverse simulation. The x-axis represents time (s). The triangular wave is the trace for output 1 and the sinusoidal trace is at output 2 of the model**

about eigenvalues and transmission zeros for the closed-loop system, and this provides additional insight for use in assessing the inverse simulation.

The benefits of this approach are mainly in terms of computational speed. Overall, the computation time is similar to the time required for a conventional forward simulation run. However, it should be noted that the range of eigenvalues for the inverse simulation may be greater than that for the forward simulation model and integration methods for stiff systems may be necessary, even if these are not needed for the forward simulation. This may increase the time needed for the inverse simulation.

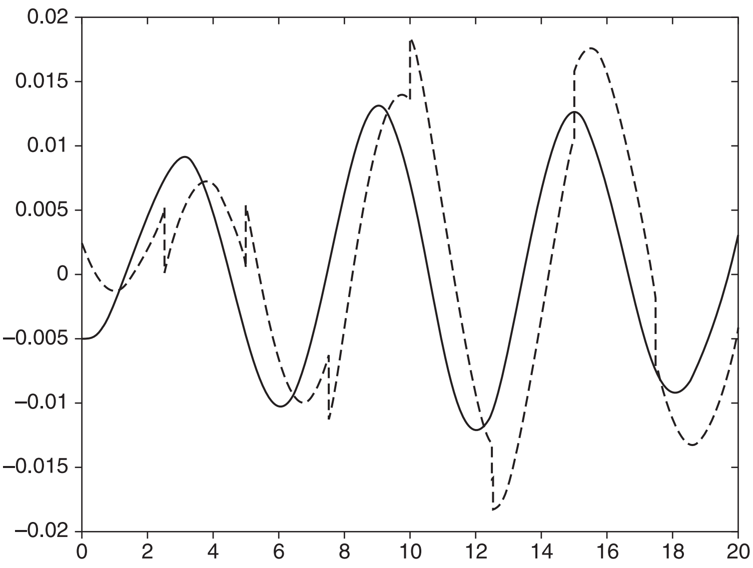


Figure 4.7 Errors between desired outputs and outputs from forward model subjected to inputs found from inverse simulation. The x-axis scale represents time (s). In this case the error between the desired output and the output from the forward simulation at output 1 is represented by the trace with discontinuities, while the smoother trace shows the error at output 2

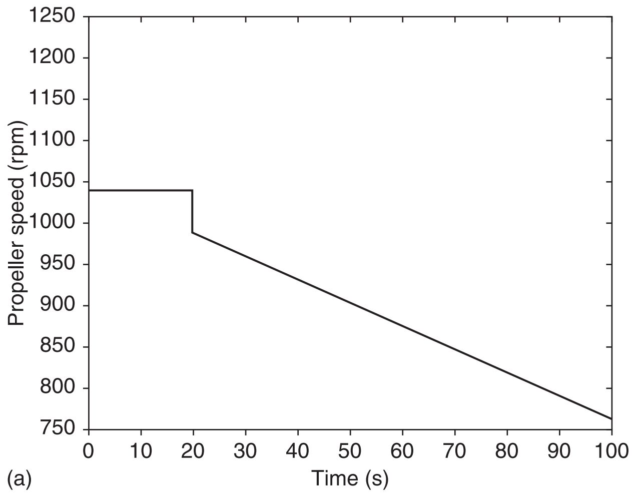
4.4 Case study: an application involving a nonlinear unmanned underwater vehicle (UUV) system model

Three illustrations of the use of the feedback approach to the nonlinear UUV model of Appendix A1 are considered here. The first case involves inverse simulation of the model to determine the propeller input time history required to achieve a given profile of surge velocity versus time. This involves proportional control with feedback of surge velocity in the model and comparison with the desired surge velocity. The second case does the same for the yaw rate state variable in terms of the required rudder deflection while the third case involves two feedback loops which both incorporate proportional control, one with feedback of surge velocity and the other yaw rate.

Case involving a specified pattern of surge velocity

Figure 4.8 shows results obtained by inverse simulation for a demanded input involving an initial demanded surge velocity held at a constant value of 3.88 m/s followed by a negative-going ramp change in surge velocity starting at time $t = 20$ s. The initial condition in the model involves a propeller speed of 1039 rpm and an initial surge velocity of 3.88 m/s. The propeller speed is limited to a maximum of 1500 rpm.

The results in Figure 4.8 were obtained using proportional control with a gain factor of 100 000, which was determined using elementary principles of feedback system design, together with some further adjustment based on trial and error methods. The only input applied involved the propeller speed and all other inputs for the forward simulation were zero. The results were obtained using a fixed-step Runge-Kutta integration method with integration step time of 0.01 s. The use of larger integration steps can lead to problems of numerical instability. Plot (a) shows the inverse solution in terms of the propeller speed pattern required to perform the manoeuvre. Plot (b) shows that when that pattern of propeller speed was applied to the forward simulation of the UUV, the surge velocity was (as required)



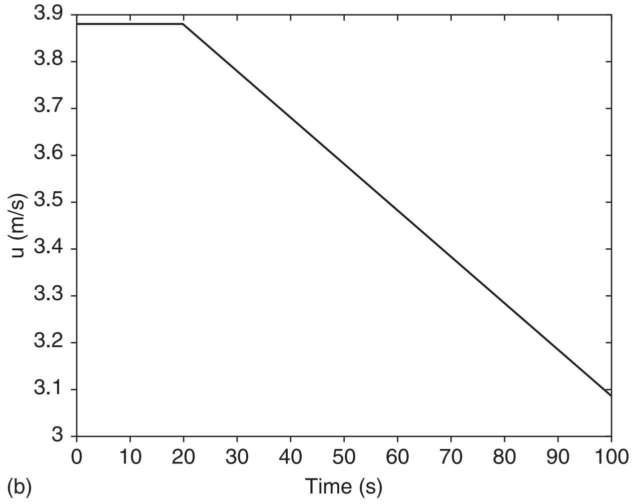
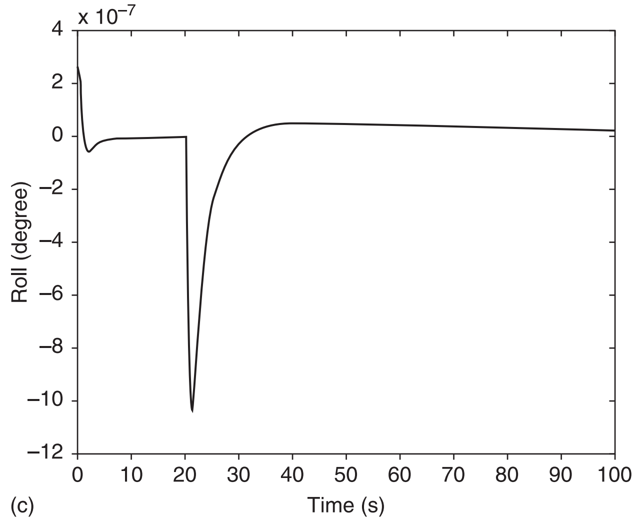


Figure 4.8 **Results for a single-loop feedback arrangement to generate the propeller input needed to produce a specified pattern of output involving a demanded surge velocity of 3.88 m/s from time *t* = 0 until time *t* = 20 s, followed by a negative-going ramp in terms of surge velocity change. The required propeller speed is shown in (a) and the plots (b), (c) and (d) show the surge velocity, roll angle and yaw angle of the UUV when subjected to that pattern of propeller speed**

(Continued)



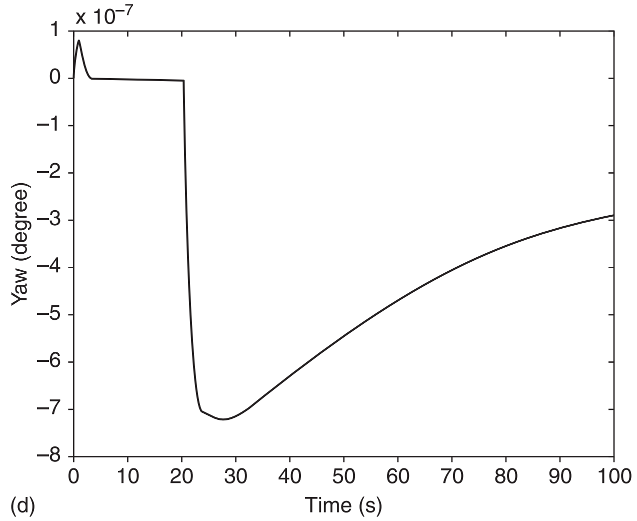


Figure 4.8 *(Continued)*

constant until the start of the negative going ramp at time t = 20 s. Thereafter the propeller speed started to fall, in a linear fashion, following a ramp that matched the demanded pattern. The results in plots (c) and (d) show respectively the roll and yaw angle time histories of the vehicle in response to the propeller speed input pattern generated by inverse simulation. It can be seen that the effect of the input on both these variables is very small and the same is true of all the other unconstrained state variables of the model, such as sway velocity and heave velocity. These results show that inverse simulation can be used successfully for this case involving a single input.

As can be seen from the results in Figure 4.8, the behaviour of the feedback loop used for the inverse simulation is stable and shows no unsatisfactory transient behaviour for this the chosen gain factor. Coupling to other variables is very small and this satisfactory situation is also reflected in Figure 4.9,

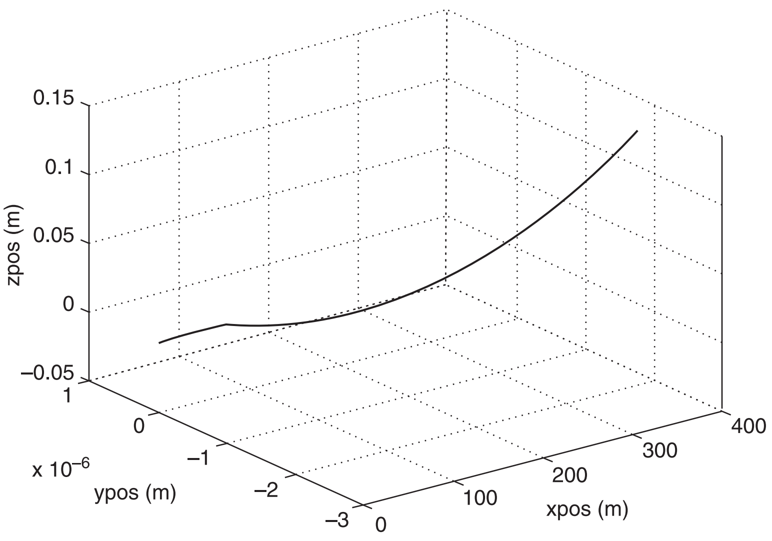


Figure 4.9 **Trajectory of the UUV in the earth-fixed axis system when subjected to the input shown in Figure 4.8 (a)**

which shows the trajectory of the vehicle in terms of the earth-fixed axis system.

More detailed examination of the response of the model to the input generated through inverse simulation shows that there is a steady-state error, as would be expected for proportional control, but this is very small because of the very large value of gain factor used.

It should be noted that if the required manoeuvre were modified so that it involves a demanded surge velocity of 5.0 m/s over the initial period from t = 0 to t = 15 s, with the initial condition as before (surge velocity of 3.88 m/s), the required propeller speed found from the inverse simulation reaches a limiting value of 1500 rpm over the initial part of the manoeuvre and the surge velocity takes some time to reach the required value, as would be expected. These results are shown in Figure 4.10 and confirm that the feedback approach can be used in situations where input variables reach their limiting values.

Case involving a specified pattern of yaw rate

In this case the rudder is the actuator used in the feedback system for inverse simulation, with feedback being taken from the yaw rate variable. This produces the inverse simulation result shown in plot (a) of Figure 4.11 in terms of the rudder deflection needed for a demanded yaw rate which is initially zero and at t = 50 s jumps to a value of –2.86 deg/s (–0.05 rad/s). With the chosen gain factor of 50, it may be found from plot (b) that the yaw response achieved has the correct form but involves a yaw-rate error of 0.06 deg/s, approximately. From the forward simulation it is found, as would be expected, that transients occur in variables such as sway velocity, roll angle (see plot (c)) and pitch rate when the rudder is deflected at time t = 50 s.

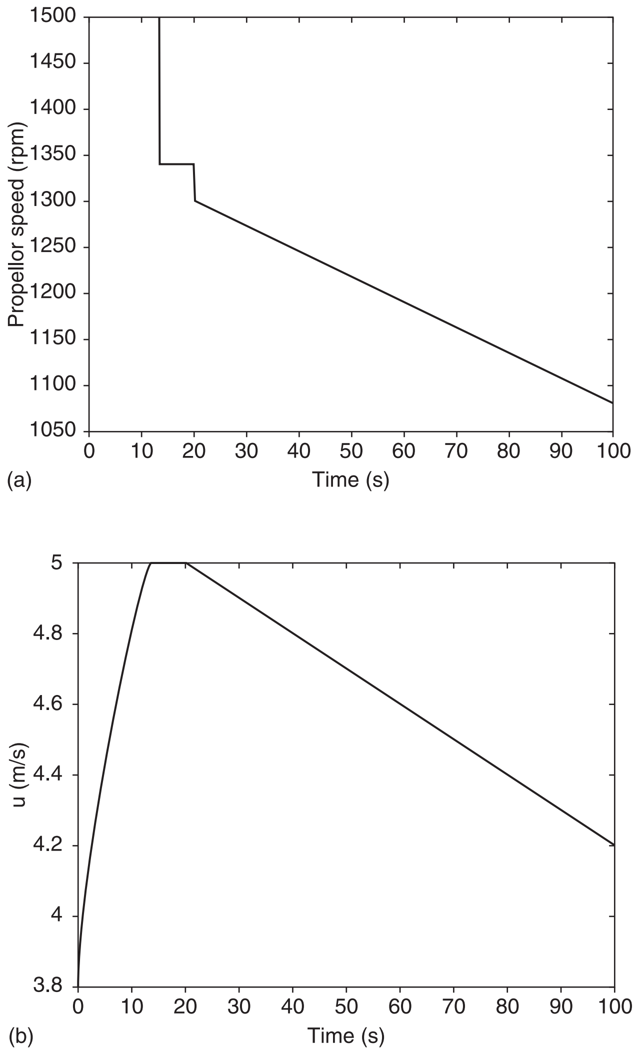


Figure 4.10 Results showing the propeller input for a demanded surge velocity of 5 m/s from time t = 0 until time t = 15 s, followed by a negative-going ramp in terms of surge velocity change. The required propeller speed (with the limiting value of 1500 rpm imposed) is shown in (a) and the plot (b) shows the surge velocity of the UUV when subjected to that pattern of propeller speed

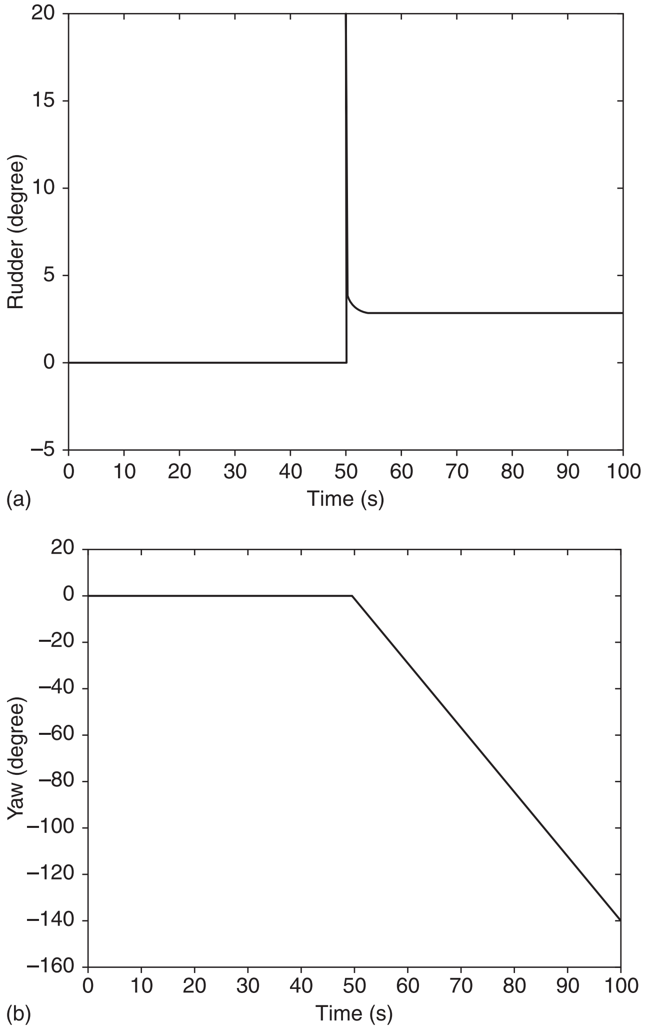


Figure 4.11 Plot (a) shows results obtained by inverse simulation in terms of the rudder deflection time history to give constant yaw rate of zero initially and – 0.05 rad/s from time *t* = 50 s. The gain factor in the yaw-rate feedback loop is 50 and the integration step size is 0.01 s. Plots (b) and (c) show the records of yaw angle and roll angle obtained when the rudder deflection input shown in plot (a) is applied to the forward simulation of the UUV

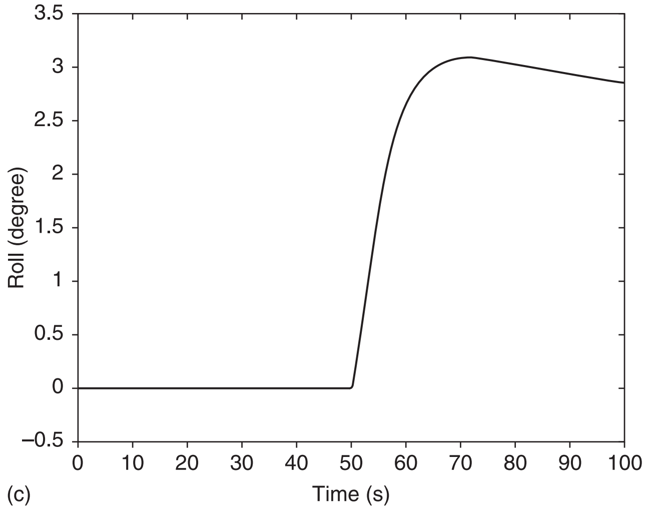


Figure 4.11 *(Continued)*

A fixed-step Runge-Kutta integration method with integration step time of 0.01 s was used to obtain these results. As was found for the case involving the propeller input, the use of larger integration steps can lead to problems of numerical instability. A hard limit of ± 20 degrees is applied to rudder deflections and it can be seen in plot (a) of Figure 4.11 that in the inverse simulation the rudder reaches the positive limit instantaneously at t = 50 s. Figure 4.12 shows the vehicle trajectory in the earth-axis system for this demanded manoeuvre.

An increased value of feedback path gain can be used to reduce the steady-state error in the inverse simulation results and values of gain factor as high as 1000 have been used successfully. However, it has been found that the use of gain factors much larger than 1000 can, in this case, give rise to limit cycle phenomena in the nonlinear closed-loop system used for inverse simulation and this limits the maximum usable gain factor to values significantly smaller than those for cases involving the propeller input.

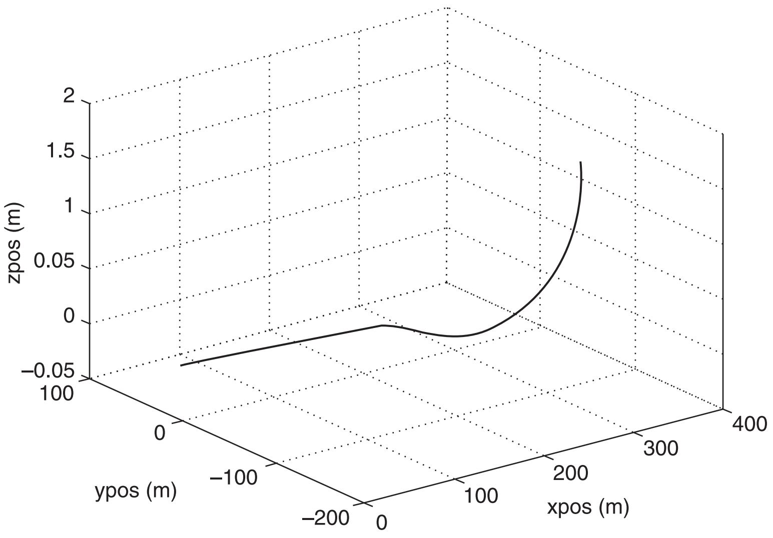


Figure 4.12 **Trajectory for the case presented in Figure 4.11, shown in the earth axis system**

Case involving a combination of demanded surge velocity and yaw rate

This case requires two feedback loops, one linking the required surge velocity to the propeller input and the other involving the required pattern of yaw rate and the rudder. For a gain of 50 in the rudder loop and 100 000 in the surge loop, with an integration step size of 0.01 s, the results from inverse simulation are as shown in Figure 4.13.

Results presented in Figure 4.13 show that the inverse simulation involving two feedback loops generates patterns of propeller and rudder inputs that lead to patterns of surge velocity and yaw rate that are close to the desired time histories. The rudder shows a very large transient at t = 50 s and it reaches the positive limit on rudder deflection of 20 degrees for a very short time. The yaw rate record resulting from this rudder input (as shown in plot (d)) is, however,

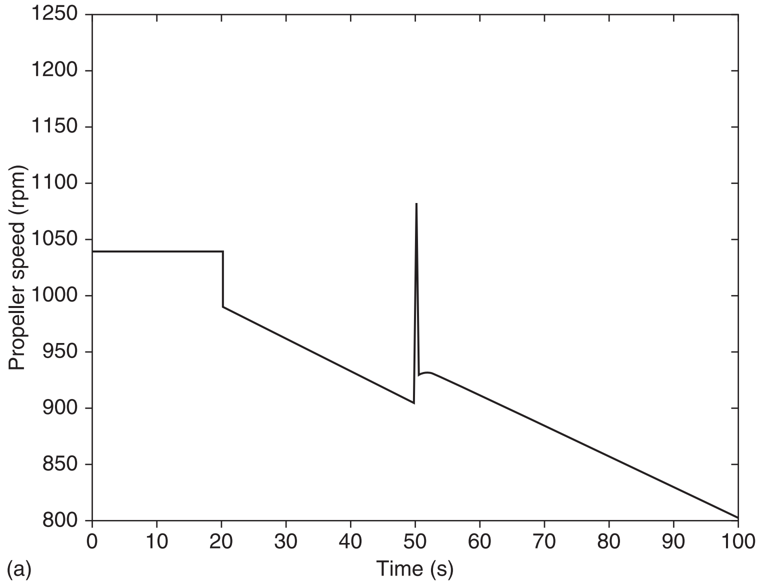
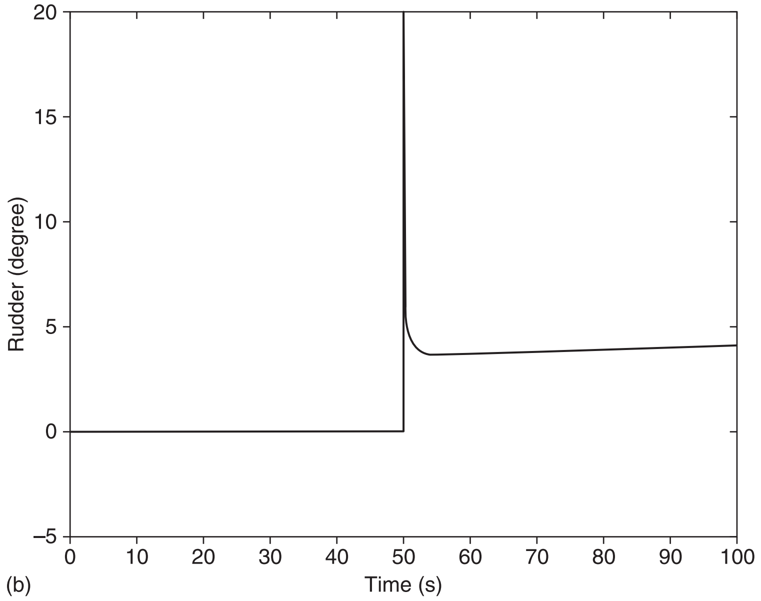


Figure 4.13 This case shows the combined effects of two inputs on the channels considered in Figures 4.8 and 4.11. Plot (a) shows the propeller input found through inverse simulation for inputs in the form of a demanded surge velocity of 3.88 m/s followed by negative ramp of surge velocity applied at time *t* = 20. This also shows the effects of the second demanded variable which involved a yaw rate of zero initially, changing suddenly to a constant value of −0.05 rad/s at time *t* = 50 s. Plot (b) shows the required rudder deflection for the same combination of demanded surge velocity and yaw rate. Plots (c) and (d) show, respectively, the surge velocity (m/s) and yaw angle (deg) resulting from the application of the propeller and rudder inputs shown in plots (a) and (b)

(Continued)



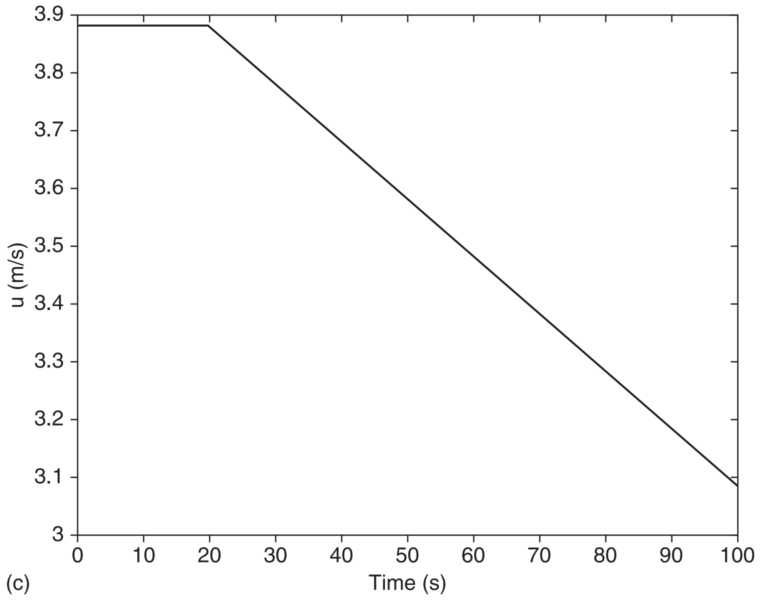


Figure 4.13 *(Continued)*

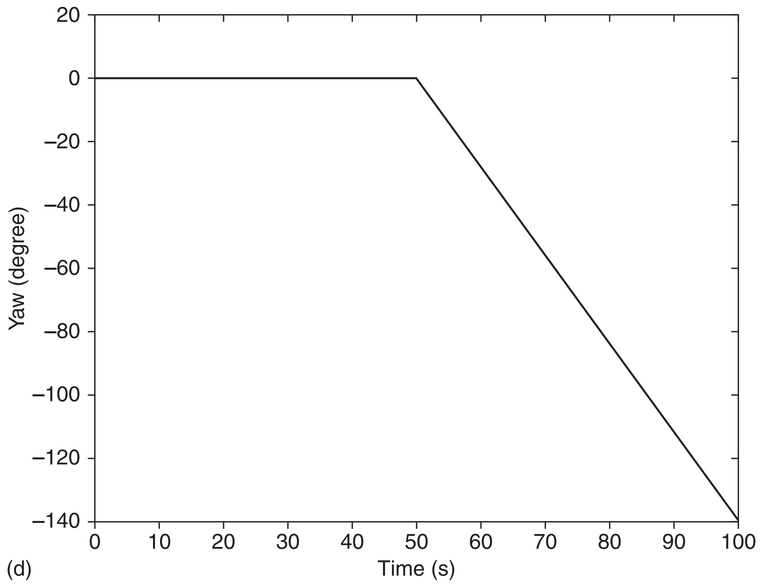


Figure 4.13 *(Continued)*

entirely reasonable. It should be noted that there is an interesting interaction between the two inputs. When the rudder changes position at time t = 50 s, there is a transient spike in the propeller speed record followed by an upward shift in the record over the remaining period. This change of propeller speed following the change of heading is clearly associated with increased resistance to motion of the vehicle and an increase in propeller speed is required to allow the demanded pattern of surge velocity to be maintained.

Although not shown in the results above, the responses of the other state variables of the forward model, when subjected to the combined propeller and rudder inputs of plots (a) and (b) in Figure 4.13, have the expected forms for these inputs. Figure 4.14 shows the corresponding earth-axis trajectory of the vehicle in response to the combined inputs.

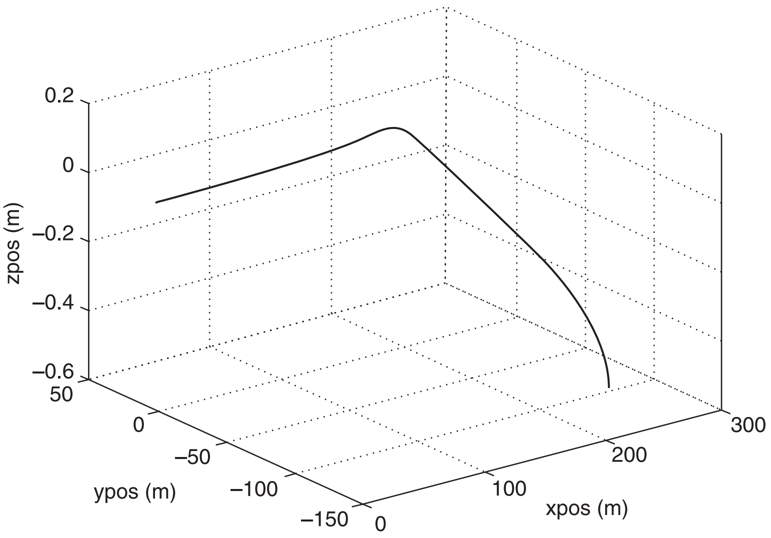


Figure 4.14 Trajectory for the case presented in Figure 4.13, shown in the earth axis system

The results presented above for the specific inverse simulation cases that have been considered are encouraging. Issues of feedback loop stability limits, interactions between inputs, saturation effects and possible limit cycle situations have been considered for the cases presented but have not given rise to any major problems in applying the feedback systems methodology. With proportional control in the feedback loops, finite steady-state errors are found (as would be expected), but these errors are generally predictable and could possibly be overcome by the introduction of more complex forms of feedback. This is a topic that could justify further investigation.

One of the most important features of the UUV model is that there are six inputs and twelve possible outputs, and decisions have to be made by the investigator about which actuators should be associated with each output variable in forming feedback loops. Physical understanding of the real system is important for this process and, although no difficulties have been encountered in this application, further research is needed in order to establish appropriate guidelines for the more general case.

The closed-loop system inevitably has a longer execution time for simulation than the forward model because the integration interval, generally, has to be shorter. Investigations based on linearised models should provide useful insight regarding the optimum choice of loop gain factors, integration step size and integration method.

4.5 Discussion

In summary, this chapter provides a review of inverse modelling and simulation methods. A number of iterative methods have been presented together with an approach based on feedback principles. Case studies have been presented illustrating the application of inverse simulation to linear and nonlinear SISO and MIMO systems, and showing that inverse simulation provides insight complementing that gained from conventional simulation.

The examples considered involve use of the feedback approach and this has been applied because it is believed to have considerable potential for modelling and simulation of integrated systems. It offers more physical understanding and improved solution speed when compared with some iterative approaches.

The feedback method does involve a trade-off between the complications of closed-loop design and possible computational benefits at run time. For the complex design problems that can arise in integrated system applications, in which optimisation may be an important part of the design process, those benefits in terms of speed of solution may be an important factor.

4.6 References

[1] Brockett, R. (1963) 'Poles, zeros and feedback: state space interpretation', *IEEE Transactions on Automatic Control*, Vol. AC-10, pp. 129–35.

[2] Dorato, P. (1969) 'On the inverse of linear dynamical systems', *IEEE Transactions on Systems Science and Cybernetics*, Vol. SSC-5, No. 1, pp. 43–8.

[3] Hirschorn, R.M. (1979) 'Invertibility of multivariable nonlinear control systems', *Journal of Guidance, Control and Dynamics*, Vol. 24, pp. 855–65.

[4] Isidori, I. (1989) *Nonlinear Control Systems: An Introduction*, Second Edition, Berlin, Germany, Springer.

[5] Lu, L. (2010) *Inverse Modelling and Inverse Simulation for Engineering Applications*, Lambert, Saarbrücken, Germany.

[6] Hunt, L.R. and Meyer, G. (1997) 'Stable inversion for nonlinear systems', *Automatica*, Vol. 33, No. 8, pp. 1549–54.

[7] Zou, Q. and Devasia, S. (2007) 'Preview-based inversion of nonlinear nonminimum-phase systems: VTOL example', *Automatica*, Vol. 43, No. 1, pp. 117–27.

[8] Slotine, J.-J. E. and Li, W. (1991) *Applied Nonlinear Control*, Englewood Cliffs, NJ, USA, Prentice Hall.

[9] Bradley, R. (1996) *The Flying Brick Exposed: Nonlinear Control of a Basic Helicopter*, Glasgow Caledonian University, TR/MAT/RB/96-51.

[10] Thomson, D.G. and Bradley, R. (1998) 'The principles and practical application of helicopter inverse simulation', *Simulation Practice and Theory*, Vol. 6, pp. 47–70.

[11] Murray-Smith, D.J. (2000) 'The inverse simulation approach: a focused review of methods and applications', *Mathematics and Computers in Simulation*, Vol. 53, pp. 239–47.

[12] Thomson, D. and Bradley, R. (2006) 'Inverse simulation as a tool for flight dynamics research – principles and applications', *Progress in Aerospace Sciences*, Vol. 42, pp. 174–210.

[13] Kato, O. and Sugiura, I. (1986) 'An interpretation of airplane motion and control as an inverse problem', *Journal of Guidance Control and Dynamics*, Vol. 9, No. 2, pp. 198–204.

[14] Thomson, D.G. and Bradley, R. (1990) 'Development and verification of an algorithm for helicopter inverse simulation', *Vertica*, Vol. 14, No. 2, pp. 185–200.

[15] Thomson, D.G. and Bradley, R. (1994) 'The contribution of inverse simulation to the assessment of helicopter handling qualities', in *Proceedings of the 19th ICAS Conference, Anaheim, USA, September 1994*, Paper 7.3.2.

[16] Hess, R.A., Gao, C. and Wang, S.H. (1991) 'A generalized technique for inverse simulation applied to aircraft maneuvers', *Journal of Guidance Control and Dynamics*, Vol. 14, pp. 920–6.

[17] Rutherford, S. and Thomson, D.G. (1997) 'Helicopter inverse simulation incorporating an individual blade rotor model', *AIAA Journal of Aircraft*, Vol. 34, No. 5.

[18] Lu, L., Murray-Smith, D.J. and Thomson, D.G. (2008) 'Issues of numerical accuracy and stability in inverse

simulation', *Simulation Modelling Practice and Theory*, Vol. 16, pp. 1350–64.

[19] de Matteis, G., de Socio, L.M. and Leonessa, A. (1995) 'Solution of aircraft inverse problems by local optimization', *Journal of Guidance Control and Dynamics*, Vol. 18, No. 3, pp. 567–71.

[20] Lee, S. and Kim, Y. (1997) 'Time-domain finite element method for inverse problem of aircraft manoeuvres', *Journal of Guidance, Control and Dynamics*, Vol. 20, No. 1, pp. 97–103.

[21] Celi, R. (2000) 'Optimization-based inverse simulation of a helicopter slalom manoeuvre', *Journal of Guidance, Control and Dynamics*, Vol. 23, No. 2, pp. 289–97.

[22] Murray-Smith, D.J. (2011) 'Feedback methods for inverse simulation of dynamic models for engineering systems', *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 17, No. 5, pp. 515–41.

[23] Thomson, D.G. and Bradley, R. (1986) 'An analytical method for quantifying helicopter agility. Paper 45', in *Proceedings of the 12th European Rotorcraft Forum, Garmisch-Partenkirchen, Federal Republic of Germany*, September.

[24] Thomson, D.G. and Bradley, R. (1987) 'Recent developments in the calculation of inverse solutions of the helicopter equations of motion', in *Proceedings of UK Simulation Council Conference, University College of North Wales, 9–11 September 1987*, pp. 227–34, UKSC, Ghent, Belgium.

[25] Liceago-Castro, E. (1988) *A Geometric Control System with Application to Helicopters*, PhD Thesis, University of Glasgow.

[26] Lu, L., Murray-Smith, D.J. and Thomson, D. (2007) 'Sensitivity analysis method for inverse simulation',

Journal of Guidance, Control and Dynamics, Vol. 30, No. 1, pp. 114–21.

[27] Avanzini, G. and de Matteis, G. (1999) Two-timescale integration method for inverse simulation, *Journal of Guidance, Control and Dynamics*, Vol. 22, No. 3, pp. 395–401.

[28] Avanzini, G. and de Matteis, G. (2001) 'Two-timescale inverse simulation of a helicopter model', *Journal of Guidance, Control and Dynamics*, Vol. 24, No. 2, pp. 330–9.

[29] Nelder, J.A. and Mead, R. (1965) 'A simplex method for function minimization', *Computer Journal*, Vol. 7, pp. 308–13.

[30] Fritzson, P. (2004) *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*, IEEE Press, Piscataway NJ, USA.

[31] Thümmel, M., Looye, G., Kurze, M., Otter, M. and Bals, J. (2005) 'Nonlinear inverse models for control', in Schmitz, G. (ed.), *Proceedings of the 4th International Modelica Conference, Hamburg, March 7–8, 2005*, pp. 267–79, Modelica Association, Linköping, Sweden, 2005 (online): *www.Modelica.org/events/Conference2005* (accessed 1 January 2012).

[32] Williams, R.W. (1961) *Analogue Computation*, p. 215, Heywood, London, UK.

[33] Charlesworth, A.S. and Fletcher, J.R. (1967) *Systematic Analogue Computer Programming*, pp. 117–18, London, UK, Pitman.

[34] Buchholz, J.J and von Grünhagen, W. (2004) *Inversion Impossible?*, Technical Report, University of Applied Sciences, Bremen, Germany, September.

[35] Buchholz, J.J and von Grünhagen, W. (2005) *Inversion dynamischer Systeme mit Matlab*, Technical Report,

University of Applied Sciences, Bremen, Germany, August.

[36] Gray, G.J. and von Grünhagen, W. (1998) 'An investigation of open-loop and inverse simulation as nonlinear model validation tools for helicopter flight mechanics', *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 4, No. 1, pp. 32–57.

[37] Anonymous (1996) *MATLAB® Control Systems Toolbox User's Guide*, The Mathworks Inc, Natick, MA, USA.

[38] MATLAB®/Simulink® modelling and simulation software, The Mathworks Inc (online): *www.mathworks.com/products* (accessed 30 June 2011).

5

Methods and applications of parameter sensitivity analysis

Abstract: Parameter sensitivity analysis provides an efficient way of assessing parametric dependencies in mathematical models and computer simulations. This is important for design optimisation, for estimating the effects of modelling errors and uncertainties in the analysis of system performance, for understanding issues such as test input design in experimental modelling and in the external validation of models. This chapter provides a review of methods of parameter sensitivity analysis and considers applications involving linear and nonlinear lumped parameter models.

Key words: sensitivity function, output sensitivity, sensitivity model, sensitivity bond graph.

5.1 Fundamental concepts of parameter sensitivity analysis

Parameter sensitivity analysis techniques are important for establishing how responses of a model change when parameters are varied and which of its parameters most influence the model behaviour (see e.g. [1] and [2]). Models are never exact and it is important to be able to assess parametric dependencies at the model development stage as part of an investigation of modelling assumptions, simplifications and overall credibility. This can lead to an understanding of the effects of component tolerances and how the system performance may degrade as components change with environmental conditions or through the processes of ageing. Sensitivity information is also very important for system optimisation in design and it should be noted that methods of optimisation based on gradient methods make direct use of parameter sensitivity measures.

For the applications being considered in this book, the model may be in lumped parameter or distributed parameter form, continuous or discrete, linear or nonlinear. The sensitivity may also be characterised in a number of ways. Common measures are based on the time domain, but frequency-domain measures can also be very important in some fields, as are measures involving a performance index (see e.g. [2]).

Information about parameter sensitivities, when taken together with structural information, can also be of considerable value in experimental modelling and, more generally, in the iterative development and refinement of complex system models. These issues are addressed in Chapters 6 and 7. In situations where changes in system parameters interact with structural issues to cause discontinuous changes in the overall behaviour, the sensitivity analysis may be termed ‘singular perturbation’ analysis (see e.g. [3] and [4]).

Early research by Tomović and others at the Pupin Institute in Belgrade (see e.g. [1], [4] and [5]) provided a foundation for much published work on sensitivity analysis of dynamic models. Important contributions have also been made by Frank [2] and by Rosenwasser and Yusupov [6], who used a more mathematical approach which emphasises the fact that sensitivity theory can be interpreted as part of a more general theory of systems in which parameter variations are considered as system inputs.

Much of the early literature on sensitivity issues put a particular emphasis on properties of feedback systems and automatic control, and Bode [7] was responsible for much early work on parameter sensitivity in the frequency domain. Many of the issues raised originally by Bode in the general context of systems with feedback were considered further by Horowitz [8] in relation to automatic control system design and applications.

5.2 The sensitivity function

All methods of sensitivity analysis involve the concept of the *sensitivity coefficient* or *sensitivity function*. In the time domain, which has special significance in the context of state-space models, the sensitivity function is defined in terms of a Taylor series expansion for a system variable in terms of the parameter that is varying. For example, if $y(t,q_0)$ is a model output response to a given input and this is a function of time (t) as well as a parameter q, the difference between a response $y(t,q_0)$ where q has the particular value q_0, and a response $y(t,q_0 + \Delta q)$ where the parameter q takes a new value $q_0 + \Delta q$ is given by:

$$y(t,q_0+\Delta q) = y(t,q_0) + \frac{\partial y}{\partial q}\Delta q + \frac{1}{2!}\frac{\partial^2 y}{\partial q^2}(\Delta q)^2 + \ldots \tag{5.1}$$

If Δq is sufficiently small:

$$y(t,q_0+\Delta q) \approx y(t,q_0) + \frac{\partial y}{\partial q}\Delta q \tag{5.2}$$

The first-order *output sensitivity function* is the quantity $\frac{\partial y}{\partial q}$. The effective linearisation due to truncation of the series after the second term allows the superposition principle to be used to find the effect of simultaneous changes of the values of several different parameters. This can be extended to the trajectory sensitivity function in the case of a state-space description $\frac{\partial \mathrm{x}}{\partial q}$ where x is the vector of state variables. As in the case of output sensitivity, the trajectory sensitivity is evaluated for perturbations in the parameter value q about a specific value q_0.

For quantitative comparisons of the influence of particular parameters, the *relative sensitivity function* may be a useful measure. Where the comparisons all involve the sensitivity of one specific model variable x to a number of different parameters q, an appropriate measure of relative sensitivity is $q_i \frac{\partial x}{\partial q_i}$. For cases involving several variables, an entirely dimensionless form of relative sensitivity measure $\frac{q_i}{x_{mj}} \frac{\partial x_j}{\partial q_i}$ may be more useful, where the quantity x_{mj} is some appropriate measure of the variable $x_j(t,q_0)$, such as the maximum value or the mean value.

5.3 Methods of sensitivity analysis involving repeated solutions

From Equation (5.2) it is possible to write:

$$\frac{\partial y}{\partial q_0} \approx \frac{y(t,q_0+\Delta q) - y(t,q_0)}{\Delta q} \tag{5.3}$$

And thus the sensitivity function can be found using a finite difference approximation. This requires two repeated simulation runs and selection of an appropriate size for the perturbation Δq. The accuracy of the estimation thus depends on that choice and repeated tests may be needed with a number of trial values of Δq to find a suitable value. Separate sensitivity function evaluations are needed for all the parameters of interest.

The main objection to the use of this approach, in addition to the fact that it requires repeated simulation runs, is one of accuracy. Taking the difference between the two solutions with inherent numerical errors ε_1 and ε_2, respectively, leads to the following:

$$\frac{\partial y}{\partial q_0} \approx \frac{y(t,q_0+\Delta q)-y(t,q_0)}{\Delta q}+\frac{\varepsilon_1-\varepsilon_2}{\Delta q} \tag{5.4}$$

Since Δq must be small in order to give a small parameter perturbation, the resulting error due to the term $(\varepsilon_1 - \varepsilon_2)/\Delta q$ may be significant [1].

The relative sensitivity measures corresponding to Equation (5.3) are:

$$\frac{\partial y}{\partial q_0} \approx q\frac{y(t,q_0+\Delta q)-y(t,q_0)}{\Delta q} \tag{5.5}$$

$$\frac{\partial y}{\partial q_0} \approx \frac{q}{y_m}\frac{y(t,q_0+\Delta q)-y(t,q_0)}{\Delta q} \tag{5.6}$$

where y_m is a measure of the variable $y(t,q_0)$, such as the maximum value or the mean.

5.4 Methods of sensitivity analysis involving sensitivity models

If a general form of nonlinear dynamic model is described by a set of equations:

$$f_i(\dot{x}_i;x_1, x_2, \ldots, x_n; u_1, u_2, \ldots .u_r;t;q_0) = 0 \tag{5.7}$$

$$g_i(y_i;x_1, x_2, \ldots, x_n; u_1, u_2, \ldots .u_r;t;q_0) = 0 \tag{5.8}$$

where x_1, x_2, . . ., x_n are state variables, u_1, u_2, . . . u_r are inputs and y_1, y_2, . . . y_m are output variables, it is clear from Section 5.1 that the sensitivity of this system to variation of the parameter q may be obtained by partial differentiation with respect to q. This process gives a set of *sensitivity equations* which can also be termed the *sensitivity model* or *co-system*. These equations have the form:

$$\frac{\partial f_i}{\partial \dot{x}_i}\frac{\partial \dot{x}_i}{\partial q_0} + \frac{\partial f_i}{\partial x_1}\frac{\partial x_1}{\partial q_0} + \ldots\ldots\ldots\ldots + \frac{\partial f_i}{\partial x_n}\frac{\partial x_n}{\partial q_0} + \frac{\partial f_i}{\partial q_0} = 0 \tag{5.9}$$

$$\frac{\partial f_i}{\partial y_i}\frac{\partial y_i}{\partial q_0} + \frac{\partial g_i}{\partial x_1}\frac{\partial x_1}{\partial q_0} + \ldots\ldots\ldots\ldots + \frac{\partial g_i}{\partial x_n}\frac{\partial x_n}{\partial q_0} + \frac{\partial g_i}{\partial q_0} = 0 \tag{5.10}$$

These sensitivity equations, when solved in conjunction with the system model equations, provide solutions for the state variable and output sensitivity functions. They are, in general, linear ordinary differential equations with time-varying coefficients and may be solved by analytical or simulation methods. Figure 5.1 is a schematic diagram for the generation of sensitivity functions using this approach. For a linear model with constant coefficients, the structure of the sensitivity model is generally very similar to that of the system model but, for each output, a given model must have as many sensitivity models as the number of parameters of interest, although methods have been developed that allow simultaneous estimation of many sensitivity functions using a single sensitivity model for particular cases.

Initial conditions for sensitivity models are, in most cases, zero. This issue is considered in detail by Frank [2].

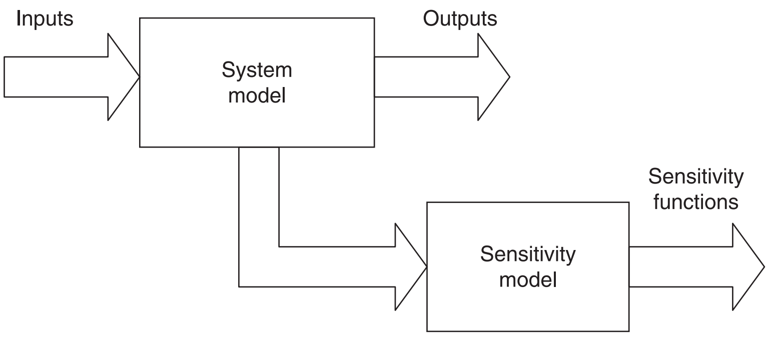


Figure 5.1 **Block diagram illustrating the relationship between the system model and the corresponding sensitivity model for the general case of a multi-input multi-output system model**

Non-zero initial conditions can arise, for example, in the sensitivity models for cases in which the system model has a variable structure.

One important quantity based on sensitivity functions is the *sensitivity matrix*. This is defined as the matrix of partial derivatives of model variables, such as state variables, with respect to parameters of interest in a model. For the case of n variables $y_1, y_2, \ldots y_n$ and p parameters of interest $q_1, q_2, \ldots q_p$ the sensitivity matrix has the form:

$$X = \begin{bmatrix} \frac{\partial y_1}{\partial q_1} & \cdots & \frac{\partial y_n}{\partial q_p} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial q_1} & \cdots & \frac{\partial y_n}{\partial q_p} \end{bmatrix} \tag{5.11}$$

Sensitivity matrices and related quantities are discussed in greater detail in Chapter 6 in the context of system identification and parameter estimation techniques.

5.4.1 Parameter sensitivity analysis of models in state-space form

Consider a model of order *n* having *m* inputs and *p* outputs, and described by a set of linearised state-space equations of the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} \tag{5.12}$$

$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du} \tag{5.13}$$

Then, taking Laplace transforms and assuming initial conditions for the state variables $\boldsymbol{x}$ to be zero, one can write:

$$s\boldsymbol{X}(s) = \boldsymbol{AX}(s) + \boldsymbol{BU}(s) \tag{5.14}$$

$$\boldsymbol{Y}(s) = \boldsymbol{CX}(s) + \boldsymbol{DU}(s) \tag{5.15}$$

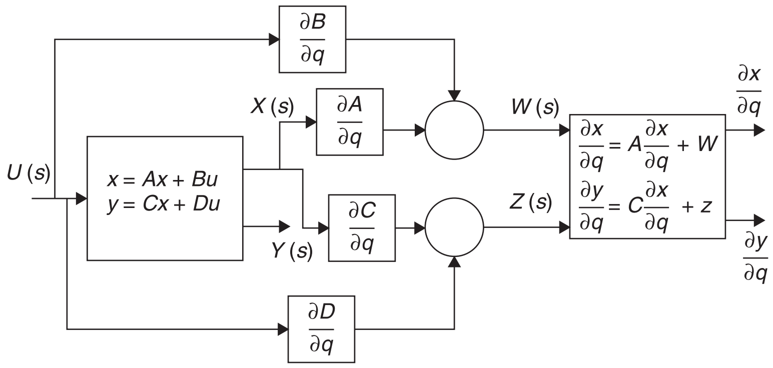
Then, differentiating with respect to a parameter *q*, which can affect any or all of the coefficients in the matrices ***A***, ***B***, C and ***D***, we get the following set of equations:

$$s\frac{\partial \boldsymbol{X}}{\partial q} = \boldsymbol{A}\frac{\partial \boldsymbol{X}}{\partial q} + \frac{\partial \boldsymbol{A}}{\partial q}\boldsymbol{X} + \boldsymbol{B}\frac{\partial \boldsymbol{U}}{\partial q} + \frac{\partial \boldsymbol{B}}{\partial q}\boldsymbol{U} \tag{5.16}$$

$$\frac{\partial \boldsymbol{Y}}{\partial q} = \boldsymbol{C}\frac{\partial \boldsymbol{X}}{\partial q} + \frac{\partial \boldsymbol{C}}{\partial q}\boldsymbol{X} + \boldsymbol{D}\frac{\partial \boldsymbol{U}}{\partial q} + \frac{\partial \boldsymbol{D}}{\partial q}\boldsymbol{U} \tag{5.17}$$

Since *q* is a parameter of the system model and not of any of the inputs $\boldsymbol{U}(s)$, it follows that $\frac{\partial U}{\partial q}$ is zero. Hence the solution of the sensitivity equations may be generated from a simulation block diagram of the form shown in Figure 5.2.

The structural similarities between the system model and the sensitivity model provide practical benefits in the case of linear models but, for a nonlinear model, the structural relationship between the model and the sensitivity equations is more complicated, although some similarities remain. The complexity in such cases depends on the parameter of interest



$\frac{\partial B}{\partial q}$
$X(s)$
$\frac{\partial A}{\partial q}$
$W(s)$
$\frac{\partial x}{\partial q}$
$U(s)$
$x = Ax + Bu$
$y = Cx + Du$
$\frac{\partial x}{\partial q} = A\frac{\partial x}{\partial q} + W$
$\frac{\partial y}{\partial q} = C\frac{\partial x}{\partial q} + z$
$Z(s)$
$\frac{\partial C}{\partial q}$
$Y(s)$
$\frac{\partial y}{\partial q}$
$\frac{\partial D}{\partial q}$

Figure 5.2 **Block diagram showing system model in state-space form together with the corresponding sensitivity model. Here, the lines connecting blocks can represent either a single variable or a vector, depending on the application**

and whether or not it is a parameter associated with a nonlinearity of the model. Issues arising with nonlinear models are considered in more detail by Frank [2].

An example, involving the UUV system of Appendix A1, is included in Section 5.5 where it is used to illustrate the application of parameter sensitivity analysis to nonlinear and linearised models.

5.4.2 Parameter sensitivity analysis of linear models in transfer function form

Sensitivity analysis for system models described by lumped parameter linear equations with constant coefficients and arranged in transfer function form can be handled using an approach developed first by Kokotović [5] which is known as the 'sensitivity points' method. This work was extended to state-space models by Wilkie and Perkins [9].

If a SISO linear system with input $u(t)$ and output $y(t)$ is described by a transfer function $G(s)$ then, in terms of Laplace transformed variables, the sensitivity of the output, $Y(s)$, to small changes of any parameter qi of the model may be described by the equation:

$$\frac{\partial Y(s)}{\partial q_i} = \frac{\partial G(s)}{\partial q_i} U(s) = \frac{1}{G(s)} \frac{\partial G(s)}{\partial qi} Y(s) \tag{5.18}$$

The sensitivity function $\frac{\partial y(t))}{\partial q_i}$ can therefore be found if the system output $y(t)$ is applied as input to a sensitivity model which has transfer function $\frac{1}{G(s)} \frac{\partial G(s)}{\partial q_i} Y(s)$. It has been shown by Kokotović [5] that, for multi-loop linear models, the appropriate sensitivity model involving a ratio of two polynomials in s may be represented by a multi-loop feedback structure. This may be demonstrated readily using an example for a system described initially by a transfer function of order n of the form:

$$G(s) = \frac{K\Sigma_0^m a_i s^{m-i}}{\Sigma_0^n b_j s^{n-j}} = \frac{K\Sigma_0^m a_i s^{m-n-i}}{\Sigma_0^n b_j s^{-j}} \tag{5.19}$$

where $a_0 = b_0 = 1$.

It may then be shown that:

$$\frac{\partial Y(s)}{\partial b_r} = \frac{-s^{-r} Y(s)}{\Sigma_0^n b_j s^{-j}} \tag{5.20}$$

Figure 5.3 shows a block diagram model for the case of $m = n = 3$ and the corresponding sensitivity model block diagram for the denominator coefficients is shown in Figure 5.4. The points in the sensitivity model at which sensitivity functions are obtained are the outputs of each block involving

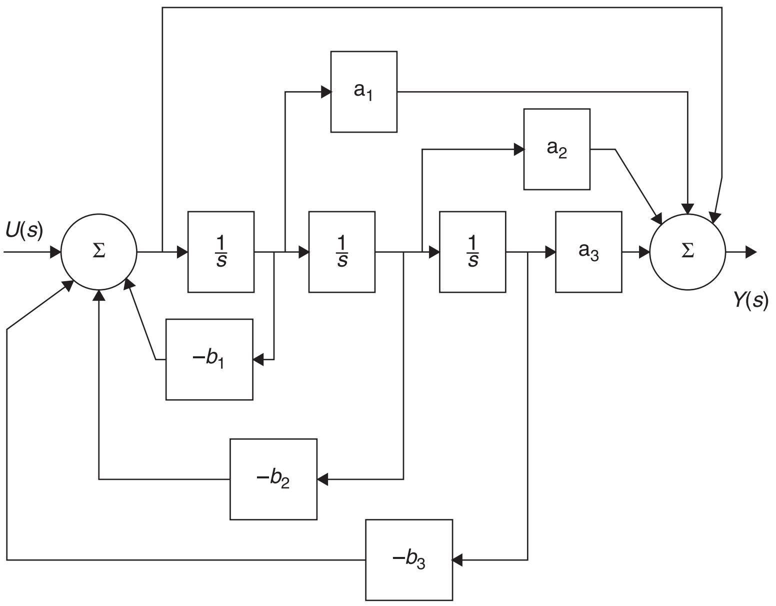


Figure 5.3 Block diagram for linear system transfer function *G*(*s*) for the forms of numerator and denominator given in Equation (5.19). In this diagram the Σ sign indicates the sum (positive) of all the incoming signals at that summation point

an integrator element. The sensitivities for all the coefficients of the denominator of $G(s)$ may thus be found simultaneously.

For numerator coefficients:

$$\frac{\partial Y(s)}{\partial a_r} = \frac{Ks^{m-n-r}U(s)}{\Sigma_0^n b_j s^{-j}} \tag{5.21}$$

Again, for the third-order example used above, Figure 5.5 shows the sensitivity model for the numerator coefficients and it may be seen that the integrator block outputs (i.e. the state variables) within the system model $G(s)$ give the sensitivity functions for the numerator coefficients directly, as shown in Figure 5.5.

Using a block-diagram oriented simulation tool such as Simulink®, it is thus very easy to set up a simulation that generates, simultaneously, all of the sensitivity functions of the system output with respect to all the coefficients of the numerator and denominator. The sensitivity co-system is linked directly to the output of the corresponding simulation models and, for the third-order example presented here, the sensitivity model requires three integrator blocks.

The representation of the system in the block diagram form of Figure 5.3 corresponds to a system state matrix in *companion-canonic form* where the output of each integrator block is taken as a state variable. It has been shown [9] that the sensitivity of each state variable to each parameter is

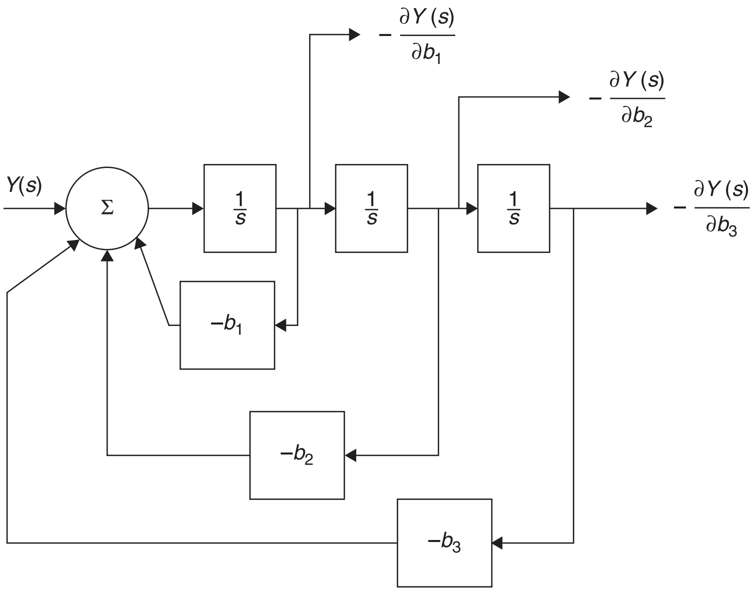
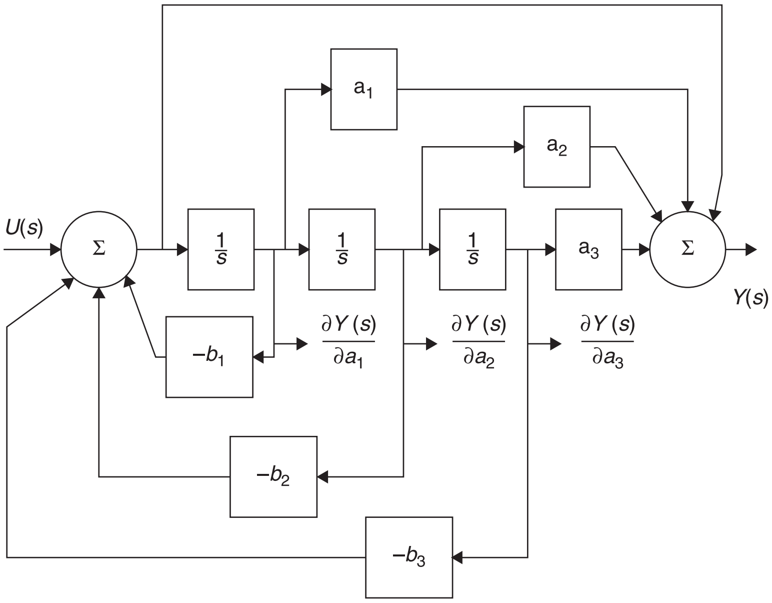

$-\frac{\partial Y(s)}{\partial b_1}$
$-\frac{\partial Y(s)}{\partial b_2}$
$Y(s)$
Σ
$\frac{1}{s}$
$\frac{1}{s}$
$\frac{1}{s}$
$-\frac{\partial Y(s)}{\partial b_3}$
$-b_1$
$-b_2$
$-b_3$

Figure 5.4 **Block diagram for sensitivity model for denominator coefficients of the transfer function represented in Figure 5.3**



a₁
a₂
a₃
$U(s)$
Σ
$\frac{1}{s}$
Σ
$Y(s)$
$\frac{\partial Y(s)}{\partial a_1}$
$\frac{\partial Y(s)}{\partial a_2}$
$\frac{\partial Y(s)}{\partial a_3}$
$-b_1$
$-b_2$
$-b_3$

Figure 5.5 **Generation of sensitivity functions for the numerator coefficients of the transfer function represented in Figure 5.3**

directly available from a single sensitivity model for a system matrix in companion-canonic form. For multi-input linear systems it can be shown [9] that for *p* inputs at most 2*p*-1 *n*th order sensitivity models are needed in order to generate all of the sensitivity coefficients of the state variables with respect to any parameter.

5.5 Case study: sensitivity analysis applied to the unmanned underwater vehicle (UUV) model

The model of the UUV system of Appendix A1 could, in principle, be used to illustrate the application of this methodology for nonlinear models. However, for the nonlinear case, each parameter has to be considered separately and this leads to a very large number of sensitivity models in order to analyse the system model completely. Given the relatively complex form of the UUV model, this would involve a set of sensitivity models which would be time-consuming in terms of the setting-up process and also in terms of the simulation runs needed if the analysis involved more than one or two parameters of the model. However, if the system equations can be linearised, the process becomes much more straightforward since all the sensitivity functions can be found from a single sensitivity model and this can also, very often, provide useful physical insight that cannot be obtained so readily from sensitivity analysis of the full nonlinear model. Such insight can be particularly useful when the linearised model provides a basis for design or optimisation, as often happens with control system design and integration.

The full nonlinear model may be vital for simulation-based assessment of a design at the final stage before implementation, but parameter sensitivity analysis may be of secondary importance by that stage. In many cases, therefore, parameter sensitivity analysis of the linearised model can be as useful as simulation results from a nonlinear model. The use of linearised models is illustrated through the example that follows.

5.5.1 Sensitivity analysis of the linearised UUV model for diving motion

From Section A1.1.1 the linearised dynamics describing diving motion of the underwater vehicle model of Appendix

A1 involves a third-order model described by the following set of first-order equations:

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \dfrac{M_q}{I_y - M_{\dot{q}}} & -\dfrac{(z_G - z_B)W}{I_y - M_{\dot{q}}} & 0 \\ 1 & 0 & 0 \\ 0 & -u_0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} \dfrac{M_\delta}{I_y - M_{\dot{q}}} \\ 0 \\ 0 \end{bmatrix} \delta_s \tag{5.22}$$

The state variables q, θ and z are the pitch rate, pitch angle and vertical displacement, as given in Appendix A1, and δ_s is the stern plane deflection. The parameters M_q, $M_{\dot{q}}$ and M_δ are hydrodynamic coefficients, I_y is the moment of inertia of the vehicle for pitching motion, W is the weight of the vehicle, $z_G - z_B$ is the vertical distance between the centre of gravity and the centre of buoyancy, and u_0 is the forward speed of the vehicle.

In simplified form these equations may be written as:

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 1 & 0 & 0 \\ 0 & -u_0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix} \delta_s(t) \tag{5.23}$$

It may be seen from Equations (5.22) and (5.23) that the parameter a_{11} depends directly on M_q and inversely on I_y and $M_{\dot{q}}$. Similarly, the parameter a_{12} depends directly upon W, z_G and z_B, and inversely on I_y and $M_{\dot{q}}$, while b_1 depends directly on M_δ and inversely on I_y and $M_{\dot{q}}$. The only other significant parameter of Equation (5.23) is $-u_0$, which represents the forward speed of the vehicle.

For typical operating conditions, $a_{11} = -0.7$, $a_{12} = -0.3$, $u_0 = 1.832$ m/s, $b_1 = 0.035$.

The corresponding sensitivity model has the form:

$$\begin{bmatrix} \frac{\partial \dot{q}}{\partial \gamma} \\ \frac{\partial \dot{\theta}}{\partial \gamma} \\ \frac{\partial \dot{z}}{\partial \gamma} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 1 & 0 & 0 \\ 0 & -u_0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial q}{\partial \gamma} \\ \frac{\partial \theta}{\partial \gamma} \\ \frac{\partial z}{\partial \gamma} \end{bmatrix} +$$

$$\begin{bmatrix} \frac{\partial a_{11}}{\partial \gamma} & \frac{\partial a_{12}}{\partial \gamma} & 0 \\ 0 & 0 & 0 \\ 0 & -\frac{\partial u_0}{\partial \gamma} & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} \frac{\partial b_1}{\partial \gamma} \\ 0 \\ 0 \end{bmatrix} \delta_s(t) \qquad (5.24)$$

For the particular case where the parameter of interest, γ, is the parameter a_{11}, the sensitivity equation (Equation (5.24)) becomes:

$$\begin{bmatrix} \frac{\partial \dot{q}}{\partial \gamma} \\ \frac{\partial \dot{\theta}}{\partial \gamma} \\ \frac{\partial \dot{z}}{\partial \gamma} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 1 & 0 & 0 \\ 0 & -u_0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial q}{\partial \gamma} \\ \frac{\partial \theta}{\partial \gamma} \\ \frac{\partial z}{\partial \gamma} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ z \end{bmatrix} \qquad (5.25)$$

and this model may be represented by the block diagram of Figure 5.6.

This diagram in Figure 5.7 is a specific example of the general block diagram shown in Figure 5.2. The only coupling in this case from the system model to the sensitivity model is through the state variables $\boldsymbol{x}$. The

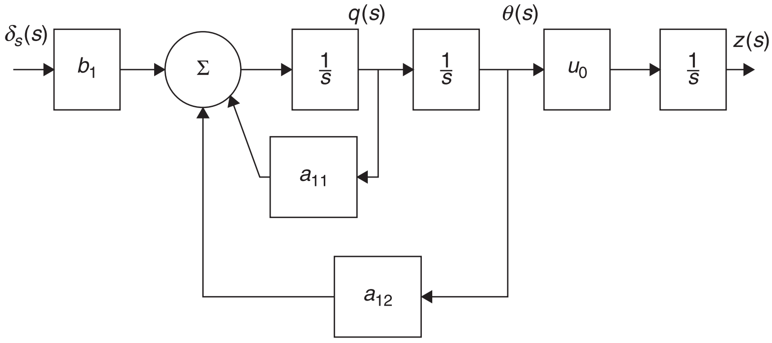


Figure 5.6 **Block diagram of the linearised state-space system model of the UUV, corresponding to Equation (5.22)**

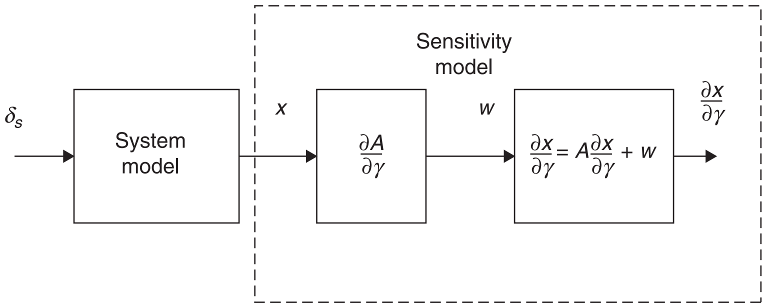

δ_s
System model
Sensitivity model
x
$\frac{\partial A}{\partial \gamma}$
w
$\frac{\partial x}{\partial \gamma} = A\frac{\partial x}{\partial \gamma} + w$
$\frac{\partial x}{\partial \gamma}$

Figure 5.7 **System model and sensitivity model for the linearised model of the UUV**

blocks $\frac{\partial B}{\partial \gamma}, \frac{\partial C}{\partial \gamma}$ and $\frac{\partial D}{\partial \gamma}$ are all made up of zero elements while the block $\frac{\partial A}{\partial \gamma}$ involves only one non-zero element $\frac{\partial a_{11}}{\partial a_{11}}$ which is unity. This makes determination of the sensitivity functions $\frac{\partial x_i}{\partial \gamma}$ for each of the state variables x_i very straightforward through simulation. For a sensitivity model-based approach, either the state-space description (as in Figure 5.6) or a transfer function approach may be used.

Figure 5.8 shows the block diagram of the sensitivity model for determination of the sensitivity functions $\frac{\partial \theta}{\partial a_{11}}$ and $\frac{\partial \theta}{\partial a_{12}}$. Figures 5.9 and 5.10 show the sensitivity functions $\frac{\partial \theta}{\partial a_{11}}$ and $\frac{\partial \theta}{\partial a_{12}}$ for the case of a step change of stern plane deflection δ_s. These two sensitivity functions are distinctively different in form, with parameter a_{11} influencing only the initial transient in the pitch θ and parameter a_{12} influencing both the initial transient and the steady-state pitch angle. This can be related to the underlying parameters of Equation (5.22) where it can be seen that, for the two coefficients considered,

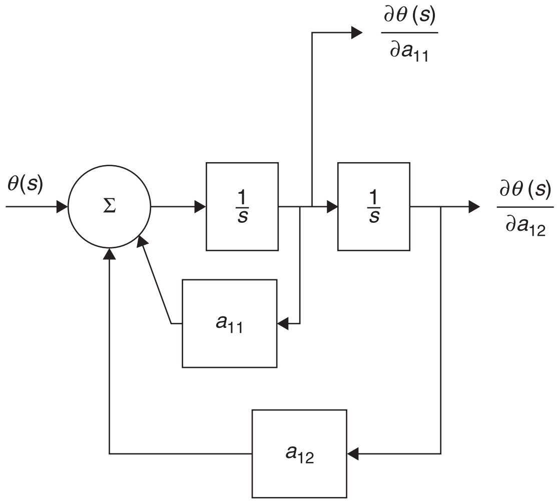

$\frac{\partial \theta(s)}{\partial a_{11}}$
$\theta(s)$
Σ
$\frac{1}{s}$
$\frac{1}{s}$
$\frac{\partial \theta(s)}{\partial a_{12}}$
a_{11}
a_{12}

Figure 5.8 **Block diagram of the part of the sensitivity model for system model of Figure 5.6 for the variable θ**

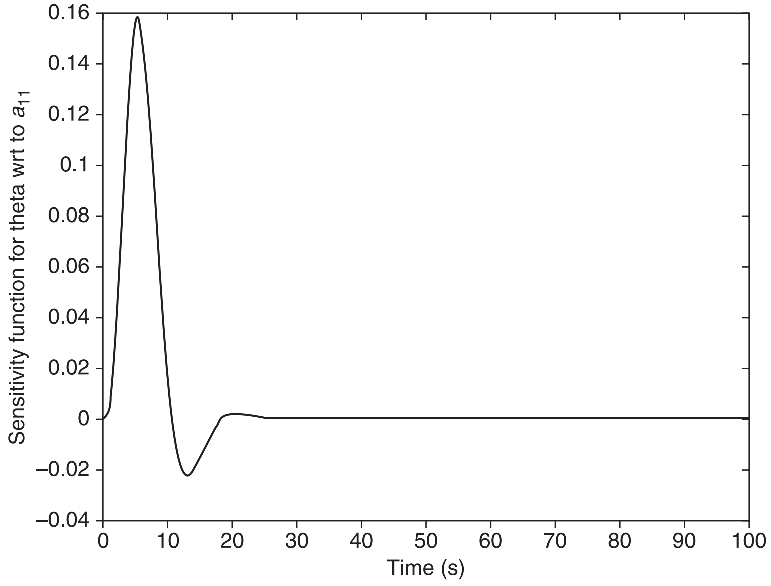


Figure 5.9 **Sensitivity function $\frac{\partial \theta}{\partial a_{11}}$ generated using the sensitivity model approach for the case of a step change of stern plane deflection**

the steady state is influenced by the parameters W, z_G and z_B while other underlying parameters involved in both coefficients, such as I_y and $M_{\dot{q}}$, only influence the transient response. For quantitative comparisons, the relative sensitivity functions $a_{11}\frac{\partial \theta}{\partial a_{11}}$ and $a_{12}\frac{\partial \theta}{\partial a_{12}}$ may be more useful than the simple un-scaled functions $\frac{\partial \theta}{\partial a_{11}}$ and $\frac{\partial \theta}{\partial a_{12}}$. For this simple example, the same conclusions could be reached by examining the form of the transfer function relating $\theta(s)$ to $\delta_s(s)$ but the use of the approach based on the sensitivity model becomes important in more complex SISO cases and for MIMO models.

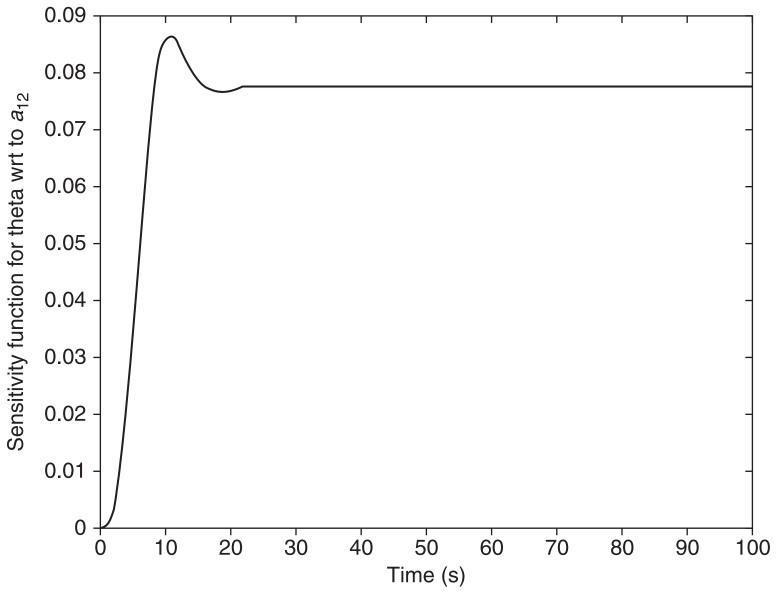


Figure 5.10 **Sensitivity function $\frac{\partial\theta}{\partial a_{12}}$ generated using the sensitivity model approach for the case of a step change of stern plane deflection**

Figure 5.11 shows results for $\frac{\partial\theta}{\partial a_{11}}$ obtained using the parameter perturbation method presented in Section 5.3 for a change of value of parameter a_{11} of 1 per cent.

Although the results obtained by the sensitivity model and parameter perturbation methods are almost identical in this case, it should be noted that the sensitivity model approach provides interesting additional information. For example, block diagram analysis shows readily that the parameter sensitivity functions for $\frac{\partial\theta}{\partial a_{11}}$ and $\frac{\partial q}{\partial a_{12}}$ have exactly the same form. Similarly, $u_0\frac{\partial\theta}{\partial a_{12}}$ and $\frac{\partial z}{\partial a_{12}}$ are the

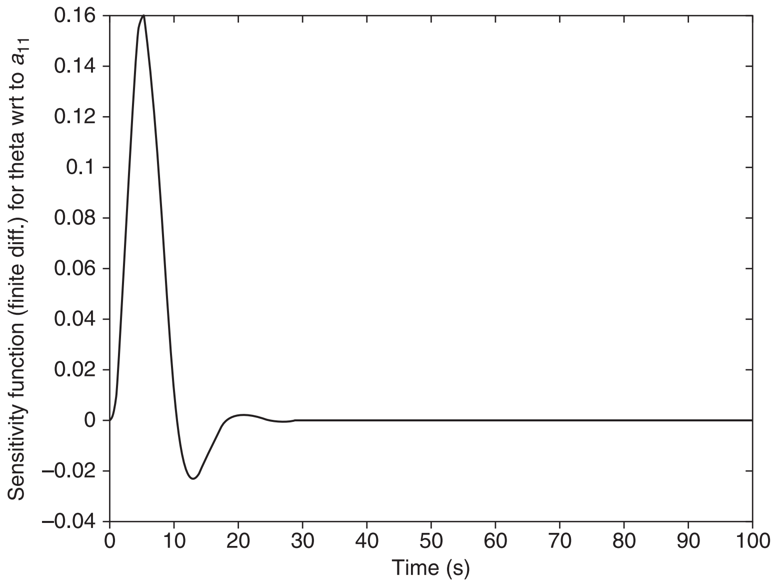


Figure 5.11 **Sensitivity function $\frac{\partial \theta}{\partial a_{11}}$ generated using the parameter perturbation approach**

same. Other relationships of this kind can be shown to exist between sensitivity functions for different combinations of parameters and system variables, and such insight could not be gathered using the parameter perturbation approach.

5.6 Sensitivity analysis using bond graphs

The bond-graph approach to sensitivity analysis is a *component-based approach*. Sensitivity models are developed for each component of the system and a system sensitivity ordinary differential equation is then derived from the component descriptions. It therefore differs fundamentally from the standard approach to parameter sensitivity analysis outlined in previous sections, which starts from the overall equations of the system.

This approach, which was used by Cabanellas *et al.* [10] for system optimisation, and by Roe and Thoma [11], provides physical insight and allows for algebraic generation of sensitivity functions as compared with purely numerical approximations. With the present-day wide availability of symbolic computational tools, this is an attractive feature of the approach.

Gawthrop [12] has extended the work of Cabanellas *et al.* [10] and has shown that *sensitivity bond graphs* can be created that have a structure identical to that of the system bond graph. This means that a sensitivity bond graph can be generated in a direct way from the system bond graph. Borutzky *et al.* [13] have pointed out that this approach is based on *pseudo-bond graph* methods, since the variables associated with the bonds are first-order sensitivities of the effort and flow with respect to a parameter and are therefore not power variables. They have suggested another approach which involves development, from the initial bond graph, with nominal parameters, of an *incremental bond graph* for increments of power variables due to small parameter changes [13]. Another interesting development has been the proposal by Dauphin-Tanguy and Kam [14] for *uncertainty bond graphs* for power variables in robustness investigations. Borutzky *et al.* [13] have demonstrated that uncertainty bond graphs also allow parameter sensitivities to be determined.

5.7 Sensitivity analysis in inverse simulation

In principle, sensitivity analysis can be carried out for all methods of inverse simulation. For the iterative methods of inverse simulation outlined in Section 4.2., this involves using finite changes of parameters of the forward model and repeated solutions. The problems of accuracy mentioned in Section 5.3 therefore apply and, in addition, there are inevitable problems of computation time. The feedback approach to inverse simulation has obvious advantages because the sensitivity model approach can be applied directly.

5.7.1 Inverse sensitivity using the feedback approach to inverse simulation

Figure 5.12 shows the block diagram for inverse simulation based on the use of feedback system principles, as described in Section 4.2.5.

In the case of a linear model, *G(s)*, which involves a set of parameters ***q***, then for a given parameter q_i:

$$\frac{\partial w(s)}{\partial q_i} = -\frac{C(s)v(s)C(s)\dfrac{\partial G(s)}{\partial q_i}}{(1+C(s)G(s))^2} \tag{5.26}$$

$$= -w(s)\frac{C(s)}{(1+C(s)G(s))}\frac{\partial G(s)}{\partial q_i} \cong -w(s)\frac{1}{G(s)}\frac{\partial G(s)}{\partial q_i}$$

The sensitivity of the signal *w*(*s*), which represents the inverse solution, to the parameter q_i is therefore found by passing the signal through a block having transfer function:

$$-\frac{1}{G(s)}\frac{\partial G(s)}{\partial q_i}$$

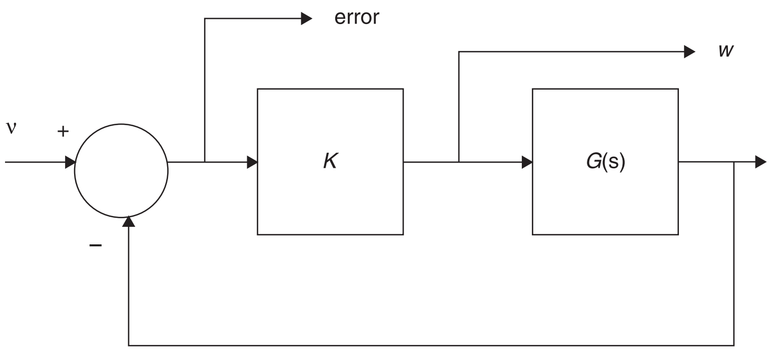


Figure 5.12 Block diagram illustrating the feedback system approach to inverse simulation

This transfer function depends entirely on the characteristics of the forward model $G(s)$ and is easily found using analytical methods.

5.7.2 Example: inverse sensitivity for a simple transfer function model

Consider a simple linear model:

$$G(s) = \frac{K(1 + sT_1)}{1 + sT_2} \tag{5.27}$$

The structure of the sensitivity model for simultaneous determination of the sensitivity functions for the inverse model with respect to the parameters of the forward model is shown in Figure 5.13 and the filter transfer functions for the parameters of the model $G(s)$ are as follows:

$$F_K(s) = \frac{1}{G(s)} \frac{\partial G(s)}{\partial K} = \frac{1}{K} \tag{5.28}$$

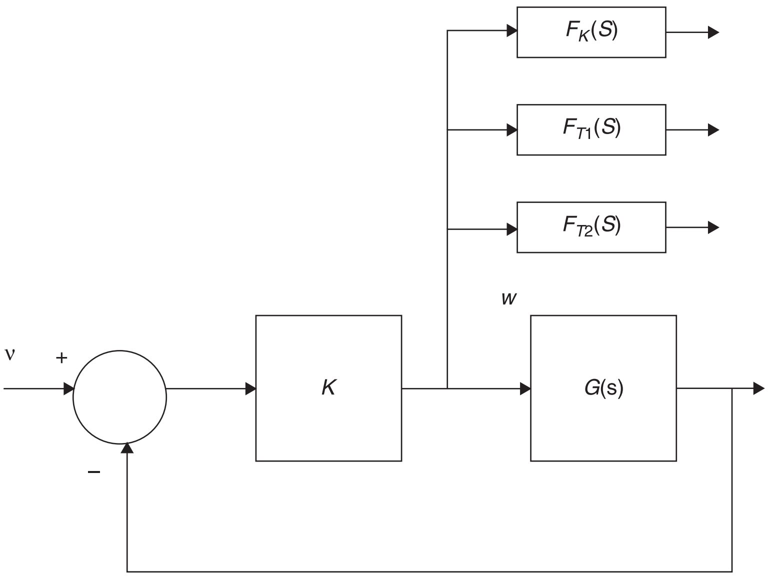


Figure 5.13 Block diagram for inverse sensitivity using a sensitivity model approach

$$F_{T1}(s) = \frac{1}{G(s)}\frac{\partial G}{\partial T_1} = \frac{s}{1+sT_1} \tag{5.29}$$

$$F_{T2}(s) = \frac{1}{G(s)}\frac{\partial G}{\partial T_2} = -\frac{s}{1+sT_2} \tag{5.30}$$

For the case where the parameters have values $K_1 = 10$, $T_1 = 1$ s, $T_2 = 2$ s and the desired model output signal given in Figure 5.14, the input found from inverse simulation to produce that output is given in Figure 5.15. It can be seen that for a steady output of one, the required steady input has a value of approximately 0.1, as would be expected from the value of gain factor K_1 of 10.0. The peak values of the input (0.2 and –0.1) and the associated transients in the input signal are needed to overcome the first-order time lag characteristics of the model.

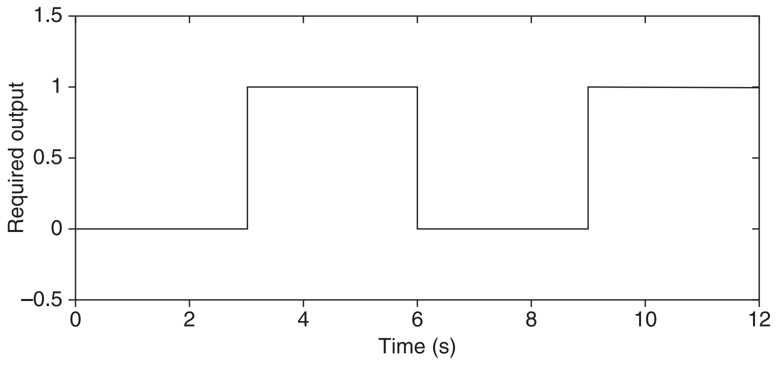


Figure 5.14 **Desired model output (dimensionless) plotted against time (s)**

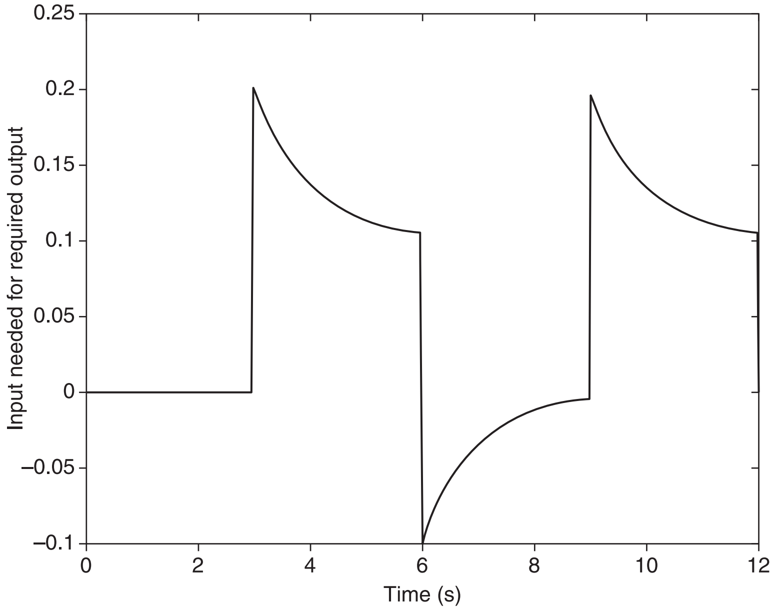


Figure 5.15 **Input (dimensionless) generated using inverse simulation to produce the model output shown in Figure 5.14. The horizontal axis represents time (s)**

The sensitivity function for the parameter T_2 found using the sensitivity model approach is shown in Figure 5.16. Although the corresponding result obtained by the finite difference approach for a perturbation ΔT_2 of magnitude 0.01 s is so close as to be indistinguishable from that shown in Figure 5.16, the successful application of the perturbation method does require careful choice of ΔT_2; this usually involves a trial-and-error approach and therefore requires several simulation runs. The sensitivity function shows, as would be expected from physical reasoning, that the parameter T_2 influences the transient part of the inverse solution but does not affect the steady states.

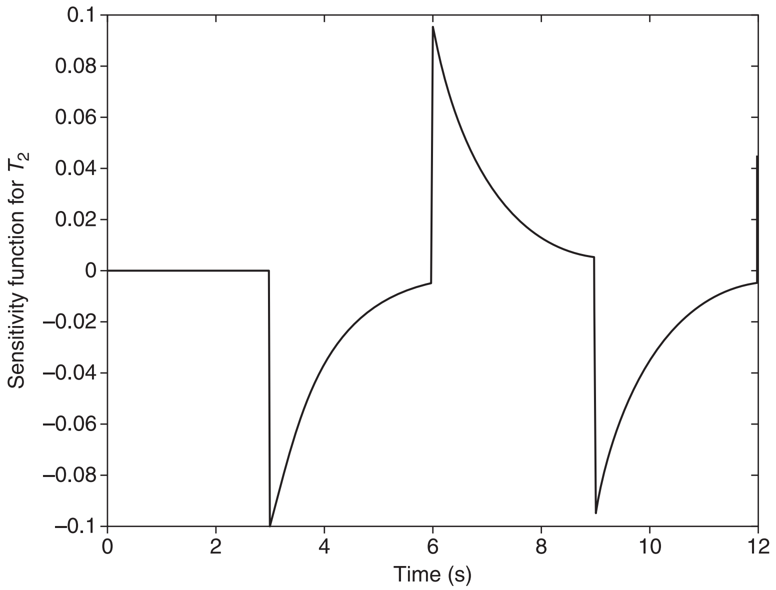


Figure 5.16 **Sensitivity function for parameter T_2 as a function of time found using the sensitivity model approach**

5.7.3 Example: sensitivity analysis for inverse simulation of a linearised model of the UUV

The linearised dynamics for diving motion of the underwater vehicle model of Appendix A1 involves a third-order system with the following equation:

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 1 & 0 & 0 \\ 0 & -u_0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix} \delta_s(t) \tag{5.31}$$

This is the linearised model already considered in the context of sensitivity analysis in Section 5.5.1. As already noted, the state variables q, θ and z are the pitch rate, pitch angle and vertical displacement, as given in Appendix A1, and δ_s is the stern plane deflection and for typical operating conditions $a_{11} = -0.7$, $a_{12} = -0.3$, $u_0 = 1.832$ m/s and $b_1 = 0.035$.

If we require a sinusoidal trajectory in terms of the pitch angle, θ, the stern plane deflection time history determined from inverse simulation using simple proportional plus rate feedback is as shown in Figure 5.17.

The high frequency oscillation superimposed on the stern plane deflection is due to the presence of lightly damped poles in the feedback system used for inverse simulation. Despite the presence of these poles, the application of this stern plane deflection signal to the forward model of the UUV gives an output which is almost identical to the required sinusoid (Figure 5.18).

Application of the technique outlined in Section 5.7 for determination of the sensitivity of the inverse solution to parameters of the given model, such as a_{11} and a_{12}, involves the application of appropriate filters $-\frac{1}{G(s)}\frac{\partial G(s)}{\partial q_i}$ to the

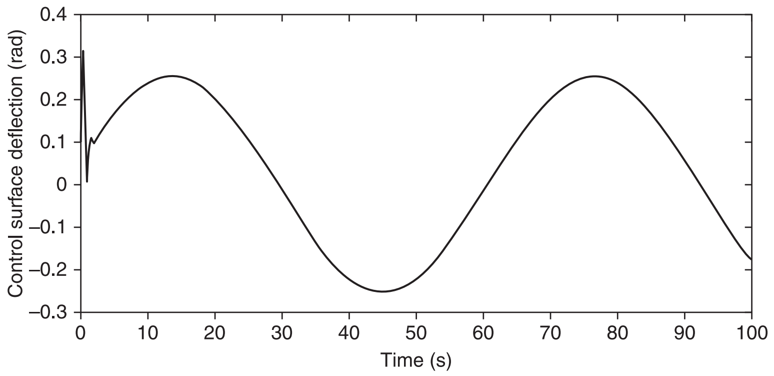


Figure 5.17 Stern-plane input signal found from inverse simulation for required trajectory involving sinusoidal change of pitch angle

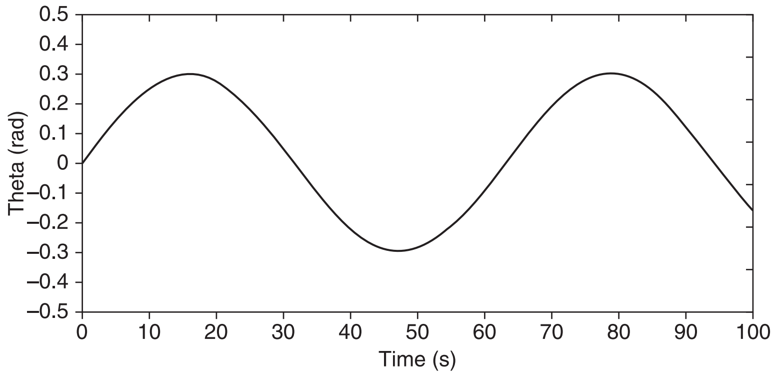


Figure 5.18 Pitch angle record found from application of stern-plane deflection signal of Figure 5.17 to the forward model of the UUV

simulated signal which represents the required control surface input $\delta_s(t)$.

In the case of the parameter a_{11}, the filter can be shown to have the transfer function:

$$\frac{-s}{s^2 - a_{11}s - a_{12}} \tag{5.32}$$

while for a_{12} the transfer function is:

$$\frac{-1}{s^2 - a_{11}s - a_{12}} \tag{5.33}$$

Results obtained, using this method of sensitivity analysis for the inverse simulation model, are shown in Figures 5.19 and 5.20. It can be seen that (neglecting the short initial transient which is an artefact of the inverse simulation method being used) the parameter sensitivity function $\frac{\partial \delta_S}{\partial a_{11}}$ is dominated by a cosine function of amplitude approximately 0.07 and frequency the same as that of the required pitch change signal. The sensitivity function $\frac{\partial \delta_S}{\partial a_{12}}$ is sinusoidal in

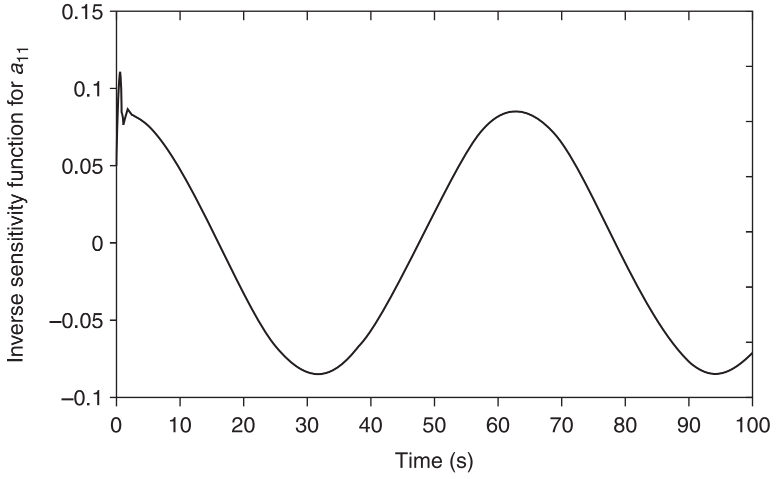


Figure 5.19 **Sensitivity function found from sensitivity model for parameter a_{11}**

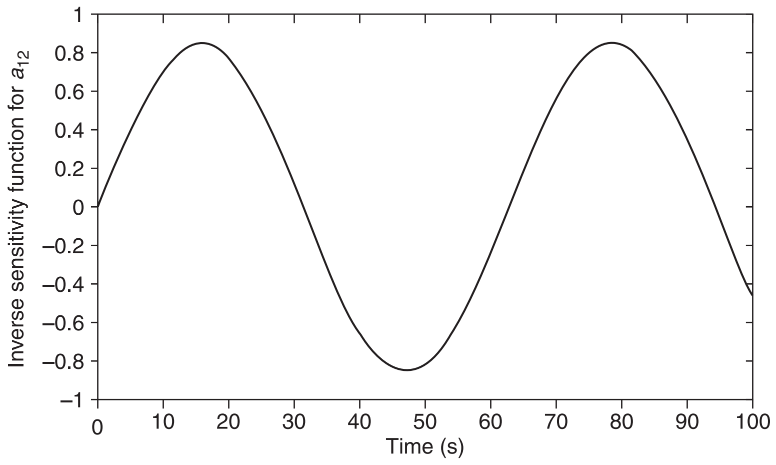


Figure 5.20 **Sensitivity function found from sensitivity model for parameter a_{12}**

form and identical in frequency to that for a_{11} with a peak value of approximately 0.82. As would be expected from the form of the filters given above, the sensitivity function for a_{11} is shifted in phase compared with that for a_{12} by 90° and these two parameters therefore each have greatest influence on the stern plane deflection signal at different times. The influence of each of the parameters can be compared quantitatively using the relative sensitivity measures $a_{11}\frac{\partial \delta_S}{\partial a_{11}}$ and $a_{12}\frac{\partial \delta_S}{\partial a_{12}}$. On that basis, the stern plane deflection signal may be shown to have a maximum sensitivity to parameter a_{12} which is approximately five times larger than the maximum sensitivity to a_{11}. Such information is potentially useful when considering actuator limits and when carrying out control system optimisation studies. Similar information on inverse sensitivity could be derived in the same way for the other parameters of the model.

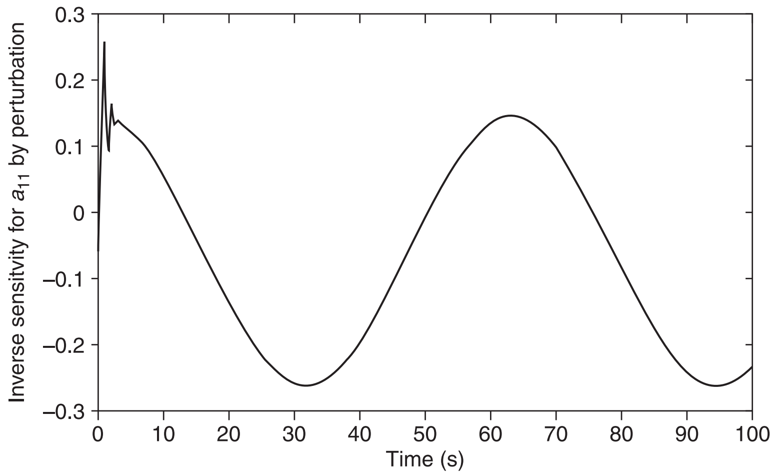


Figure 5.21 **Sensitivity function found using finite difference approximation approach for the case of parameter a_{11}**

For comparison with the results of Figures 5.19 and 5.20, Figure 5.21 and 5.22 show sensitivity functions for the parameters a_{11} and a_{12} found directly from the inverse simulation results using a perturbation approach. It can be seen that the sensitivity functions found by the perturbation method and the sensitivity model approach are almost identical in both cases, apart from the initial transient in the sensitivity function for a_{11}. That transient is, of course, associated with the choice of feedback loop gain factors and is therefore an artefact of the feedback system approach to inverse simulation, as mentioned above. As in the case of sensitivity analysis for conventional forward models, it is believed that additional physical insight and computational simplicity once again give the sensitivity model approach advantages when compared with perturbation methods.

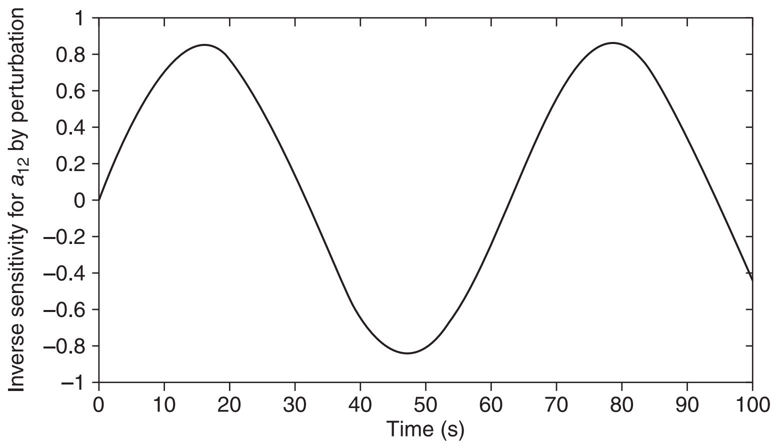


Figure 5.22 **Sensitivity function found using finite difference approximation approach for the case of parameter a_{12}**

5.8 References

[1] Tomovic, R. (1963) *Sensitivity Analysis of Dynamic Systems*, McGraw-Hill, New York NY, USA.

[2] Frank, P.M. (1978) *An Introduction to System Sensitivity Theory*, Academic Press, London, UK.

[3] Kokotovic, P.V., O'Malley, R.E. and Sannuti, P. (1976) 'Singular perturbations and order reduction in control theory – an overview', *Automatica*, Vol. 12, No. 2, pp. 123–32.

[4] Tomovic, R. and Vukobratovic, M. (1972) *General Sensitivity Theory*, American Elsevier, New York NY, USA.

[5] Kokotovic, P.V. (1964) 'Method of sensitivity points in the investigation and optimization of linear control systems', *Automation and Remote Control*, Vol. 25, pp. 1670–6.

[6] Rosenwasser, E. and Yusupov, R. (2000) *Sensitivity of Automatic Control Systems*, CRC Press, Boca Raton FL, USA.

[7] Bode, H.W. (1945) *Network Analysis and Feedback Amplifier Design*, Van Nostrand-Reinhold, Princeton NJ, USA.

[8] Horowitz, I.M. (1963) *Synthesis of Feedback Systems*, Academic Press, New York NY, USA.

[9] Wilkie, D.F. and Perkins, W.R. (1969) 'Generation of sensitivity functions for linear systems using low-order models', *IEEE Transactions on Automatic Control*, Vol. 14, pp. 123–9.

[10] Cabanellas, J.B., Felez, J. and Vera, C. (1995) 'A formulation of the sensitivity analysis for dynamic systems optimisation based on pseudo bond graphs', in Cellier, F.E. and Granda, J.J. (eds), *Proceedings of the 1995 International Conference on Bond Graph Modeling and Simulation*, ICBG M'95 of Simulation Series, Vol. 27, pp. 135–44, Society for Computer Simulation, La Jolla CA, USA.

[11] Roe, P.H. and Thoma, J.U. (2000) 'A new bond graph approach to sensitivity analysis', in Troch, I. and Breitenecker, F. (eds), *Proceedings MATHMOD 2000, The Third IMACS Symposium on Mathematical Modelling, Vienna, Austria*. pp. 135–44, ARGESIM, Vienna, Austria.

[12] Gawthrop, P.J. (2000) 'Sensitivity bond graphs', *Journal of the Franklin Institute*, Vol. 337, pp. 907–22.

[13] Borutzky, W., Dauphin-Tanguy, G. and Kam, C. (2006) 'Relations between two bond graph approaches to sensitivity and study of robustness', *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 12, No. 2–3, pp. 141–57.

[14] Dauphin-Tanguy, G. and Kam, C.S. (1999) 'How to model parameter uncertainties in a bond graph framework', in Horton, G., Möller, D. and Rüde, U. (eds), *Simulation in Industry, 11th European Simulation Symposium, ESS'99*, pp. 121–5, SCS Publishing, Erlangen, Germany.

6

Experimental modelling: system identification, parameter estimation and model optimisation techniques

Abstract: This chapter describes the role of system identification and parameter estimation methods and related optimisation techniques within the model development process. These techniques are central to what is termed 'experimental modelling' and emphasis is given to issues of identifiability and experimental design. Applications considered are drawn mainly from the field of helicopter modelling from flight test data. Optimisation techniques considered include simulated annealing and genetic algorithm methods as well as gradient-based and search-based techniques. Model structure estimation is approached through physical understanding of the system and the use of genetic programming methods.

Key words: identification, parameter estimation, identifiability, experimental design, test signal, optimisation, flight mechanics, helicopter.

6.1 The use of system identification and optimisation techniques in the development of physically based dynamic models

It has been emphasised in Chapter 1 that models must be appropriate for their intended purpose. Models are never unique and their development is an iterative process, involving initial formulation and testing, followed by repeated modification and retesting. The form of model used at a particular stage in a project must therefore take account of the objectives, the amount of detail needed at that stage and uncertainties in the information about the real system. In some situations, particularly when working with existing systems or sub-systems, there may be a need for experimental investigations before a detailed physically based quantitative model can be developed.

System identification and *parameter estimation* are established techniques involving use of measured response information from a real system to develop a mathematical or computer-based dynamic model. In this approach the model usually has a general form involving ordinary differential or difference equations and an associated set of parameters that have to be estimated. In general, the structure (as defined by the number of differential equations and any associated algebraic relationships) also involves uncertainties and the most appropriate form may have to be established using measured response data. Techniques of system identification and parameter estimation are reviewed in many textbooks, such as those by Ljung [1], Söderström and Stoica [2], Nelles [3] and Raol *et al.* [4].

Many approaches to identification and system parameter estimation involve optimisation of a specified *cost function* involving the difference (error) between a model response and the corresponding measured variable. Decisions about the most appropriate structure for the model usually require background knowledge and physical understanding of the system, as well as examination of available experimental data and consideration of the intended application. Once an initial structure has been established and uncertainties have been critically assessed, the parameters of the model can be estimated, usually iteratively, through a specific cost function and optimisation method such as *least-squares minimisation*. The iterative processes of parameter adjustment continue until the responses of the model match those of the system to some predefined level. For problems that are nonlinear in the parameters and involve many unknowns, one important issue is the potential presence of large numbers of local maxima and minima in the surface that represents the cost function. Therefore, it is possible that many sets of locally 'optimal' parameters may be found from test data and care should be taken to find the set that corresponds to the global optimum.

The textbook by Raol *et al.* [4] provides a useful review of least-squares methods in the context of modelling, system identification and parameter estimation. The treatment of optimisation methods presented in that book establishes links between the properties of classical gradient-based optimisation techniques and methods commonly used in the application of system identification and parameter estimation to engineering systems. These include the *generalised least-squares* and *nonlinear least-squares* methods, and techniques such as the *equation-error* and *output-error* methods that have been used extensively in the identification of physically based models arising in the modelling of fixed-wing aircraft, helicopters, ships and underwater vehicles (see e.g. [4], [5] and [6]).

Model structure optimisation involves optimisation of complexity, since the number of separate equations and the number of adjustable parameters, which provide a crude measure of complexity, depend on the chosen structure. Also, with more parameters, a model becomes more flexible since the number of forms of behaviour that could be exhibited increases. A model that is too simple will not capture the behaviour of the system and will give poor predictions. In addition, if the data available for parameter estimation and subsequent testing of the model are inadequate, even a relatively complex model may perform badly. Thus, the complexity of a model must always be appropriate for the intended task. Optimising the complexity of a model is closely linked to questions of model validation (see Chapter 7) which involves critical assessment of the model performance for a specific application.

One approach that is well suited to the system identification of physically based continuous system models has been presented by Knudsen (e.g. [7] and [8]). This is based on parameter sensitivity information (as discussed in Chapter 5) which is used in selecting model structures, for experimental design and for the validation of models identified from experimental data. The essential point of incorporating parameter sensitivity information into the identification approach is that accurate estimation of any parameter requires that the cost function upon which the estimation is based should be sensitive to that parameter and the most sensitive parameters are likely to be the ones that are estimated most accurately. Comprehensive knowledge of parameter sensitivities thus provides important information for assessing identification results. A MATLAB™ toolkit 'Senstools' has been developed by Knudsen and is available for downloading [9].

It is important that the performance of a model should be assessed using a 'test' data set that is not the same as the 'training' data set that was used in estimating the structure and parameters of that model. In this way, the *generalisation* ability of the identified model may be assessed. 'Generalisation' describes the capability of the model to predict system outputs for experimental situations that are not exactly the same as those used at the development stage.

The terms 'under-fitting' or 'over-fitting' are widely used. If the model is relatively simple but cannot match observed behaviour, the situation may involve under-fitting and the model structure should be reviewed. If a relatively complex model is used and the training appears satisfactory, but the generalisation is poor, there may be over-fitting. In such cases the cause of the problem may be bias on the estimated values of model parameters due to noise on the data used for identification, or an inappropriate structure.

Physically based dynamic models used in practical engineering applications are usually *nonlinear in the parameters*. In such cases iterative methods must be used to find the set of parameters to give model responses that match experimental data, even if the model structure itself is linear. Many nonlinear optimisation techniques and methods for iterative solution of nonlinear equations have been developed and detailed information about the relevant algorithms may be found elsewhere (see e.g. [3] and [10]).

Nonlinear optimisation methods can be classified as local or global [3]. Although they converge to local optima, local methods often settle at points close to the initial parameter set, particularly if search directions are obtained from first- and second-order derivatives. Such algorithms tend to become stuck at local minima or maxima and an extremum elsewhere in the parameter space may be neglected. Global nonlinear optimisation methods may overcome this difficulty and rely on the inclusion of random components that help the algorithm to avoid becoming trapped. Well-known global optimisation techniques include *simulated annealing* (SA) (e.g. [11] and [12]) and *evolutionary* algorithms such as the *genetic algorithm* (GA), the principles of which are discussed in more detail in the books by Holland [13] and Goldberg [14]. Further information about the techniques of simulated annealing and evolutionary computing applied to system modelling may be found in Sections 6.4 and 6.5.

6.2 Applications of conventional methods of system identification and parameter estimation to physically based models

As has already been stated, system identification in its most general form may be defined as the quantitative estimation of the structure and parameters of a model using experimental measurements from the corresponding real system. Traditionally it is applied to the development of self-adaptive control systems and other online and self-tuning control techniques. However, it may also be regarded as a set of analytical and computational tools that provide additional insight at various stages in the development of physically based models.

In model-based design and development, system identification can be helpful for choosing a suitable model structure, perhaps from several different candidate descriptions. Identification can then provide estimates of parameters for that structure.

How identification and parameter estimation techniques are used depends on the intended application. Different models and methods may be appropriate depending on the purpose of the model. For example, for many control system applications, linear forms of model may be used since control system design methods frequently require linearised plant models. However, when system identification and parameter estimation methods are used within the process of simulation model validation, nonlinear models are often essential.

System identification and parameter estimation is a mature field and details of methods of identification such as the equation error, output error or maximum-likelihood methods are not included in this book. Well-documented software for system identification and parameter estimation is available from a variety of sources, including well-established MATLAB® toolboxes (e.g. [15], [16] and [9]). Instead of repeating information that is readily available elsewhere, more attention is given here to ways in which such established methods may be applied to the development, validation and application of models in a practical engineering context.

6.2.1 Issues of identifiability

The precision of any parameter estimate may be expressed in terms of the variance of the estimate and this depends both on the experiment used for identification and the estimation technique. Often the objective is to obtain unique and reliable estimates of all of the parameters of a model. It is important to investigate whether or not this is theoretically feasible for a given model structure and a given experiment. The concept of *identifiability* is central to issues of this kind and tests for identifiability provide an indication of potential problems before an identification method is chosen or issues such as experimental design are considered.

Global or *structural unidentifiability* arises if a model has an excess of parameters so that some specific parameters cannot be estimated uniquely for any possible experimental design. This depends on the model structure and not on values of parameters or the design of the identification experiment. It arises when a model has too many parameters to allow all of them to be found for any possible input stimulus.

Structural identifiability is only a minimal necessary condition for obtaining unique estimates of model parameters. Bellman and Åström [17] were among the first to discuss this issue. Although their findings were presented in the context of biological compartmental models, the results are applicable to other problems. They showed that classical transfer function theory could be used as a basis for the investigation of identifiability. If each coefficient of the transfer function matrix is expressed as a combination of the unknown parameters, a set of nonlinear equations is defined and the model is identifiable in a global sense if these equations have a unique solution.

Pathological or *numerical unidentifiability* arises when a structurally identifiable model is being used with an experimental data set that is inappropriate for the application. This could be because the length of the available record is too short compared with the dominant time constants or the period of oscillatory components of the response. It could also arise if the measured response data is very heavily corrupted by noise.

Beck and Arnold [18] have shown that model parameters can be estimated only if the parameter sensitivity coefficients for the output variable with respect to each parameter are linearly independent over the range of observations. Problems of numerical unidentifiability may be deduced, in simple cases, directly from the time histories of sensitivity functions (as defined in Chapter 5). The problem can also be investigated more systematically by examining properties of the sensitivity matrix $\boldsymbol{X}$ and the closely associated *parameter information matrix* $\boldsymbol{M} = \boldsymbol{X}^T\boldsymbol{X}$. This type of analysis allows interdependencies to be investigated that are more complex than those found by direct inspection of sensitivity function time histories. Pathological unidentifiability is linked to linear dependence of the columns of ***X*** and this is reflected in the determinant of matrix ***M*** or in the condition number of the matrix (the ratio of the largest eigenvalue of ***M*** to the smallest eigenvalue of ***M***). If the condition number is large, or if the determinant is small, the confidence region for the estimates is large and the parameter estimates are therefore not well defined.

The matrix $\boldsymbol{M}^{-1}$ is also important and is known as the *dispersion matrix*, commonly denoted by ***D***. The determinant of ***D*** can be shown to be a useful indicator of numerical (or pathological) unidentifiability [18].

Correlations between pairs of parameters can be investigated using the *parameter correlation matrix* ***P***. This matrix is commonly defined in terms of its elements:

$$p_{ij} = \frac{m_{ij}^{-1}}{\sqrt{m_{ii}^{-1} m_{jj}^{-1}}} \tag{6.1}$$

where p_{ij} is the element of ***P*** in row *i* and column *j* and m_{ij}^{-1} is the element of $\boldsymbol{M}^{-1}$ in row *i* and column *j*. The diagonal elements of the matrix ***P*** have values which are unity and the off-diagonal elements all lie between –1 and 1. Conditions close to unidentifiability arise if the modulus of one or more of the off-diagonal terms is close to unity, with a value of 0.95 being regarded as a limiting value [18]. Small values of the off-diagonal elements of ***P*** indicate that the parameters are essentially decoupled.

6.2.2 Design of experiments and the selection of test-input signals

In designing experiments and choosing test signals for system identification and parameter estimation, we must have a quantitative basis for comparisons. It is also assumed that the estimator is efficient (e.g. [19]) and that these experimental aspects are independent of the estimator. Test signal design can involve quantities such as the parameter information matrix and the dispersion matrix, both of which have theoretical origins in the *Cramer-Rao bound* [18], through which the variance of parameter estimates may be related to elements of the dispersion matrix $\boldsymbol{D}$. Since this is based on the parameter sensitivity matrix $\boldsymbol{X}$, which depends on measurements, the elements of $\boldsymbol{D}$ can be derived from tests. In general terms, inputs giving a dispersion matrix with small elements are to be preferred over inputs producing large values and this leads to test input design algorithms that minimise appropriate functions of the dispersion matrix or of the parameter information matrix.

The sensitivity matrix $\boldsymbol{X}$ and the parameter information matrix $\boldsymbol{M}$ thus provide a basis for measures of the quality of an experiment using relationships which are of the general form:

$$J = f(\mathbf{M}) \tag{6.2}$$

where f is an appropriate scalar function. One widely used criterion for experiment design is the so-called *D-optimal criterion* [20] which involves the dispersion matrix and has the form:

$$J_D = \det(\mathbf{D}) = \det(\mathbf{M}^{-1}) \tag{6.3}$$

This criterion gives a test signal that places equal emphasis on the estimation of all of the parameters. In cases where a subset of parameters is more important, use of a *truncated D-optimal design criterion* has been advocated [21], and this has the form:

$$J_{Dt} = \det(\boldsymbol{M}_{ii}^{-1}) \tag{6.4}$$

where $\boldsymbol{M}_{ii}$ is a sub-matrix of the full information matrix referring only to the *i* parameters of interest. Use of this truncated criterion involves elements of the sensitivity matrix $\boldsymbol{X}$ which themselves depend on values of model parameters. Thus it is only possible to use the criterion to investigate and compare experimental designs in a general way. Use of the criterion to generate a truly optimal experiment is impossible because, to do that, exact parameter values for the system under test would have to be known *a priori*.

Many accounts exist of applications in which the concepts of identifiability and experiment design have been used to good effect. Much early research on identifiability analysis involved biomedical applications, but these techniques have also been applied in many other fields. One typical application has been described by Keskitalo and Leiviskä [22], involving calibration of models of activated sludge systems for wastewater treatment. Experimental estimation of model parameters is essential in matching activated sludge models to real processes, but difficulties arise because the model structure does not allow unique estimates to be obtained for all the parameters and the available measurements are usually of low quality. Experience suggests that models of this kind require regular recalibration and Keskitalo and Leiviskä have proposed the development of a more automated approach, using identifiability analysis combined with global optimisation methods, which avoids much trial-and-error work [22].

Although the activated sludge model is typical of problems encountered with environmental system models and process system models, where there is often very limited scope of experimentation, other physical systems may offer more flexibility in the generation of test data sets for identification and eventual model validation.

6.3 System identification and parameter estimation applied to helicopter flight mechanics models

Applications of system identification and parameter estimation techniques to problems of helicopter flight mechanics modelling and control are important, especially for the validation of models developed using physical laws and principles. Conventional simulation models derived on a physical basis and using wind-tunnel data, taken alone, seldom provide a basis for the development of usable models. Extensive flight testing programmes using prototype vehicles are still essential as part of the certification procedures for new aircraft, but these test programmes are time-consuming and costly. Changes required in the hardware following prototype testing are often linked directly to deficiencies in the underlying models used for the initial stages of vehicle design.

Figure 6.1 illustrates an application of identification techniques and comparisons with conventional simulation results for a well-established helicopter flight mechanics model (involving the DLR simulation program SIMH [23]). The uppermost traces show test input time histories for the lateral and longitudinal cyclic controls, as measured in a flight experiment on a BO105 helicopter [24]. The records below are for roll rate and pitch rate predicted by the SIMH simulation model (dashed lines) together with the corresponding variables measured in the flight test programme (continuous lines). Some significant differences between the simulation results and the measured results are immediately evident. The bottom two sets of results show the same flight test measurements along with predicted outputs for the same variables for a model based on

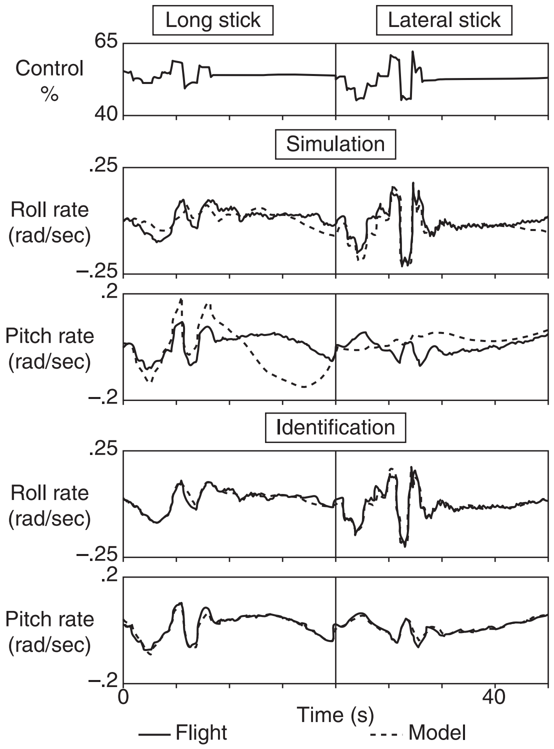


Figure 6.1 **Flight data (continuous lines) and corresponding model responses (dashed lines) for typical test inputs. Results under the heading 'Simulation' show roll rate and pitch rate variables for a physically based model in response to measured inputs together with the corresponding measured variables. The results under the heading 'Identification' relate to a reduced model structure with parameters estimated from flight data, with model predictions compared with the roll and pitch rate variables from flight. (The original version of this figure was published by the Advisory Group for Aerospace Research and Development, North Atlantic Treaty Organisation (AGARD/NATO) in AGARD Advisory Report 280 'Rotorcraft System Identification', in September 1991)**

parameters estimated from flight tests. The difference between the measured results and model predictions are much smaller in this case, showing the increase in accuracy possible when using appropriate parameter estimates.

Although results shown in Figure 6.1 are encouraging, it should be noted that helicopters present special difficulties in terms of the use of identification methods. For linearised multi-input multi-output (MIMO) models of the complete vehicle, it is normal to be faced with test records that are short compared both with the dominant time constants and periods of the dominant oscillatory modes. Also, these models involve many parameters and a wide range of frequencies, and measurements can involve high levels of noise. Along with the short test records usually available, these are not desirable factors for identification.

Within the aircraft industry, the benefits of system identification relate mainly to reduction in flight testing for certification of new designs and for fine-tuning of the vehicle's agility and handling qualities. Identification methods may also be useful for improving engineers' confidence in physically based models used in design and for reducing levels of uncertainty. Estimation of parameters from flight is now an increasingly important part of prototype testing and is especially relevant for some aerodynamic stability and control parameters. Although flight testing is costly, additional tests carried out specifically for system identification and parameter estimation purposes could allow essential design modifications to be made through virtual prototyping methods more quickly, more efficiently and at a lower cost than by using traditional approaches.

In the late 1980s and 1990s, active control technology began to be applied to helicopters. Essentially, this is the fly-by-wire approach that had, by then, already been accepted for civil and military fixed-wing aircraft. The improvements in performance and capabilities expected from active-control technology could only be achieved using accurate and proven mathematical models (e.g. [25]). The publication in 1989 and in 1994, in the USA, of revised handling qualities requirements for military helicopters [26] provided an additional stimulus to these developments in helicopter flight control and created new interest in multivariable control system analysis and design methods. Enhanced performance requirements and developments in materials and rotor technology have produced improvements in vehicle characteristics which mean that much enhanced performance is possible but use of traditional loop-by-loop control design methods may present difficulties and the expected performance gains may not be realised in practice.

Multivariable control system design techniques, which exploit the multivariable structure of vehicles such as helicopters, have been applied successfully in several investigations (e.g. [27], [28] and [29]). However, ensuring accuracy in the MIMO models used for active control system design is challenging. Models must perform adequately over a defined range of frequencies and for a range of manoeuvre amplitudes. For example, high-bandwidth model-following flight control systems may incorporate improved feed-forward control pathways to provide improved agility for large and rapid manoeuvres, but such an approach requires accurate models of the vehicle [27]. System identification is also increasingly important for validation of ground-based simulators for rotorcraft of all types and highly accurate mathematical models are needed for simulators that are to be used for pilot training (see e.g. [25]).

6.3.1 Some examples of rotorcraft applications

Before the 1990s, most published accounts of system identification techniques applied to helicopters involved time-domain methods of identification. However, the use of frequency-domain methods is now seen to have advantages. In this case the measured response data is transformed first into the frequency domain using an appropriate implementation of the Fast Fourier Transformation. This allows attention to be focused on particular parts of the frequency range, and data lying outside the areas of interest can be given less emphasis or discarded. Thus, for the identification of six-degrees-of-freedom rigid-body models, the rotor degrees of freedom, which involve higher frequencies, can be excluded. Conversely, for the identification of rotor dynamics, the lower frequencies involving the rigid-body response can be excluded. This procedure allows, in a sense, a form of model reduction within the identification process [30].

Details of a frequency-domain approach to helicopter system identification may be found in a paper by Black and Murray-Smith [31] and this approach was one of several methods of helicopter system identification successfully used by the NATO-supported AGARD Flight Mechanics Panel Working Group WG18 for work leading to the preparation of the AGARD Advisory Report 280 [32] on Rotorcraft System Identification and the associated Lecture Series volume [33]. Frequency-domain methods have been used widely in the years since publication of that AGARD report, especially using the now widely available CIFER software developed by Dr Mark Tischler and his colleagues at the US Army Aeroflightdynamics Directorate [34].

Although it may be stated, without question, that identification and parameter estimation techniques are potentially important for helicopter development and flight testing, it has to be accepted that the benefit of these tools has not yet been fully realised. Many of the difficulties are associated with issues of robustness and these have been classified under the following headings [35]:

1. robustness and reliability of *a priori* information;
2. robustness of the identified model structure;
3. robustness of estimated parameters; and
4. robustness of the resulting overall model.

In terms of these robustness issues, the properties of different estimators are probably less important than questions of identifiability, the quality of measured data and experimental design. However, an understanding of the properties of different estimators is essential if they are to be applied appropriately.

Klein has provided a useful account of identification techniques for aircraft system identification [36] and the theoretical properties of different estimators. For example, the maximum likelihood approach, when applied in its most general form, allows parameters to be estimated for a linearised aircraft model from flight data involving measurement noise and process noise, such as unmodelled disturbances in the form of gusts. Less general forms of output-error method involve assumptions that only the measurements are corrupted by noise. This means that estimates obtained using output-error methods can be degraded in the presence of unmeasured and unmodelled disturbances giving relatively poor estimates with large variances. Similarly, there are problems with equation-error methods, since these are not only affected by process noise and by measurement noise, but can produce biased estimates even if all the measurement and process noise components have zero mean.

Although the values of the variances associated with parameter estimates are useful indicators of robustness, it is important to understand that any comparison of variances obtained for different model structures is impossible. Checks of residuals can be useful and an interesting measure of the robustness of parameter estimates can be obtained from plots of each estimated parameter value versus the length of the experimental record.

Possible dependencies of parameter estimates on record lengths are also linked to the relationships between parameter estimates and the frequency range of the data. Knowledge of how estimates vary with the frequency range of measured signals used in the estimation can provide valuable insight concerning robustness. Ideally we want to maximise the range of frequencies over which parameter estimates are more or less constant and any indication of high sensitivity of estimates to frequencies over a part of the relevant range of frequencies may indicate problems of model structure or experiment design.

Checks of the overall robustness of an identified model must be made using data sets that were not used during the original identification process. These additional sets must be broadly similar in spectral properties and amplitude distribution to sets used for identification. One way of using such additional sets is simply to carry out additional identification runs and compare the different estimates obtained. Clearly the situation would be judged unsatisfactory if variations of parameter values obtained in this way were greater than could be expected from variance estimates for the first set of parameters obtained. Another approach involves the identified model being subjected to inputs not used for identification (See e.g. [37]). The predicted output from the model must then be compared with the corresponding measurements. This is a form of external validation and is discussed further in Chapter 7.

6.3.2 Test inputs and experimental design for helicopter system identification and parameter estimation

Test inputs commonly used for helicopter system identification include doublet signals, other forms of multi-step signals such '3-2-1-1' pseudo-stochastic signals (see [38] and [39]) and frequency-sweep signals. For the 3-2-1-1 the numbers used to describe the input refer to time units between input signal reversals. Figure 6.2 shows practical doublet, 3-2-1-1 and frequency-sweep test signals, together with some typical responses. It may be seen that the frequency-sweep signal provides a broader range of frequencies, although it should be noted that the record length for this input is greater than for the 3-2-1-1 or doublet. This could be important for cases where the stability margin of the vehicle is small since the use of such a long input sequence may limit the length of record permissible during the unforced part of the response.

Designs of test signals for system identification of a helicopter or other system are inevitably based on a model of that system. Because of uncertainties within that model, the resulting signals are unlikely to be optimal. Indeed, if uncertainties were not present, there would be no need to use system identification. This means that it is important to characterise some appropriate flight data from the vehicle in question using relatively simple forms of input as a first step towards experimental design.

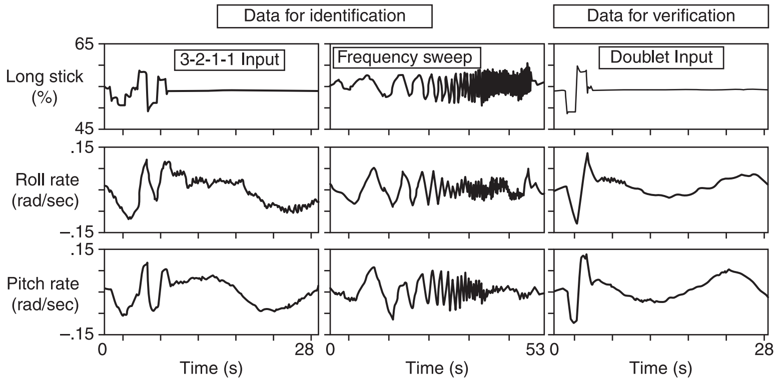


Figure 6.2 **Examples of BO105 flight test data showing three different types of control input (3-2-1-1, frequency sweep and doublet), together with roll rate and pitch rate responses. (The original version of this figure was published by the Advisory Group for Aerospace Research and Development, North Atlantic Treaty Organisation (AGARD/NATO) in AGARD Advisory Report 280 'Rotorcraft System Identification', in September 1991)**

We need a quantitative basis for comparison of test signals and, in the approach presented here, this involves quantities such as the parameter information matrix and the dispersion matrix, as outlined in Section 6.2. It should be noted, however, that care must be taken when applying such an approach since, unless an efficient estimator is used, the approach may be invalid. Inputs designed using measures based on the dispersion matrix have been found to be especially useful in cases where long test records are available and where maximum-likelihood estimators are being applied, since such estimators are asymptotically efficient.

The coherence function may be helpful as a measure of the degree to which a given signal provides satisfactory excitation [5]. It provides a measure of the fraction of the output auto-spectrum which may be accounted for by a linear relationship with the input auto-spectrum [39]. In the ideal case the coherence is unity over the complete frequency range of interest. Values of coherency smaller than one may be associated with nonlinearity in the system under test, process noise (such as turbulence in the case of aircraft applications) or lack of input signal power and thus response power [5]. Figure 6.3 shows some typical records and plots of coherency, revealing larger coherence values for the frequency-sweep data over a wider range of frequencies compared with the other test inputs [32]. This is especially noticeable at frequencies below 1 rad/s and above 10 rad/s where the autospectra for the doublet and 3-2-1-1 show significantly reduced power levels. In general, larger and more constant coherence values over a wide range of frequencies, coupled with relatively smooth input autospectra, are seen as desirable for identification purposes [40]. This suggests that the frequency sweep has advantages, provided the longer duration of this type of test signal can be tolerated.

In cases where the aim of identification is validation of linearised flight mechanics models, the inputs used for testing must be consistent with the modelling assumptions. This means that input design methods must take account of input constraints. In addition, it is important to obtain long test records, since parameter estimates then have time to converge and efficient estimation (i.e. minimum variance estimation) is possible, thus allowing use of criteria based on the dispersion matrix.

The broad aim of research by Leith and Murray-Smith [41] was to design a test input which would give long test records while providing a dispersion matrix that is reasonably ‘small’. It is important to avoid resonances in the system, since an input that excites resonances could rapidly produce a nonlinear response and this might require the flight

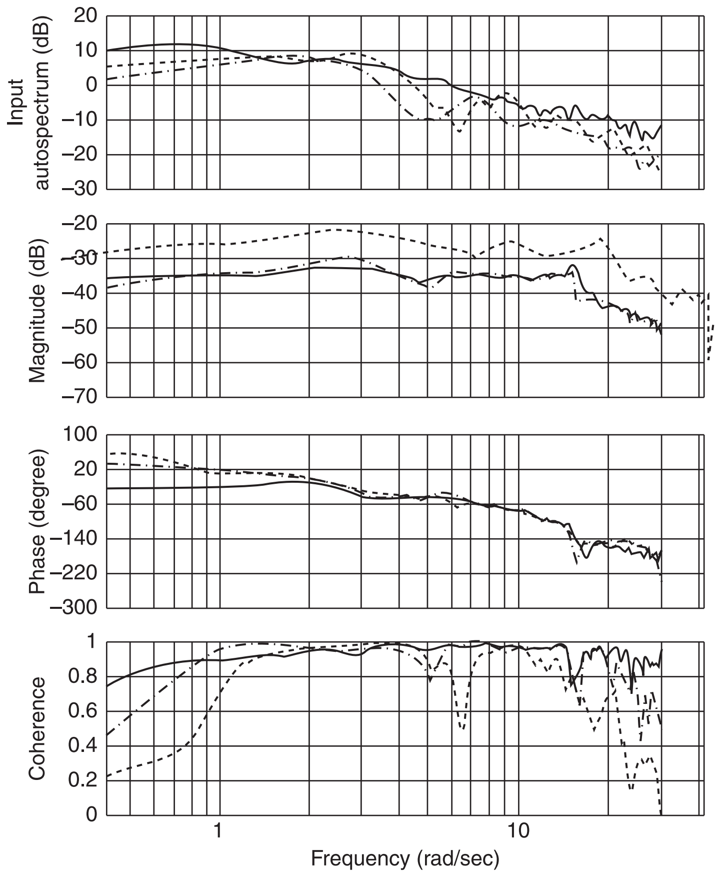


Figure 6.3 Autospectra and roll rate data in response to three types of test input signal (doublet, 3-2-1-1 and frequency sweep) applied through the lateral cyclic control for BO105 helicopter. The results reveal larger values of coherence over a wider frequency range for the frequency-sweep data (continuous line) compared with the doublet or 3-2-1-1 test inputs. (The original version of this figure was published by the Advisory Group for Aerospace Research and Development, North Atlantic Treaty Organisation (AGARD/NATO) in AGARD Advisory Report 280 'Rotorcraft System Identification', in September 1991)

experiment to be prematurely aborted. Inputs should also be chosen to ensure that the signal has no steady-state component. A constant component in the input will tend to produce a steady-state constant component in the response and this can shift the operating point. If the operating point is significantly different from that used for linearisation of the theoretical model, the parameter estimates obtained experimentally will be inconsistent with that model, thus making the whole procedure invalid.

A method of autospectrum design was developed ([41] and [42]) that:

- ensures that resonances are avoided, to give longer test records;
- avoids exciting frequencies around the resonances, to give robustness;
- excites the remaining frequencies, to give a reasonably 'small' dispersion matrix; and
- allows users to choose inputs that are relatively simple in form, so that they can be applied manually by the pilot.

An optimal spectrum program was then written to produce a binary multi-step input having an auto-spectrum that satisfies a specification of the type outlined above and this approach was applied successfully to the design of test inputs for a Lynx helicopter. Flight trials were performed for a test input applied to the longitudinal cyclic control of the vehicle for a flight condition of 80 knots level flight. The optimal test signal design process ensured that the input auto-spectrum had no DC component, that it avoided known resonances at about 0.3 rad/s and that the input excited frequencies between 2 and 3 rad/s but not above 3 rad/s. The upper limit of 3 rad/s was imposed because previous experience suggested that the theoretical model was useful only for frequencies below about 3 rad/s. At higher frequencies, dynamic effects within the rotor sub-system are believed to have a significant influence and these were not included in the model.

A signal consisting of five steps was found to be particularly useful. This signal, a double-doublet, allowed long test records before the response became nonlinear. Typical record lengths for the double doublet with the Lynx helicopter were of the order of 30 seconds compared with 10–15 seconds for a traditional doublet input and only 3 seconds for the 3-2-1-1 input. Estimates of seven parameters of the pitching moment equation were obtained using the frequency-domain equation-error approach [41]. Other forms of multi-step input were considered and tested in flight, including a 1-2-2-1 signal. Overall, the double-doublet gave results that were consistently better than those obtained from the use of other inputs and this also appears to be more robust to errors and uncertainties in the theoretical model used in its design [41].

Note that frequency-domain methods were chosen for this application, partly because of the physical insight that these provide in the subsequent application of the models for flight control system design and also because the frequency domain offers the possibility of separating the six-degrees-of-freedom vehicle dynamics from the rotor dynamics.

Further discussion of results from the application of system identification methods to helicopter flight mechanics model development are presented in the AGARD Advisory Report 280 [32] and the associated Lecture Series volume [33].

6.4 Some selected methods of local and global parameter optimisation

The techniques available for the optimisation of physically based dynamic models have much in common with methods of optimisation used in other application areas such as design. They include gradient methods and other local optimisation techniques along with more general search-based methods for determining global optima.

Methods that employ gradient information for local optimisation are widely used in system modelling (see e.g. [3] and [4]). The simplest gradient-based method is the *steepest-descent* approach, which does not require second-order derivatives of the cost function, but converges slowly. *Newton's method* involves the inverse of the Hessian matrix and depends on second-order derivatives resulting in additional computational overheads. Newton's method is also computationally demanding because it involves matrix inversion but use of the *quasi-Newton method* reduces the complexity by using an approximation to the inverse Hessian.

Conjugate-gradient methods, such as the *Fletcher-Reeves* algorithm, can be less computationally demanding than the Newton and quasi-Newton methods. Instead of using the Hessian matrix or an approximation to the Hessian, conjugate-gradient methods compute an estimate of the search direction more directly. Although they usually require more iterations than the quasi-Newton and Newton methods to converge, the overall speed tends to be better.

Nonlinear least-squares methods are preferred for cases in which the loss function is of the sum-of-squares type. Two well-used nonlinear least-squares methods are the *Gauss-Newton* method and the *Levenberg-Marquardt* approach. As discussed by Söderström and Stoica [2], the Gauss-Newton algorithm is closely associated with the general and modified forms of the *Newton-Raphson* algorithm for solution of numerical search problems. The Newton-Raphson algorithm provides the basis of two of the traditional iterative approaches to inverse simulation, as discussed in Chapter 4.

The simplest general-purpose nonlinear local optimisation techniques are termed 'direct search' methods and make use only of loss function values in their search for local optima. Such methods include the *simplex search*, *Hooke-and-Jeeves* and *Nelder-Mead* methods. These methods are typically rather slow and are often only used if the derivatives of the loss function are not available or can be estimated only at considerable computational cost. The Nelder-Mead approach is also mentioned in Chapter 4 in the context of one approach to inverse simulation.

6.4.1 Simulated annealing (SA)

Simulated annealing (SA) is a probabilistic hill-climbing technique based on the annealing of metals (see e.g. [11], [12] and [43]). This natural process occurs after the heat source is removed from molten metal and the temperature of the metal starts to fall as heat passes to the environment. At each temperature level, the energy of the metal molecules decreases and the metal becomes more solid. This continues until the temperature of the metal equals the temperature of the surroundings and, at this stage, the energy has reached its minimum. The simulated annealing process mimics this natural annealing process as it searches for an optimum.

In the SA algorithm, the solution space is searched by imposing perturbations on the estimates of the parameters that are being optimised. These perturbations depend on a 'temperature' index T and their magnitudes at any stage in the process are given by:

$$pert(T) = k \times T \times rand \tag{6.5}$$

where $pert(T)$ is the perturbation at temperature index T, k is a scaling constant and $rand$ is a uniformly distributed random number lying between 0 and 1. In this algorithm the temperature index T becomes smaller with each step, thus reducing the size of the parameter perturbation as conditions come close to the optimum. Each set of parameters arising from this procedure is substituted into the equations of the model and the performance is evaluated through simulation. This performance evaluation involves comparison of the desired and simulated responses, and is quantified using the relative cost (C). If the cost value is smaller than the previous best cost, the new parameter set replaces the previous set. If the new cost is not smaller, the new set of parameters is not immediately discarded and the cost value is subjected to a check in which the probability, P, of the cost associated with the new parameters (C_{new}) is compared with the previous best cost (C_{prev}) through the equation:

$$p = \exp\left(\frac{C_{prev} - C_{new}}{T}\right) \tag{6.6}$$

This has the same form as Boltzmann's Equation and the result obtained from its application is compared with a threshold number, n. If $P > n$, the new parameter values are accepted in the same way as if $C_{new} < C_{prev}$, but the new values are rejected if $P > n$. This so-called *Metropolis Criterion* [43] ensures that the SA avoids premature convergence to a local optimum. The criterion is illustrated diagrammatically by the flow diagram of Figure 6.4.

Following this step, the temperature index is reduced by the *annealing schedule* involving an equation:

$$AS(T) = T_d = \gamma^d T_0 \tag{6.7}$$

where T_0 is the initial temperature, γ is the reduction constant and d is the number of iterations. The whole process is repeated until either the cost has reached some preset threshold level or the temperature value has become so small

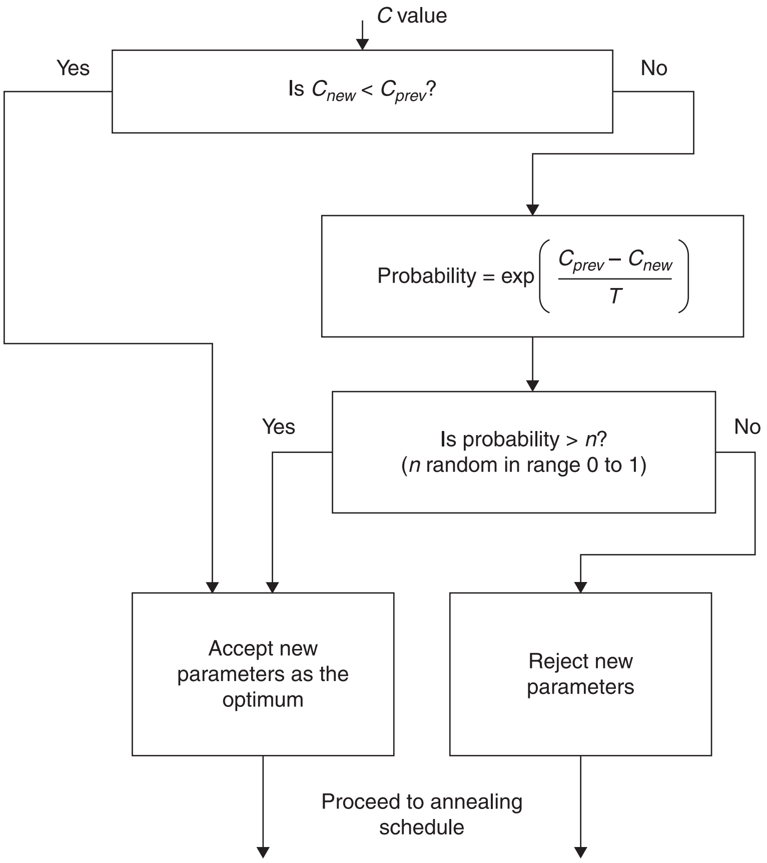

C value
Yes
Is $C_{new} < C_{prev}$?
No
Probability = $\exp\left(\frac{C_{prev} - C_{new}}{T}\right)$
Yes
Is probability > n?
(n random in range 0 to 1)
No
Accept new parameters as the optimum
Reject new parameters
Proceed to annealing schedule

Figure 6.4 **Flow diagram illustrating the operations involved in applying the Metropolis Criterion**

that the parameters are no longer being perturbed significantly. If the cost value has reached the minimum level, it follows that the SA should provide the optimum set of parameters but if the temperature is too small, the results may not be optimal.

The complete simulated annealing process outlined above may be summarised in the flow diagram of Figure 6.5.

A modified form of simulated annealing approach, known as *segmented simulated annealing* (SSA), involves a number

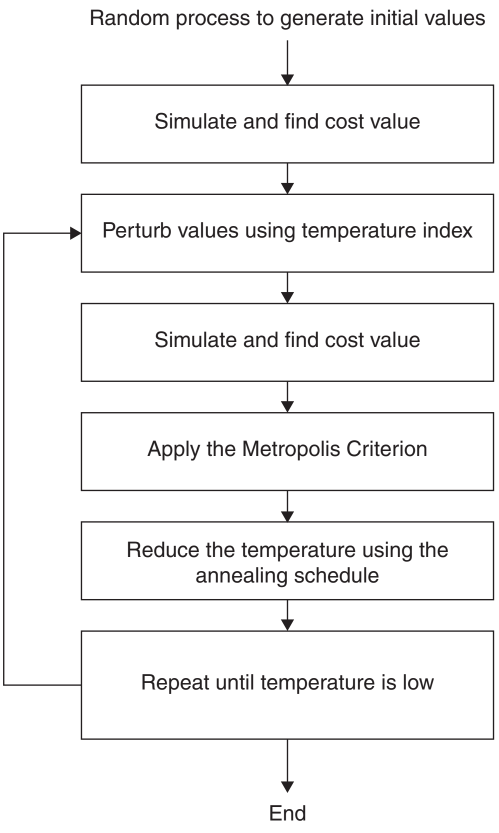

Random process to generate initial values
Simulate and find cost value
Perturb values using temperature index
Simulate and find cost value
Apply the Metropolis Criterion
Reduce the temperature using the annealing schedule
Repeat until temperature is low
End

Figure 6.5 **Flow diagram illustrating the operations involved in the simulated annealing (SA) process**

of SA processes applied consecutively (see e.g. [44] and [45]). The limited convergence properties of the SA method may be overcome through this approach since the search space is segmented into a number of smaller regions. The final cost values arising from the separate runs are sorted and the parameter values corresponding to the best cost value form the result. The SSA approach has been shown to provide practical benefits compared with the SA method for the optimisation of nonlinear controllers in a marine engineering context [45] and these advantages apply also to optimisation problems in system modelling.

6.4.2 Genetic algorithms (GA)

The *genetic algorithm* (GA) approach to optimisation is based upon the concept of survival of the fittest (see [13] and [14]). The GA emulates the processes of evolution and is therefore an *evolutionary algorithm*. In such a process the strongest elements become stronger while the weakest elements are eliminated.

The solution of an optimisation problem using the GA methodology involves a stochastic search of the solution space using strings of integers, known as *chromosomes*, which represent the parameters being optimised. Each integer within these chromosomes is known as a *gene* and, for these modelling applications, each gene has a decimal value between 0 and 9. It should be noted that this is not the traditional GA approach where genes are binary quantities. The advantage of the decimal representation for this type of application is that it allows a wider range of possible values in smaller chromosomes and is particularly suitable for both model and design optimisation.

An initial population of chromosomes is generated at random and these are decoded to obtain the corresponding parameters. These parameter values are then introduced into the system model. A simulation is run and results are obtained for each set of parameters within the population, using a measure of performance based on a cost function similar to that used in simulated annealing. When the cost values are all found, they are sorted into ascending order along with the corresponding chromosomes. As before, the smallest cost values are chosen as the best and are then subjected to operations involving *reproduction, crossover* and *mutation*.

The *reproduction* procedure involves retaining the best chromosomes (say 20 per cent) for the next population. The other chromosomes are replaced by new chromosomes formed through processes of crossover and mutation. This reproduction process is termed 'rank-based selection' and allows only the elite chromosomes to proceed to the next iteration. This method is therefore an *elite genetic algorithm* [46].

Crossover is a process in which two chromosomes from the current generation (*parent* chromosomes) engage in a procedure in which some genes from one chromosome are interchanged with genes from the corresponding positions in the other. This process produces two new chromosomes (*offspring*) and the procedure is repeated until there are sufficient offspring to replace the 80 per cent of the present population that have the worst cost values.

Mutation involves selection, on a random basis, of a certain number of the genes in the current population and random alterations are then made to their values. This provides a random element within the GA search process so that more of the search space is considered.

Once the chromosomes have been changed to form the new population they have to be evaluated, as were the previous generation. The whole procedure is then repeated for a predefined number of iterations (*generations*) to produce a final solution. This complete procedure is illustrated by the flow diagram of Figure 6.6.

It should be noted that, as well as being useful for optimisation of nonlinear physically based models, genetic algorithms and simulated annealing are also useful for system modelling directly from empirical data and for linearisation of models [47].

The work of Keskitalo and Leiviskä [22], discussed in the context of system identification and parameter estimation in Section 6.2.2, involved the use of genetic algorithms and

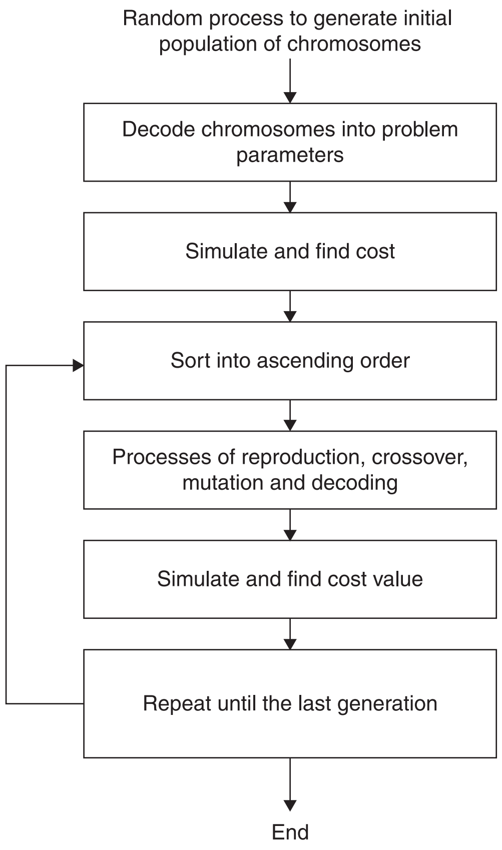

Random process to generate initial population of chromosomes
Decode chromosomes into problem parameters
Simulate and find cost
Sort into ascending order
Processes of reproduction, crossover, mutation and decoding
Simulate and find cost value
Repeat until the last generation
End

Figure 6.6 **Flow diagram illustrating the operations involved in the genetic algorithm (GA)**

another evolutionary technique, known as *differential evolution* (DE), for global optimisation of complex nonlinear models [48]. Like the GA approach, differential evolution is based on a population of evolving solutions but the DE method is real-coded and is more directly applicable to parameter optimisation problems. As with the GA approach, there are many variations on DE methods and those considered in the activated sludge model application of Keskitalo and Leiviskä are detailed in [22].

6.5 Genetic programming (GP) for model structure estimation

6.5.1 Principles of the genetic programming approach

Unlike the GA approach where, in a control engineering context, the objective is parameter optimisation, the methodology known as *genetic programming* (GP) involves no prior specification of the size, shape or structure of the solution, and algebraic expressions evolve from a database of nonlinear algebraic functional elements [49]. Like the GA it is an evolutionary optimisation method but, unlike the GA, it does not require a structure that is rigid in form. While the problems to which the GA has been applied involve a set number of tuning parameters and a fixed-length string representation for the solutions, the GP algorithm leads to a situation in which the size and shape of solutions evolve dynamically and can thus provide structure as well as an optimal set of parameters.

The GP approach allows optimisation of a tree structure representation of a symbolic expression. The tree structure has a variable length and is made up of a series of nodes. These can be *terminal nodes*, representing an input variable or a constant, but they may also be *non-terminal nodes* representing functions involving some form of operation on one or more variables of the system and could take the form of a block diagram element (such as a Simulink® function block). Figure 6.7 shows an example of a tree structure and, in this particular case, the terminal nodes are system inputs, variables of the system under investigation or constants. The non-terminal nodes represent the operations of forming a square root, addition and subtraction.

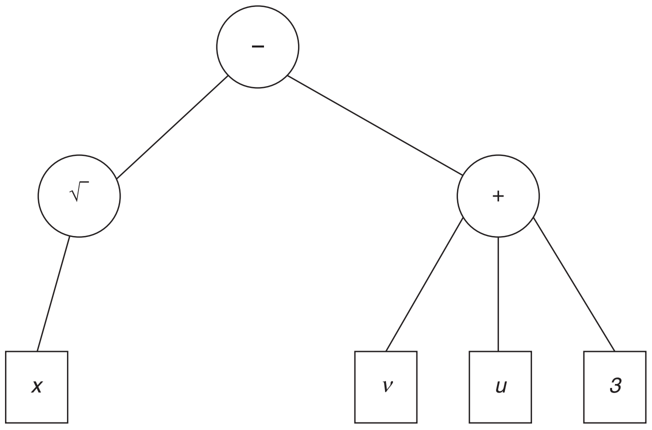


Figure 6.7 Structure of GP tree representing the function $y = \sqrt{x} - (v + u + 3)$. Here the circles represent non-terminal nodes whereas the rectangular blocks are terminal nodes

The GP algorithm chooses possible elements from a library to build trees of this kind and each tree is evaluated as an algebraic expression to provide a fitness function value. A population of trees is established and this population evolves through the processes of crossover, selection and mutation towards a structure that is optimum for the chosen fitness function. The process is not deterministic and repeated runs therefore produce different solutions. Some analysis of runs must be carried out before an expression that is potentially useful can be found.

6.5.2 Nonlinear model structure estimation using the GP algorithm

The GP approach can be used to introduce an element of automation within experimental modelling. A set of possible model structures evolves through many generations and, at each stage, equations generated through genetic programming to represent part of the model structure are combined with other well-established equations involved in the model description to produce a simulated time response which must be compared with experimental data to give a fitness value for that model. Figure 6.8 is a schematic diagram illustrating this procedure.

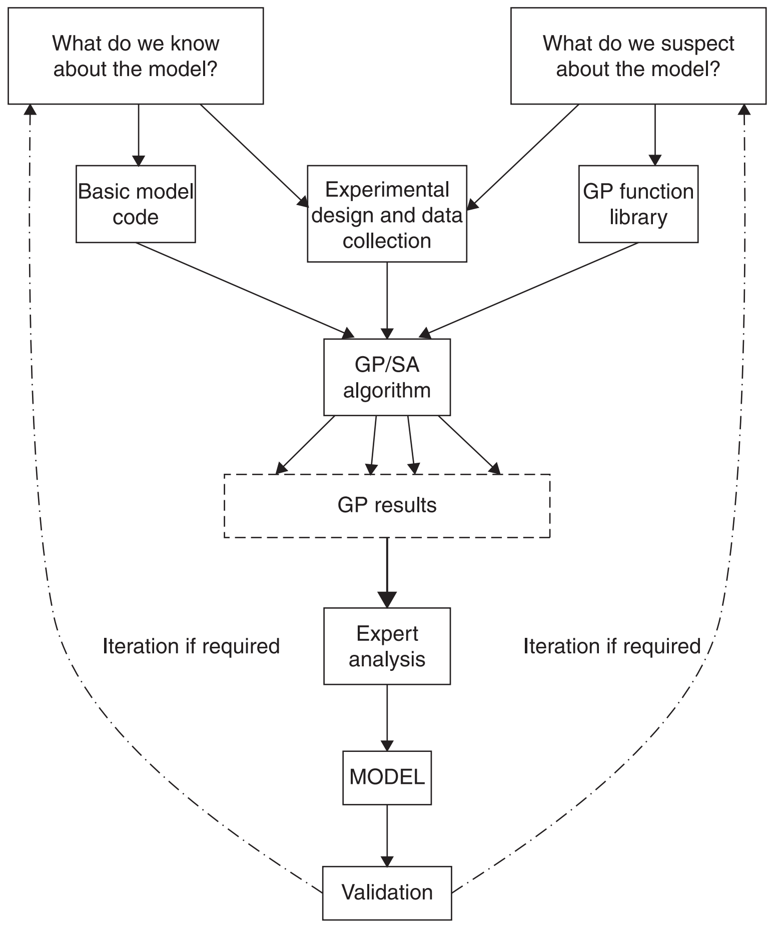


Figure 6.8 **Schematic diagram illustrating the GP modelling procedure (from [6.54])**

The parameters of the candidate models can be estimated using other numerical optimisation methods involving simulated annealing, or simulated annealing combined with Nelder-simplex optimisation [10]. It should be noted that gradient methods of optimisation cannot be used in the parameter estimation process because many models generated through the GP process contain linearly dependent parameters or parameters that have no effect on the model output. The fitness function value from the best parameter fit is then used by the GP algorithm to define the fitness of that specific function tree.

Experimental design is of particular importance in the case of nonlinear systems, since the system must be excited over the whole frequency range of interest and also, in terms of amplitude, over the range of all the nonlinearities within the system. A large training data set is therefore needed. However, large data sets imply additional experimental costs and also significant computational demands in terms of the chosen optimisation process, so there are inevitable trade-offs between model accuracy, optimisation time and cost.

Genetic programming methods have been applied successfully to the estimation of nonlinear model structures for continuous-time models for a number of application areas (e.g. [50], [51], [52], [53] and [54]). These range from simple simulated systems to chemical process system models, solid oxide fuel cell models and a model of a system for engine and rotor speed control in an MBB BO 105 helicopter. The candidate models may be described in a number of ways, including block-diagram or ordinary differential equation-based representations and prior knowledge of the physical system can be incorporated within those descriptions. Aspects of the model that involve unknowns evolve in the GP approach as expressions within the set of equations that make up the model.

The GP algorithm builds the models from a library of available functions. This library is very important and must be sufficiently flexible to allow for a wide range of functions, but not so general that a purely empirical representation can evolve which lacks any physical foundation. It should include basic algebraic operations (such as addition, subtraction, multiplication, squaring) together with functions that represent common forms of dynamic characteristics (such as first- or second-order linear sub-models) that might be expected to appear as elements within a model. Figure 6.9 shows an example involving a block diagram type of description. Note that any model structure found using the GP approach needs to be validated using a data set that is different from the data set used for the optimisation.

Results obtained from published applications suggest that genetic programming can be used to fit a model intelligently, in terms of the topology and block structure employed, while parameters can be estimated through the application of the

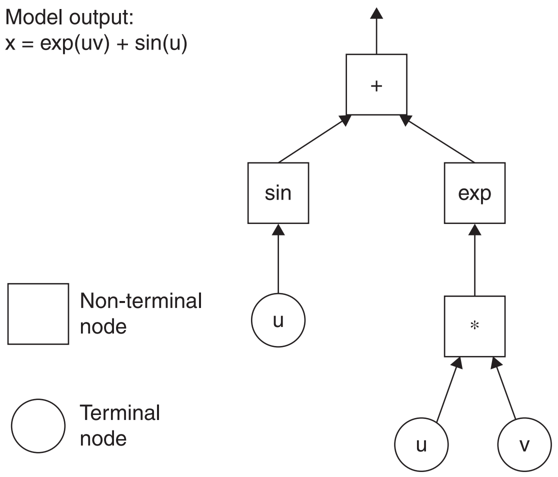


Figure 6.9 **Illustration of a GP tree for a typical block diagram function (from [6.54])**

GA or simulated annealing. With suitable constraints, this approach could provide additional insight regarding physically based model structures or could be used to validate a given nonlinear model using experimental data. This approach thus provides an automated and more systematic version of the trial-and-error processes generally used for model structure estimation. It allows poor features to be eliminated and good features to be combined to give new forms of sub-model, and it also allows more candidate model structures and components to be evaluated than would be possible in any manually directed procedure. A model structure that has evolved from the application of the GP approach can often reveal new information about the system under investigation or lead to additional experimental testing that may, in turn, throw new light on the physical processes involved.

Although the process of selection of the optimum description from among the candidate models is automated in this approach, the human skill in the choice of fitness function is vitally important for the ultimate success of the method. Physical understanding of the real system is also essential in the selection of the set of candidate models. In addition, it should be noted that the simulation methods used should be numerically efficient and fast, because each evaluation of the fitness function involves one simulation run and many evaluations may be needed, thus requiring a very large number of simulation runs in total.

6.6 Some practical issues in global parameter optimisation

Although evolutionary techniques such as the GA and GP approaches can be very much more efficient than any kind of exhaustive search algorithm, the computational costs may still be very significant. In practical terms this means that, for complex problems, decisions must be made about the extent to which extensive global optimisation can be justified and the accuracy required in solutions. If a near-optimal solution can be found quickly, the question must be asked about whether further searching of the solution space is necessary.

From the discussion in Sections 6.4 and 6.5, it is clear that evolutionary techniques, such as the GA, can produce different solutions in repeated optimisation runs for the same set of initial conditions and the same fitness functions. This is inevitable because of the processes of mutation that are an essential feature of these methods. In some situations it is appropriate to compare a number of near-optimal solutions of this kind and, if they are all found to be similar, an average may be used. In other cases, however, this multiplicity of solutions can be to the user's advantage and there may be benefits in choosing one solution from this set of acceptable solutions using additional factors, such as ease of implementation or robustness, along with the fitness function value. Additional factors of this kind are often difficult to include within an objective measure and, in some applications, a solution with slightly poorer performance may be preferred. One approach to simulation model optimisation which develops this idea is based on multimodal optimisation methods and aims to find several local optima in a search space through a single optimisation run using the *Crowding Clustering Genetic Algorithm* [55].

Although evolutionary algorithms offer a potentially important element of automation for optimisation procedures, both for model development and for design, their application requires good understanding of the likely physical phenomena in the system under investigation and therefore does not, in any way, imply a fully automated approach. For example, the success of the GP approach most widely used for model structure identification depends critically on the selection of appropriate functions for a function library. Examples of the information needed in establishing this function library and the initial form of model may include the following: first estimates of the order of the model, first estimates of the forms of nonlinearity most likely to be involved, known interactions between variables of the system and the form and limitations of existing models of similar systems. It is also important for the investigator to have an understanding of the availability of experimental data, the limitations of experimental design and the possibility that the resulting experimental data could be unevenly distributed over the operating range.

The role of the investigator is still vitally important and interaction between the user and the evolutionary optimisation tools is essential at various stages. Similar conclusions can be reached in the context of artificial neural networks and the closely associated methods involving local model networks and multiple models [56]. In those approaches, factors such as the choice of sub-models, the number of hidden layers and the number of neurons in a neural network, the choice of learning rates and other factors have to be chosen by the user, mostly on the basis of previous experience. Indeed, virtually all methods of system modelling involve issues of this kind where manual intervention by the user is essential.

In some cases, intervention involves the selection of adjustable parameters which are essentially 'fiddle factors', whereas in others the manual process involves more fundamental choices involving changes of model structure. Usually, however, the reason for undertaking these procedures manually is the fact that available algorithms for the more automated aspects of the system identification and model development process are not sophisticated enough to allow the optimisation to be completed automatically. It may not even be possible to express the objectives of the optimisation in a sufficiently simple fashion. In many applications, constraints have to be considered and there may also be a number of different objectives that have to be satisfied simultaneously.

6.7 Further examples of system identification, parameter estimation and model optimisation techniques in integrated systems applications

The applications to helicopter flight mechanics modelling discussed in the sections above are, in many ways, also typical of system identification, parameter estimation and model optimisation techniques for the development of models in other application areas. For example, the issues of identifiability, test input design and global optimisation described in the context of helicopter applications are equally important in other areas involving integrated systems such as wind turbines [57], bipedal robots [58] and surface and underwater vehicles [59].

The key issue in the helicopter system identification work and in these other types of application is uncertainty concerning the structure or parameters within a physically based model. For example, a recent tutorial paper by Pao and Johnson [57] provides useful insight regarding problems encountered in modelling wind turbines. The tower dynamics, substructure dynamics and foundation dynamics must all be taken into account and, in the case of offshore turbines, hydrodynamic effects also become important. External conditions in terms of turbulent air inflow, sea waves and sea currents may also have to be represented. If the wind turbine operates along with other turbines in a wind farm, there can be additional complications due to aerodynamic interactions. Because of the inherent difficulties in modelling wind turbine dynamics on an entirely physical basis from first principles, due to uncertainties in the details of many aspects of the physical model, there is growing interest in the use of system identification techniques for establishing linear time-varying models, especially for use in the design and development of improved turbine control systems (e.g. [60]). Further discussion of issues relating to the modelling of wind turbines and wind farms may be found in [61].

As a second illustration, a recent paper by Park *et al.* [58] provides a very interesting account of the use of system identification and parameter estimation techniques in the design of a bipedal robot. The physics-based model that provides the basis for the design involves many model parameters for which reliable estimates were not available *a priori*. Motor torque constants, rotor inertias, spring stiffness and preload values, cable stretch stiffness values and damping coefficients and various friction coefficients all had to be estimated from experiments carried out on sub-systems or on the complete robot. The improved model led to development of improved controllers and thus to significant enhancements of the robot's performance, especially in terms of the robustness of a walking gait [58].

The issues that arise in helicopter flight mechanics modelling and in the other engineering applications mentioned above are typical of the reasons why experimental modelling techniques based on system identification, parameter estimation and optimisation methods are increasingly being recognised as important for the development and enhancement of models for integrated systems applications. However, experimentally determined models are not necessarily fit for purpose and it is always important that models are subjected to appropriate checking procedures to ensure that they are appropriate and fully satisfy the requirements of the application. These issues of verification, validation and accreditation of models are considered in detail in Chapter 7.

6.8 References

[1] Ljung, L. (1999) *System Identification: Theory for the User*, Second Edition, Prentice Hall, Upper Saddle River NJ, USA.

[2] Söderström, T. and Stoica, P. (1989) *System Identification*, Prentice Hall, New York, USA.

[3] Nelles, O. (2001) *Nonlinear System Identification*, Springer, Berlin, Germany.

[4] Raol, J.R., Girija, G. and Singh, J. (2004) *Modelling and Parameter Estimation of Dynamic Systems*, IET Control Engineering Series No. 65, IET, London, UK.

[5] Tischler, M.B. (1991) 'Identification techniques – frequency domain methods', in *AGARD Lecture Series 178 (AGARD-LS-178), Rotorcraft System Identification*, Section 6, AGARD, Neuilly sur Seine, France, October.

[6] de Leeuw, J.H. (1991) 'Identification techniques – model structure and time domain methods', in *AGARD Lecture Series 178 (AGARD-LS-178), Rotorcraft System Identification*, Section 5, AGARD, Neuilly sur Seine, France, October.

[7] Knudsen, M., *Experimental modelling of dynamic systems*, Lecture Note, Department of Control

Engineering, Aalborg University, Aalborg East, Denmark (online): *www.control.aau.dk/~mk/ExpMod/Publicdoc/LectureNote02pdf.pdf* (accessed 7 November 2010).

[8] Knudsen, M. (2006) 'Experimental modelling of dynamic systems: an educational approach', *IEEE Transactions on Education*, Vol. 49, No. 1, pp. 29–38.

[9] Knudsen, M., Senstools (online): *www.control.auc.dk/~mk/public_html/ExpMod* (accessed 7 November 2010).

[10] Press, W.H., Teukolsky, S.A., Vetterling, W.J. and Flannery, B.P. (1992) *Numerical Recipes in C*, Cambridge University Press, Cambridge, UK.

[11] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983) 'Optimization by simulated annealing', *Science*, Vol. 220 (4598), pp. 671–80.

[12] van Laarhoven, P.J.M. and Aarts, E.H.L. (1987) *Simulated Annealing: Theory and Applications*, Lancaster, Dordrecht, the Netherlands.

[13] Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems,* MIT Press, Cambridge, USA.

[14] Goldberg, D. (1989) *Genetic Algorithms in Searching, Optimisation and Machine Learning*. Addison-Wesley, Reading, USA.

[15] The Mathworks Inc, *System Identification Toolbox* (online): *www.mathworks.com/products/sysid* (accessed 25 August 2010).

[16] The Mathworks Inc, *Frequency Domain System Identification Toolbox* (online): *www.mathworks.com/products/connections/poduct_detail/product_35570.html* (accessed 25 August 2010).

[17] Bellman R. and Åström, K.J. (1970) 'On structural identifiability', *Mathematical Biosciences*, Vol. 7, pp. 329–39.

[18] Beck, J.V. and Arnold, K.J. (1977) *Parameter Estimation in Science and Engineering*, Wiley, New York, USA.

[19] Silvey, S.D. (1975) *Statistical Inference*, Chapman and Hall, London, UK.

[20] Federov, V.V. (1972) *Theory of Optimal Experiments*, Academic Press, New York, USA.

[21] Hunter, W.G., Hill, W.J. and Henson, T.L. (1969) 'Designing experiments for precise estimation of some of the constants in a mechanistic model', *Canadian Journal of Chemical Engineering*, Vol. 47, pp. 76–80.

[22] Keskitalo, J and Leiviskä, K. (2010) 'Evolutionary optimisers in calibration of activated sludge models', in Šnorek, M., Buk, Z., Čepek, M. and Drchal, J. (eds), *Proceedings of the 7th EUROSIM Congress on Modelling and Simulation*, *September 6–10 2010*, *Prague*, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic.

[23] von Grünhagen, W. (1988) *Modellierung und Simulation von Hubschraubern*, DLR Institute Report IB 111 88/06 (in German).

[24] Kaletka, J. (1991) 'BO 105 identification results', *AGARD Lecture Series 178 (AGARD-LS-178), Rotorcraft System Identification*, Section 10, AGARD, Neuilly sur Seine, France, October.

[25] Hamel, P.G. (1994) 'Aerospace vehicle modelling requirements for high bandwidth flight control', in Cook, M.V. and Rycroft, M.J. (eds), *Aerospace Vehicle Dynamics and Control*, pp. 1–31, Clarendon Press, Oxford, UK.

[26] Anonymous (1994) *Aeronautical Design Standard ADS-33D. Handling qualities specifications for military*

rotorcraft, Directorate for Engineering, US Army Aviation and Troop Command, St. Louis MO, USA, July.

[27] Manness, M.A. and Murray-Smith, D.J. (1992) 'Aspects of multivariable flight control law design for helicopters using eigenstructure assignment', *Journal of the American Helicopter Society*, Vol. 37, No. 3, pp. 18–32.

[28] Tischler, M.B. (1991) 'System identification requirements for high-bandwidth rotorcraft flight control system design', in *AGARD Lecture Series 178 (AGARD-LS-178), Rotorcraft System Identification*, Section 14, Neuilly sur Seine, France, AGARD, October.

[29] Prempain, E. and Postlethwaite, I. (2005) 'Static H_∞ loop shaping control of a fly-by-wire helicopter', *Automatica*, Vol. 41, No. 9, pp. 1517–28.

[30] Padfield, G.D., Thorne, R., Murray-Smith, D.J., Black, C. and Caldwell, A.E. (1987) 'UK research into system identification for helicopter flight mechanics', *Vertica*, Vol. 11, No. 4, pp. 665–84.

[31] Black, C.G. and Murray-Smith, D.J. (1989) 'A frequency-domain system identification approach to helicopter flight mechanics model validation', *Vertica*, Vol. 13, No. 3, pp. 343–68.

[32] Anonymous (1991) *Rotorcraft System Identification*, AGARD Advisory Report 280 (AGARD-AR-280), AGARD, Neuilly sur Seine, France, September.

[33] Anonymous (1991) *Rotorcraft System Identification*, AGARD Lecture Series 178 (AGARD-LS-178), AGARD, Neuilly sur Seine, France, October.

[34] Tischler, M.B. and Remple, R.K. (2006) *Aircraft and Rotorcraft System Identification*, AIAA, USA.

[35] Murray-Smith, D.J. (principal author) (1991) 'Robustness issues', in *AGARD Advisory Report 280, Rotorcraft System Identification (AGARD-AR-280)*, pp. 213–22. AGARD, Neuilly sur Seine, France.

[36] Klein, V. (1989) 'Estimation of aircraft aerodynamic parameters from flight data', *Progress in Aerospace Sciences*, Vol. 26, pp. 1–77.

[37] Padfield, G.D. (1991) 'SA 330 Puma identification results', in *AGARD Lecture Series 178: Rotorcraft System Identification (AGARD-LS-178)*, Section 10, AGARD, Neuilly sur Seine, France.

[38] Plaetschke, E. and Shulz, G. (1979) 'Practical input signal design', in *AGARD Lecture Series 10*, Section 3, AGARD, Neuilly sur Seine, France.

[39] Kaletka, J. (1979) 'Rotorcraft identification experience', in *AGARD Lecture Series 104*, Section 7, AGARD, Neuilly sur Seine, France.

[40] Otnes, R.K. and Enochson, L. (1978) *Applied Time Series Analysis*, Wiley, New York, USA.

[41] Leith, D.J. and Murray-Smith, D.J. (1989) 'Experience with multi-step test inputs for helicopter parameter identification', *Vertica*, Vol. 13, No. 3, pp. 403–12.

[42] Leith, D.J. and Murray-Smith, D.J. (1993) 'The design of energy and amplitude constrained optimal inputs for system identification', *Systems Analysis-Modelling-Simulation*, Vol. 13, No. 3–4, pp. 209–38.

[43] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953) 'Equations of state calculation by fast computing machine', *Journal of Chemical Physics*, Vol. 21, pp. 1087–92.

[44] Atkinson, A.C. (1992) 'A segmented algorithm for simulated annealing', *Statistics & Computing*, Vol. 2, pp. 221–30.

[45] McGookin, E.W., Murray-Smith, D.J. and Li, Y. (1996) 'Segmented simulated annealing applied to sliding mode controller design', in *Proceedings of the 13th World Congress of IFAC, San Francisco, USA, July 9–12, 1996*. Volume D, pp. 333–8, IFAC.

[46] Brooks, R.R., Iyengar, S.S. and Chen, J. (1996) 'Automatic correlation and calibration of noisy sensor readings using elite genetic algorithms', *Artificial Intelligence*, Vol. 84, pp. 339–54.

[47] Tan, K.C., Li, Y., Murray-Smith D.J. and Sharman, K.C. (1995) 'System identification and linearisation using genetic algorithms with simulated annealing', in *Proceedings First IEE/IEEE International Conference on GA in Engineering Systems: Innovations and Applications*, *12–14 September 1995*, *Sheffield, UK*, IEE Conference Publication Vol. 414., pp. 164–9, IEE London, UK, 1995.

[48] Price, K.V., Storn, R.M. and Lampinen, J.A. (2005) *Differential Evolution – A Practical Approach to Global Optimization*, Springer, Berlin, Germany.

[49] Koza, J.R. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA.

[50] Affenzeller, M., Wagner, S., Winkler, S. and Beham, A. (2009) *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*, CRC Press, Singapore.

[51] Gray, G.J., Li, Y., Murray-Smith, D.J. and Sharman, K.C. (1996) 'Structural system identification using genetic programming and a block diagram oriented simulation tool', *Electronics Letters*, Vol. 32, No. 15, pp. 1422–4.

[52] McKay, B., Willis, M. and Barton, G. (1997) 'Steady state modelling of chemical process systems using genetic programming', *Computers and Chemical Engineering*, Vol. 21, No. 9, pp. 981–96.

[53] Chakraborty, U.K. (2009) 'Static and dynamic modelling of a solid oxide fuel cell using genetic programming', *Energy*, Vol. 34, No. 6, pp. 740–51.

[54] Gray, G.J., Murray-Smith, D.J., Li, Y., Sharman, K.C. and Weinbrenner, T. (1998) 'Nonlinear model structure identification using genetic programming', *Control Engineering Practice*, Vol. 6, pp. 1341–52.

[55] Baccouche, A., Huyet, A.-L., Pierreval, H. and Feyech, B. (2010) 'Multimodal optimization of simulated systems', *Proceedings EUROSIM Congress*, Prague, September.

[56] Murray-Smith, R. and Johansen, T. (1997) *Multiple Model Approaches to Modelling and Control*, Taylor and Francis, London, UK.

[57] Pao, L.Y. and Johnson, K.E. (2011) 'Control of wind turbines: approaches, challenges and recent developments', *IEEE Control Systems Magazine*, Vol. 31, No. 2, pp. 44–62.

[58] Park, H.-W., Sreenath, K., Hurst, J.W. and Grizzle, J.W. (2011) 'Identification of a bipedal robot with a compliant drivetrain: parameter estimation for control design', *IEEE Control Systems Magazine*, Vol. 31, No. 2, pp. 63–88.

[59] Fossen, T.I. (1995) 'Identification of ship dynamics', in Fossen, T.I. (author), *Guidance and Control of Ocean Vehicles*, Section 6.8, pp. 321–52, Wiley, Chichester, UK.

[60] van Wingerden, W., Houtzager, I., Felici, F. and Verhaegen, M. (2009) 'Closed-loop identification of the

time-varying dynamics of variable-speed wind turbines', *International Journal of Robust Nonlinear Control*, Vol. 19, No. 1, pp. 4–21.

[61] Moriarty, P. and Butterfield, C.P. (2009) 'Wind turbine modelling overview for control engineers', in *Proceedings of the American Control Conference*, St. Louis, MO, pp. 2090–5.

Issues of model quality and the validation of dynamic models

Abstract: The aim of this chapter is to review some important issues concerning model quality and sources of errors in models, and to present some methods for testing, internal verification and external validation of simulation models. Questions of quality and validation are then considered in more detail through applications. Case studies illustrate external validation and model quality issues in helicopter and ship models. Issues relating to reduced models are considered using applications concerning aircraft and a hydro-turbine generator model.

Key words: quality, credibility, accuracy, errors, testing, internal verification, external validation, model reduction, helicopter, ship, aircraft, hydro-turbine.

7.1 An introduction to the issues of model quality and validation

A model is an abstraction of a real system and perfect accuracy should never be expected. The key question is the quality needed for the application and the adequacy of the chosen model in each case. Errors must be kept within specified limits for parts of the operating envelope and testing, verification and validation can be viewed as processes that define boundaries so that the necessary accuracy is achieved.

As already discussed in Chapter 2, many modern developments in engineering involve 'systems of systems' and require several teams working together. Such collaborative development means that there is no single 'designer' and soundly based models that are well understood and well documented are essential if all involved in the design process are to work together in an effective fashion.

Model building is an iterative procedure and many, including Sargent (e.g. [1]), Ören [2], Balci (e.g. [3], [4] and [5]) and Brade ([6], [7]) have all pointed out that model validation is inseparable from the processes of model building. Confidence in a model should increase steadily if appropriate methods are used and if correct methods of testing are applied.

Simple models are often used initially to investigate 'what if' situations and to assist in design trade-off studies. Error bounds on model predictions at this stage are usually large and little formal validation is possible. Assessments of model quality and fitness-for-purpose depend on experience and comparisons made with earlier models of similar systems. As the work progresses, more refined models are used and more data become available for testing. This usually involves data from component testing at first, followed by data from tests on larger blocks and, finally, data from tests on complete prototype systems.

Hence, with more test data becoming available as the design progresses, quantitative information begins to flow from the real system to the model. This contrasts with the initial situation when the flow is entirely from the model to the system being designed. Bidirectional information transfer characterises the later stages of any model-based design process, with model updates being applied as real system information becomes available.

All models have limitations and one important objective of validation is to properly define and understand those limitations. However, any practical validation investigation can cover only a finite, and often relatively small, number of test cases. Thus, one should never attempt to prove that a model is correct under all sets of conditions. Instead, a degree of confidence should be established in the model so that its results can be recognised as being reasonable for the objective for which it has been developed. General statements about the validity or quality of a model are therefore inappropriate without reference to its application and the range of conditions considered. One of the inherent problems is the fact that quantitative measures of model credibility are hard to define and as models become more complex, there are increasing problems of visualisation.

7.2 Model quality concepts, model uncertainties and modelling errors

There are good examples, often in *safety-critical* application areas, such as the nuclear, aerospace, defence, marine and off-shore sectors, where rigorous model testing and formal approval schemes are routinely applied. However, in other fields of application, model development within many engineering organisations often involves surprisingly little systematic assessment of the quality of models in terms of their useful range and limits of accuracy. Also, there may be cases where a model has a spurious justification, possibly on the grounds that it 'has always been used' or is 'based on well-known physical principles so must be right' or is 'based on an industry standard'.

The use of models that are inadequate for an application can often lead to expensive redesign at late stages in the development cycle. The more complex the system being developed, the more likely it is that problems of this kind will arise.

Reasons for *uncertainties* and *errors* in models include incorrect assumptions, errors in *a priori* information (such as parameter values), errors in numerical solutions and errors in experimental procedures and measurements. Although much attention has been given to separating different aspects of the model development, testing and checking process, and to categorising simulation model errors, uncertainty is inevitable since we never have a complete understanding of the real system and our measurements and calculations are limited in accuracy.

An untested model produces results with unknown and potentially unbounded errors. Even if the user has confidence that a model gives satisfactory results much of the time, the cases for which it produces inaccurate output cannot readily be predicted or immediately recognised unless great care is taken in using the model only within the bounds for which it has been tested successfully and found acceptable for the intended application.

Confidence in predictions depends on confidence in sub-system models as well as in the complete system model and this is particularly important when sub-system models can be tested experimentally. Comprehensive and detailed testing at the sub-model level, together with detailed documentation, helps to establish overall confidence. This allows a complex model to be extended from less well-understood situations, in a gradual way, until it can be tested successfully over the whole range of conditions likely to be encountered in service.

7.3 Model testing, verification and validation

Care must be taken about the words chosen to describe the assessment of model quality. For example, the process of deriving a computer simulation from a mathematical model can give rise to errors, but these are not the same as the types of error that arise in developing the model itself. The word 'verification' is commonly used to describe the process of establishing that a simulation is consistent with the underlying mathematical model, while the word 'validation' describes the process of demonstrating that the mathematical model representing the real system is appropriate for the application. Addition of the word 'internal' so that 'internal verification' is used to describe the process of establishing whether or not the model is simulated correctly can help to clarify the meaning. Similarly, the words 'external validation' can help in describing the processes of establishing the correctness of the structure, logic and parameters of the model itself [8]. These conventions are consistent with recommendations established in 1979 by the SCS Technical Committee on Model Credibility [9]. Although the SCS recommendations are now widely used, the words 'verification' and 'validation' are often applied very loosely and without the necessary precision. In addition, there are also specialist areas (for example, in missile system modelling) where common usage by some engineers has, unfortunately, interchanged the meanings for these two words.

Sargent [10] used a narrower definition of validation to emphasise the accuracy needed for useful model-based predictions for a specific application, and external validation may be viewed as a process leading to *confirmation* that the model output has a level of accuracy consistent with the intended use. To carry out this process of confirmation, it is essential that the accuracy requirements of the model should be established before any external validation is undertaken and not as part of that process. Results of external validation are thus also best expressed in terms of the suitability of the model for a planned application instead of as a 'good' or 'bad' description. Indeed, strictly speaking, one can never prove that a model is valid; a model can only be proved to be invalid. Available evidence can be assembled to suggest that a model is suitable for an application, but more general assertions of 'validity' must be avoided.

Whatever approach to external validation is adopted in a particular application, there are issues concerned with *identifiability* and *robustness* that must be considered. Identifiability has been discussed in Chapter 6 in connection with system identification and parameter estimation methods, and especially in connection with experimental design. Robustness in the context of model quality relates to the magnitude of error bounds on model parameter estimates, the accuracy and repeatability of model predictions, the effect of test input magnitudes and how the length of experimental records affects the accuracy of system identification.

It is thus necessary to distinguish carefully between the processes of system identification and parameter estimation that are applied in the initial stages of model development, the tuning procedures used in subsequent model optimisation and the processes applied in establishing the quality of the resulting model. The term 'model calibration' may be used to describe the repeated processes of optimisation and interactive tuning applied to a model during its development. Model calibration is not the same as model validation, as these processes take place at different points within the iterative cycle of model development.

7.3.1 Methods of internal verification and external validation

As discussed above, the words ‘internal verification’ describe the process of establishing that a computer simulation is consistent with the underlying mathematical model, whereas ‘external validation’ is a more open-ended task that involves comparisons between the model behaviour and the behaviour of the real system for chosen conditions. This can involve quantitative comparisons of the model’s performance with the real system or a more subjective assessment made by someone who has a deep and thorough practical understanding of the real system.

Internal verification

The procedures for the internal verification of a simulation model resemble processes applied more generally in the testing of software and some well-established principles of software engineering can be used.

Factors that are important in internal verification are:

- demonstration of internal consistency of the simulation program and the model upon which it is based, showing that there are no contradictions in terms of mathematics, logic or internal organisation; and
- demonstration of the simulation software in terms of the numerical algorithms being used and the associated numerical accuracy.

Internal verification procedures are needed at every stage of the development of a simulation. Every change within a model must lead to further internal verification of the associated simulation program.

Procedures of internal verification at the most basic level must include line-by-line checks of the simulation program or of connections between block diagram elements if the simulation is developed using a graphical user-interface. Connections must also be checked carefully if existing accepted (and thus internally verified and externally validated) sub-models are used. Checks should also be performed for special cases involving particular static or equilibrium conditions that can often be investigated from the underlying model using pencil-and-paper calculations. Simple checks may also be made, usually for dynamic conditions, of the appropriateness of the user's options in terms of the selected integration method and the integration time step. For example, use of an inappropriate integration method or integration step interval may result in numerical instability. This could be interpreted, incorrectly, as a feature of the model rather than an artefact of the simulation program. Similarly, the communication interval used for plots of output variables and for control of data flow between the simulation and external hardware or the operator is very important. An incorrect communication interval could lead to some transients being lost in the link between the simulation and the world outside.

Simple internal verification tests involving changes of integration method, integration step size or communication interval can often help to establish the true nature of any problem. For example, if small changes of integration step cause large changes in the overall behaviour of the model, it is likely that the underlying problem is numerical and is a feature of the simulation program rather than an error in the model.

External validation

External validation of simulation models is complicated by the fact that most models intended for practical engineering applications involve dozens or even hundreds of quantities provided by the user (e.g. as model parameters), leading to a large problem space. Similarly, many models can produce, as outputs, dozens or even hundreds of variables, each of which may contain different levels of error which may vary with time. Thus, it is important to establish, *a priori*, which output variables of a model are of greatest interest for the given application. Different users will be interested in different performance measures in different modelling studies and this emphasises that the model must be matched to the application at the outset and the errors that can be tolerated must be established *a priori*.

For external validation, a distinction should be made between 'functional' validation and 'physical' validation. *Functional validation* involves establishing the correctness, or otherwise, of a model that mimics the input-output behaviour of the real system. *Physical validation*, on the other hand, involves establishing the acceptability of the underlying assumptions and approximations [11] in addition to investigating the agreement between model and system variables. It has been pointed out by Hemez [12] that perfect matching of all available measured response data is unrealistic and that models should match available test data only to a level of accuracy appropriate for the application. This ensures that model responses match test data to an acceptable degree, while also showing satisfactory robustness to uncertainties associated with factors, such as modelling assumptions, environmental and model parameter variability or ignorance in terms of initial conditions in the real system. In model development, as in control system design, there must be a trade-off between performance and robustness.

External validation, whether of the functional or physical kind, involves two distinct stages. The first of these is concerned with establishing the range of conditions over which a model can be used for a specific accuracy level. This accuracy level can be defined generally in terms of frequency and amplitude. The second stage is concerned with establishing deficiencies in the model and the upgrades that would be necessary in order to achieve a level of performance appropriate for the intended application.

As has already been mentioned, external validation is a continuing exercise within the modelling process and is not a procedure that is performed only once at the end of the development cycle. It is also important to distinguish between holistic approaches that attempt to validate a complete model externally and model-component approaches in which much external validation is carried out at the sub-model level. Both depend on the same general principles of external validation, but the model component approach may also involve comparisons with test data (possibly from component manufacturers).

The procedures within the external validation process used to compare observed and simulated behaviour can be divided into subjective and objective categories. The subjective approach is based mainly on graphical analysis or experience using real-time simulations, while the objective approach involves quantifying the process through specific measures and statistical procedures.

Graphical methods are characterised by plots of simulated values (often continuous and represented by a line) and observed or measured values (usually discrete and represented by points) against an independent variable (often time). One issue sometimes missed by inexperienced observers is that the deviation between simulated and measured values is the vertical separation between corresponding points on the graphs and not simply the apparent distance between simulated and measured time history curves.

Another commonly used form of graph involves simulated values plotted against the corresponding measured or observed values. Ideally this plot should be a straight line at an angle of 45 degrees to the axes. Deviations from the ideal are shown by the vertical distance between the points and the 45-degree line. Points above the 45-degree line are clearly overestimated in the simulation while any points below the line are underestimated.

Although subjective, graphical methods are very useful in model validation and complement quantitative measures. Different graphical methods for displaying information about a model may provide different types of insight [13].

Quantitative measures for system and model comparison are clearly very important [14]. The most used deviance measures are the mean-square or mean absolute errors. For the case of n sets of measured and simulated values, the mean absolute error is expressed as the difference between observed values y_i and simulated values $\hat{y}_i$, by:

$$J_1 = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{7.1}$$

or using the closely related mean absolute percentage error, given by:

$$J_2 = \frac{100}{n}\sum_{i=1}^{n}\frac{\left|y_i - \hat{y}_i\right|}{\left|y_i\right|} \tag{7.2}$$

This is a relative error and is inapplicable if any of the observed values happens to equal zero. An obvious disadvantage of these two measures is their sensitivity to single extreme values.

Such an approach can be extended to include some form of weighting function. This means that errors arising in specific sections of the time history can be given special emphasis. One such cost function is:

$$J_3 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^T w_i (y_i - \hat{y}_i) \qquad (7.3)$$

where w is a weighting factor and the superscript^T indicates the transpose.

A measure that has received particular attention for external validation applications in a number of different application areas is Theil's Inequality Coefficient (TIC), which is defined as:

$$J_4 = \frac{\sqrt{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^{n} {y_i}^2} + \sqrt{\sum_{i=1}^{n} {\hat{y}_i}^2}} \qquad (7.4)$$

This measure has an advantage in providing values that lie between zero and unity, with values of TIC close to one indicating sets of model and system data that are very different. Values of TIC close to zero indicate small differences between the model and the system time histories.

Other scaled measures are also commonly used for comparing model and measured system time histories. One example is the normed root mean square output error adopted by Knudsen [15] in his approach to identification and system parameter estimation based on sensitivity functions. This measure is similar in most respects to the TIC measure, but with normalisation involving the sum of the squared values of experimental response samples only. Measures based on statistical techniques have also received attention, particularly in the context of model structure assessment.

One approach, which can be used with benefit in cases where relatively complex models are being considered, involves taking a number of key measured system or sub-system quantities and plotting these as radial lines on an appropriately scaled polar diagram of the type shown in Figure 7.1. By creating a polygon of model results and a polygon of corresponding measurements on the same polar diagram, an indication of the validity of the complete model is obtained. Closely similar polygonal shapes indicate that the model is suitable for the application. Aspects of the system that are represented accurately are immediately apparent and areas requiring further analysis are highlighted.

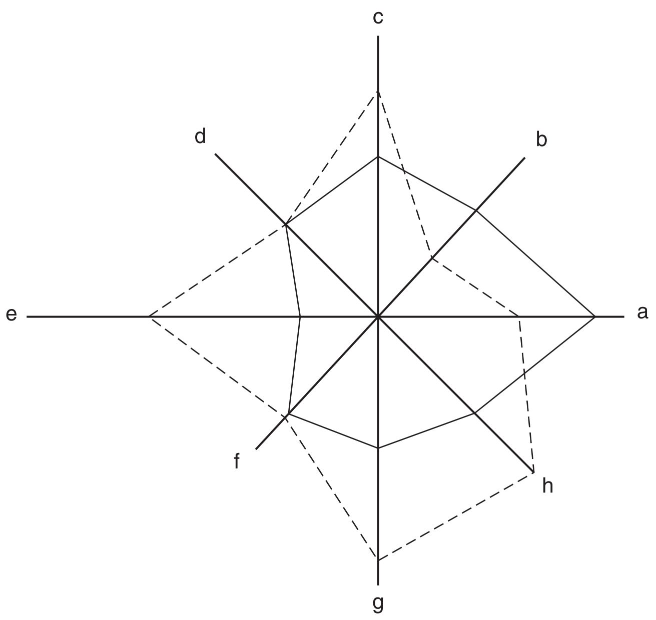


Figure 7.1 Example of polygon representation for model validation results. Here, solid lines represent model results for eight different quantities while the dashed lines indicate the corresponding measured values. This could, for example, involve parameter values within some physically-based model being compared with parameter estimates from system identification tests on the corresponding real system

Such diagrams also provide a basis for sensitivity analysis. The distortion of the model polygon following a specific change in the model provides a clear indication of sensitivities and interactions, as discussed in papers by Smith *et al.* ([16] and [17]). Shape-processed visualisation methods such as these lend themselves to image processing techniques for quantification and a number of approaches have been considered (see [18] and [19]). Using this kind of graphical comparison, it should be clear which aspects of a system are represented most accurately and which areas of the corresponding model require further investigation.

Polar diagrams such as these have been used successfully in the context of model testing for electro-optic sensor models ([16] and [17]) and have been considered in the context of fault detection applications as well as in other model-testing situations ([18] and [19]). Although they have been developed independently for model validation purposes, these diagrams have many features that are similar to those of the *Kiviat diagrams* ([17] and [19]), which are applied in the software engineering field for visualisation of metrics in computer software and hardware performance evaluation.

Diagrams of this kind are clearly applicable to problems in many areas where there is a need to display relationships involving several channels of results. One advantage of the polygon diagram approach to visualisation is that it is extremely flexible in terms of the comparisons that can be made. It is also appropriate for use with deterministic measures of performance such as the size of a system response overshoot or the frequencies of observed oscillations.

In the last ten years, there has been a growing interest in a move away from accuracy-centred assessment of models towards forms of assessment that are based more broadly on model quality (see e.g. [20]). Other discussions about the quantification of model credibility may be found in many sections of the book edited by Cloud and Rainey [21], in the textbook on the theory of modelling and simulation by Zeigler *et al.* [22] and in the work of Brade and Köster [23] and Brade *et al.* [24]. Specific suggestions have included the idea that the processes of testing and eventual accreditation of models should be based more closely on software quality assurance from the field of software engineering (see e.g. [23], [25] and [26]). Certainly, the model development procedures that are used in many organisations need to incorporate improved procedures, in terms of version control and documentation, that are now almost universally applied in software engineering and lead to proven benefits in terms of project management (see e.g. [18]).

7.3.2 Upgrading of models

Having made appropriate comparisons between the model behaviour and the corresponding behaviour of the real system, it is necessary to perform some analysis of the discrepancies and to propose upgrades for the model. Any changes in the structure or parameters of the model must be implemented through simulation and their significance evaluated in a systematic fashion, leading to further iterations within the model development cycle. In general terms, parametric changes are usually investigated before structural changes are contemplated. In lumped representations, a model parameter commonly provides an approximation to some more complex effect and there must be a limit to the range of conditions over which this approximation is valid. Tuning of model parameters to improve the functional validity of a model is an appropriate approach, provided constraints are introduced to ensure that the adjustment is within the range of uncertainties for each parameter. Parameter adjustment using global optimisation methods without regard to known uncertainties and physical limits can be very misleading. Often, when parameter values are found that appear physically meaningless, we are dealing with a situation where the model has an inappropriate structure. In this context, establishing the range of frequencies over which discrepancies occur can be very useful and a variety of frequency-domain techniques, such as analysis of coherence and partial coherence, can be applied. Deficiencies in model structure are generally more difficult to investigate and rectify, but once dealt with successfully, the upgraded model should have a broader range of applicability.

In some cases it may be possible to associate deficiencies in the model with particular state variables of the model or with specific physical phenomena. This may be attempted through statistical correlation of comparison errors with measured variables and inputs. Correlation of an error formed from the difference between a model output and the corresponding system output with a specific state variable may suggest that a more complex representation of the sub-system associated with that state variable would be appropriate. Padfield and Du Val [27] discuss the use of this type of approach in the context of helicopter flight mechanics model validation and point out that, for example, correlation of an output error with helicopter rotor speed could suggest that a more sophisticated representation of the coupled sub-system involving the engine, drive train and rotor is needed.

Correlation of model errors with derivatives of state variables of the model may also indicate that a higher-order description would be appropriate. When regression techniques cannot be used to associate such errors with specific state variables, their derivatives or some linear combination, it may be appropriate to consider possible nonlinear combinations of model states, but this should be approached (as far as possible) on a physical basis.

7.3.3 Experimental data for external validation

The choice of data sets to be used for testing models that involve parameters or structures identified using other experimental data raises some interesting issues. Data sets used for model testing need to be broadly similar to the sets used for identification, in terms of their spectral properties and amplitude distributions. It is also clear that data sets used for testing must not be too similar to those used for identification and parameter estimation during the development of the model. Responses obtained from inputs different from those used at the identification and parameter estimation stage are bound to be different in terms of amplitude, frequency and energy distribution. However, provided the spectral and amplitude distributions of the data from identification experiments and the data used for testing are not too different, the results of tests carried out with different inputs may still be very helpful in assessing the quality and limitations of the model.

One important point is that test data used for external validation must be matched to the intended application of the model. Otherwise it will not be possible to make decisions about the suitability and quality of the model for that application, and use of the model may be restricted.

In the case of linear models, the issue becomes one of obtaining experimental test records that are significantly different in form from the records used in the parameter estimation process but that are similar in terms of their amplitude and frequency ranges. Issues concerning the choice of experimental records for validation of identified models have been discussed in a number of papers and reports, some relating specifically to helicopter system identification (e.g. [27]).

For the validation of nonlinear models, the task of choosing appropriate test records is complex since the system must be excited so that all the significant nonlinearities are fully explored, while also covering the entire frequency range of interest. Ideally, what we need is some way of producing confidence intervals for model predictions. Although this goal may be elusive in the case of general nonlinear physics-based parametric simulation models, it is interesting to note that in the Gaussian Process type of nonlinear non-parametric model, such additional information is readily available (see e.g. [28]). Also, for linear models, the use of coherence estimates within frequency-domain descriptions of system outputs allows determination of the range of frequencies over which the linear model is applicable (see e.g. [29] and [30]). More research is needed concerning the application of such techniques to practical engineering problems and the development of better ways for assessing the accuracy of predictions from nonlinear physics-based models.

7.3.4 Additional issues in the external validation of nonlinear models

The external validation of nonlinear simulation models, in the general case, involves a number of important issues that depend on the nature of the nonlinearities and the intended application. For example, techniques for the identification of linear models from measured experimental data can provide insight through establishing models for different operating points across the operating envelope of the system. The trends in terms of the values of key parameters of the identified models can then be compared with trends in the values of corresponding parameters of linearised models derived from the nonlinear simulation model for the same operating conditions. Differences between the values of parameters of the identified models and the parameters of linear models derived from the nonlinear description can provide useful insight. Similarly, comparisons of trends in these parameter sets as operating conditions are changed are important indicators of the performance of the nonlinear model and may lead to its credibility being questioned. An example of this in the context of helicopter flight mechanics model validation may be found in Section 7.4.1.

If the level of agreement between the identified and theoretical models is considered adequate, a second stage of the external validation process can be attempted. This involves comparison of responses of the nonlinear model with the responses of the real system for larger perturbations and is based on the direct comparison type of approach discussed in Section 7.3.1, including the polygon type of graphical display. If, once again, the level of agreement is judged to be acceptable over an appropriate range of conditions, the model can be considered for release for the intended application. It can continue to be used until additional information or data give cause for concern. In some cases, external validation of nonlinear models may be attempted directly using simple graphical comparisons and methods involving the quantitative measures of Equations (1) to (4). However, preliminary investigations based on system identification and parameter estimation techniques can provide useful insight that may otherwise be missing.

When large inputs are applied to models having significant nonlinearities (e.g. helicopter flight mechanics models when large and aggressive control inputs are applied), traditional methods of validation based on direct comparisons of models and system have been found to have practical limitations, whether based on graphical methods or quantitative measures. Methods involving the opinions of experts involved with the real system (e.g. pilots in the case of aircraft or operators in the case of industrial systems) may provide valuable insight concerning model limitations in such cases.

7.3.5 Inverse simulation methods for the external validation of models

The use of inverse simulation methods, as discussed in Chapter 4, offers insight of a different kind from that found using conventional validation methods. This is especially true in the case of systems in which the immediate response to inputs involves integration. Drift is almost inevitably present in such systems and is due to small biases and offsets. Such offsets are unlikely to be the same in the system and the model, and can cause considerable difficulties when making model and system response comparisons, as the drift effects may have magnitudes similar to responses to the applied test inputs. This issue has been examined in detail in the context of simulated helicopter manoeuvres, where a strong case is made for the development of a validation strategy that integrates forward and inverse simulation [31]. Sensitivity analysis methods, as outlined in Chapter 5, can help in establishing the dependency of inverse simulation results on parameters of the model and may allow deficiencies in the model structure or parameter values to be established.

7.4 Issues of model validation and model quality in typical applications

There are cases, mainly involving safety-critical applications, such as in the nuclear industry and in the aerospace, defence and marine sectors, where rigorous model testing and formal approval schemes are routinely applied. However, the model development process used within many engineering organisations involves surprisingly little systematic investigation of model quality in terms of the useful range and accuracy limits of models.

The real system and the associated models should mature together, and model fidelity should increase as a design progresses. Whatever the approach being used for design, experience gained with the real system should feed into the modelling process at every stage.

Helicopter flight control system design is an example where model limitations can seriously affect the achievable performance. Until now, the success of modern design methods has been limited by the quality of the vehicle models available (see e.g. [29] and [32]). Similar situations apply in other application areas where the performance limits of a new system relate directly to the accuracy of the mathematical model upon which the design is based.

As mentioned in Section 7.3, the term 'model calibration' describes the processes of parameter estimation and other forms of interactive tuning that may be applied to a model during its development. As already mentioned, this is different from model validation and these two types of procedure are applied at different stages in the model development cycle.

For engineering applications, modelling is often associated with design, but models are also used in engineering for other purposes. For example, they form the basis for system simulators for operator training or education; they are used within automatic fault detection schemes and also in accident investigations. All such applications impose important requirements in terms of model quality. The sections that follow provide examples illustrating the application of external validation methods.

7.4.1 Case study 1: helicopter flight mechanics model development

The validation of helicopter flight mechanics models is important both for flight control system design and for human-factors investigations concerning flying qualities (see e.g. [33]). Important issues include the frequency range over which model quality needs to be assessed and the amplitude range for each variable. Frequency requirements extend beyond the range of human pilot control (0–5 rad/s approximately) to cover the whole range of frequencies that could be involved in active control of the vehicle (up to about 20 rad/s). Amplitudes, specified through translational and rotational velocities and accelerations, depend largely upon how the model is to be used.

System identification, parameter estimation and model optimisation techniques, discussed in Chapter 6, have been applied successfully to the development of linear helicopter models valid for small changes in flight conditions about a given trimmed state.

Checks of the overall robustness of models identified from flight data in this way must involve use of data sets that were not applied during the identification process. However, as discussed in Section 7.3.3, these additional data sets must be broadly similar in terms of their spectral properties and amplitude distribution to those used for identification. One approach simply involves carrying out more identification runs and comparing the different estimates. The situation would be judged unsatisfactory if variations in the values of parameters were greater than expected from variance estimates from the original identification. A slightly different approach is illustrated in Figure 7.2, where an identified model of a SA-330 Puma helicopter was assessed through a second (verification) stage involving an input that was not

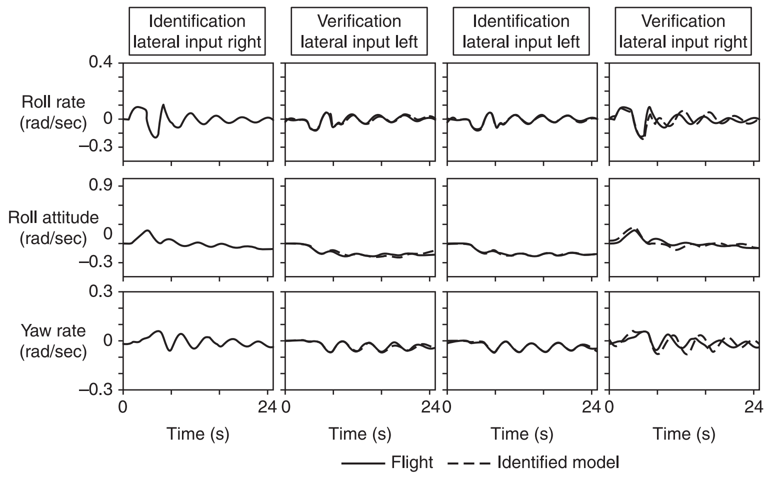


Figure 7.2 **Helicopter identification and verification results. An identified model of a Puma helicopter was tested at the verification stage by subjecting that model to an input that was not used in the identification process. The predicted output from the model was then compared with the corresponding measurements. Two cases are considered, both involving lateral control inputs. (The original version of this figure was published by the Advisory Group for Aerospace Research and Development, North Atlantic Treaty Organisation (AGARD/NATO) in AGARD Advisory Report 280 'Rotorcraft System Identification', in September 1991.)**

used in the identification process ([34] and [35]). In this verification stage, output from the model was compared with the corresponding measurements. Although an almost perfect fit was obtained between flight data and the identified model output, the fit was less good when the identified model was subjected to the new input. It is interesting to note that there is an asymmetry in these results and that the verification results are closer to the ideal (for all three variables considered) for the lateral left cyclic control input compared with those for the lateral right cyclic input. This could suggest that the model requires further fine-tuning in terms of responses to lateral cyclic inputs and the next step might well involve further sensitivity analysis to check for parameter inter-dependencies and other issues.

Nonlinear models of helicopters based mainly on physical principles can also be optimised using data from flight testing. One approach involves estimation of parameters of linearised models from flight data for different flight conditions (as suggested in Section 7.3.4). Subsequent analysis of these models for a range of flight conditions provides direct comparisons of estimated parameter values with equivalent theoretical values found by linearisation of the physically based nonlinear flight mechanics model [31]. Although such comparisons are valid only for small perturbations from a given trimmed condition, insight may be gained about the quality of the underlying nonlinear mathematical model from the comparisons at different operating points. Any consistent changes in the estimated value of a specific parameter when the flight condition (such as the forward speed of the vehicle) is changed can help in checking the credibility of the nonlinear model. Any difference between the trend in theoretical model parameter values and the trend in the estimated values for that parameter should lead to further investigations and possible changes in the nonlinear model. Figure 7.3 shows results of this kind for a series of flight experiments in which the aerodynamic derivatives L_v, L_p and N_r were estimated for an Aerospatiale Puma helicopter for forward speeds of 60 and 100 knots. It can be seen that for the parameters L_v and N_r, there is a fairly close match between theoretical and estimated values, and these were judged to be similar, within the precision of

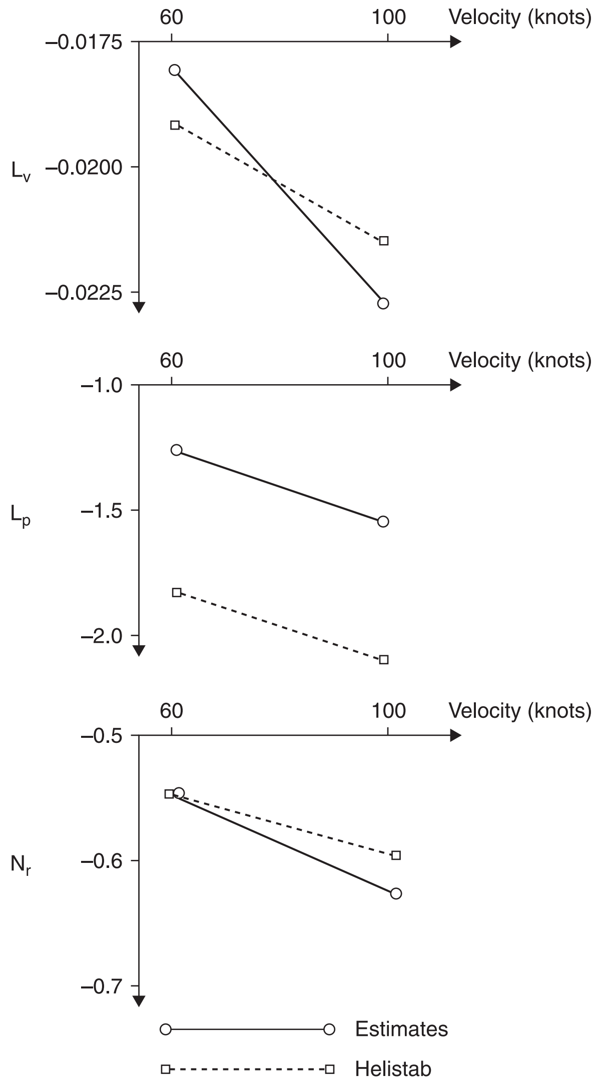


Figure 7.3 **Parameter values for two different flight conditions showing trends in predictions from a physically based nonlinear simulation model (HELISTAB) and corresponding trends in estimates from flight experiments using system identification methods (from [7.31])**

the estimation process used. On the other hand, the results for the parameter L_p show a similarity in trend but a difference in terms of absolute values, and this might justify further investigation of the nonlinear model.

Particular problems in experimental modelling of helicopters include the fact that these vehicles involve a high-vibration environment, allow only short test records due to marginally stable or unstable dynamics under open-loop test conditions, involve strong nonlinearities and operate in a non-uniform flow field. If the underlying reasons for any lack of agreement between estimated and theoretical trends are to be understood, it is also important to be able to relate parameters of the identified models to more fundamental quantities within the vehicle model, such as moments of inertia and aerodynamic parameters. There are also implicit relationships between parameters of linearised models and these must be properly understood if useful physical insight is to be gained.

One aspect of external validation that has been emphasised in helicopter flight mechanics modelling is that the external validation process may be viewed, as mentioned in earlier sections, as a form of model calibration aimed at establishing the range of operating conditions over which a model may be used successfully. Outside that range, the suitability of the model may be open to question. The external validation process can then address issues of model refinement or correction in order to extend the range of applicability.

As outlined in Section 7.3.5, it is believed that inverse simulation methods can provide additional useful information for external validation. For some output variables, the response of the vehicle to the pilot's control inputs initially involves integration and significant drift may therefore be present in measured responses. Divergence of simulated responses from equivalent flight data causes problems with conventional approaches to external validation, based on comparisons of measured and model responses. Recasting the problem so that it involves inverse simulation models (as discussed in Chapter 4) provides a possible way to avoid this difficulty, because comparisons between the system and the model are then being made of the system and model inputs needed to perform a specified manoeuvre. Figure 7.4 shows a comparison of flight data and inverse simulation results for a Lynx helicopter flying a longitudinal manoeuvre involving translation from an initial hover state to a hover state at

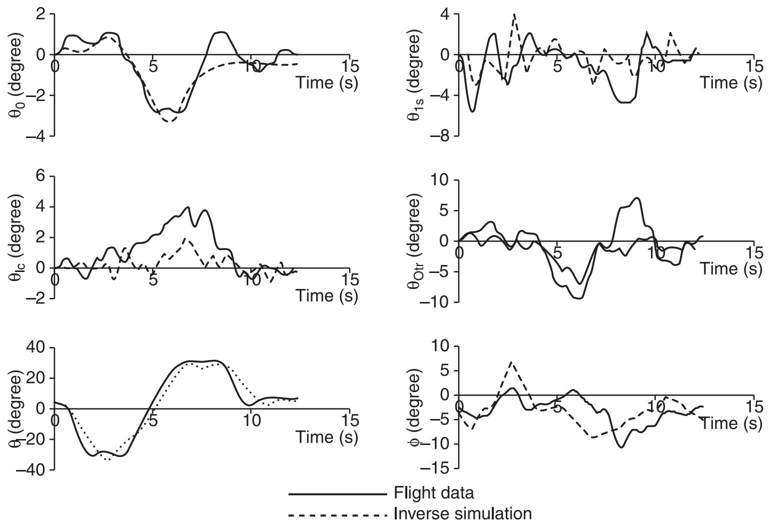


Figure 7.4 **Comparison of flight data and inverse simulation for a Lynx helicopter flying a 'quick-hop' longitudinal manoeuvre involving translation from an initial hover state to a hover state at another point close by, using the full available performance of the helicopter and with constant height and a fixed heading maintained throughout (from [7.31])**

another point close by, using the full available performance of the helicopter and with constant height and a fixed heading maintained throughout [31]. This is very different from tests involving small deviations about trimmed flight conditions and the results shown by the continuous lines of Figure 7.4 indicate a high level of pilot activity on all four controls. For inverse simulation, the desired flight path is defined and the heading is a constrained variable while other variables such as the pitch and roll angles are free to change with time. The results of the inverse simulation (discontinuous lines) show that these take realistic values, as do the control displacements. Although agreement between the flight data and the inverse simulation results are far from exact, they are sufficiently close to provide a basis for parameter estimation or optimisation, and sensitivity analysis of the inverse model, as discussed in Chapter 5, could provide a useful starting point.

7.4.2 Case study 2: model limitations in helicopter flight control system design

Good flight vehicle models are essential for the successful design of high-bandwidth full-authority active flight control systems for fixed-wing aircraft, helicopters and other forms of rotorcraft, such as tilt-rotor aircraft. Published examples show that the achievable performance of flight control systems may, in some cases, have been overestimated in initial design studies because of limitations in the flight mechanics models of the vehicle [33]. These problems may not be apparent until the flight testing stage, leading perhaps to costly redesign, extended flight test programmes and delays in certification. Improved modelling procedures and improved models offer significant benefits since, although control system designs can be made robust to compensate for poor model accuracy, this is usually achieved at the expense of performance.

Accurate linearised models are especially important in the early stages of helicopter flight control system design, as exemplified in the work of Manness and Murray-Smith [36] involving eigenstructure assignment methods. That paper shows clearly that confidence in the available model of a vehicle can allow demanding performance requirements to be satisfied. For high-performance flight control systems, it is vitally important to have highly accurate models of the vehicle in a frequency range that includes the frequencies where the phase lag of the forward path system transfer function approaches 180 degrees (the so-called ‘crossover’ region). Model uncertainties within that range lead to difficulties in guaranteeing stability and performance requirements in the closed-loop system design. Similar situations also apply in other applications in which the performance limits of a new system relate directly to the accuracy of the mathematical model upon which the design is based and where the success of control system design methods has been limited by the accuracy of the plant model.

Helicopters show significantly nonlinear behaviour over much of their useful flight envelopes and there is a need both for linearised models for the initial stages of control system design and for externally validated nonlinear simulation models to be used in assessing overall performance at a later stage. Issues of experimental design for external validation are important in this context. For example, as has already been pointed out, the frequency content of test input signals for a model intended for control system design applications must be chosen to give due emphasis to frequencies close to the nominal crossover frequency.

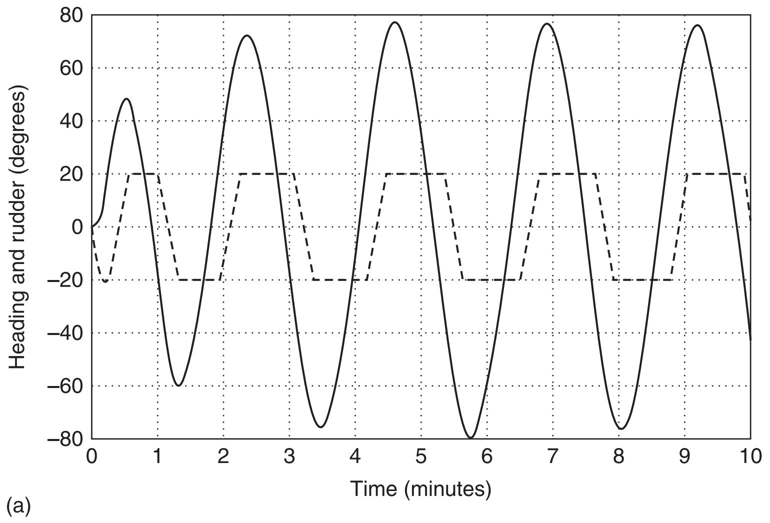
7.4.3 Case study 3: model quality issues in the design of a ship steering control system

Accurate navigation and autopilot system design are important issues for engineers in the marine field. Ship steering systems provide an interesting illustration. In order to make a large manoeuvre, a large turning moment must be generated by the flow of water over the deflected rudder. The magnitude of this depends on the rudder dimensions and the forward speed of the vessel. The rudder and associated actuators therefore need to be represented accurately in any model being used in the design of a ship steering control system. As the size of vessels has increased, as in the case of oil tankers and container ships, new problems have arisen in terms of the rudder dimensions in relation to the size of the vessel.

One established rudder model is based on equations developed by Fossen [37] using data from a paper by van Berlekom and Goddard [38]. An aspect of that model which has been the subject of debate relates to the representation of water flow over the rudder surface and the fact that the model has limitations for very large vessels. The importance of this limitation of the model became more apparent when it was used by Çimen and Banks [39] as a possible basis for the design of a nonlinear optimal controller for a large oil tanker. As pointed out by McGookin and Murray-Smith [40], the rudder sub-model used by Çimen and Banks involved terms which made the turning moment too large for a vessel of the size considered. The heading dynamics were then unrealistically sensitive to changes of rudder angle so that, in simulation studies, the vessel responded much more rapidly than it should to controller commands.

Figure 7.5(a) shows results obtained in a test involving a standard zigzag type of open-loop manoeuvrability test for a simulation based on the form of model used by Çimen and Banks. The manoeuvre involves a 20-degree step applied to the rudder position until the heading angle changes by 20 degrees. Then the rudder angle is changed to –20 degrees and the process is repeated until a steady oscillation is produced in the heading angle. The data of van Berlekom and Goddard [38] for a vessel of this size shows a sinusoidal trajectory in heading angle of a period of 8 minutes and peak magnitude of 33 degrees. The simulated results for the case of Figure 7.5(a) show a heading angle response with a period of only 2.2 minutes and amplitude about 75 degrees. The period is therefore much smaller than that found by van Berlekom and Goddard, and the amplitude of the oscillation are too large. A modified model [40], which better represents the manoeuvring capability of a vessel of this size [41], produces results shown in Figure 7.5(b). Comparing these results with those for the unmodified model, it can be seen that the heading angle has an oscillation period of about 7.7 minutes and the peak magnitude is 34 degrees. These values are now similar to those reported by van Berlekom and Goddard [38] for a real vessel of this size. These simulation results suggest that the modified open-loop ship model is significantly more realistic than the original description.

This case study shows that, although advanced controller design methodologies represent a potentially useful step, such developments must involve use of a model that is fit for purpose. Simulated results for controllers designed with an inappropriate model may be viewed by design engineers with real ship steering experience as unrealistic and this may give rise to unnecessary and unjustified doubts about other marine applications of advanced control system design methods.



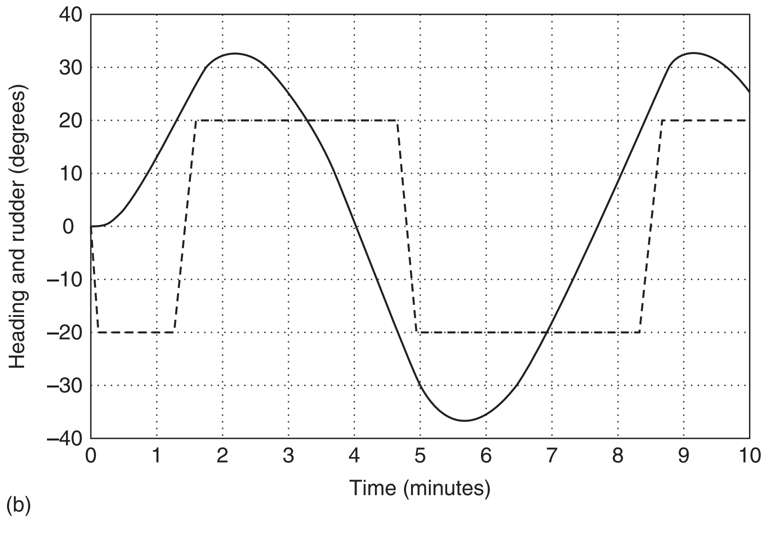


Figure 7.5 **Results for simulated open-loop manoeuvring test using two models of 190,000 dwt tanker vessel: (a) results for the unmodified model and (b) results for modified model (from [7.40]). The dashed lines represent the rudder angle and continuous lines represent the Heading angle**

7.5 Issues of model quality in model reduction

The graphical techniques and quantitative measures mentioned in Section 7.3.1 can be applied to situations in which one model is being related to another, as well as for comparisons with data from real systems. This is really a form of model comparison and can arise in situations where complex, computationally demanding and externally validated models are available but simpler simulation models which have shorter run times on the computer are needed for use in system design applications or real-time simulation.

For many applications, it is useful to retain a physically based interpretation and, recently, energy-based metrics have begun to appear which can be applied to nonlinear as well as linear descriptions. This energy-based approach and links with bond-graph structures have also been associated with the concept of a *proper* model which, in this particular context, is defined as the model with the minimal set of physical parameters required to predict dominant system dynamics ([42], [43] and [44]). Proper models tend to be more efficient in computational terms, which is helpful when models have to be simulated repeatedly. They have been associated with the concept of *model order deduction*, rather than the idea of order reduction, which starts with a given high-order description.

Energy-based modelling metrics can be linked to bond-graph methods of model development and analysis, and help ensure that physical insight is retained when using the model for design. In this approach the removal of physical phenomena that are unimportant for a proposed application is based on the power associated with each element of the model. It is argued that any element dissipating or storing a significant part of the total power supplied to the system contributes significantly to the overall behaviour. A suitable metric is based on the energy flows into and out of a given element over a given time interval. The larger the energy metric, the greater the contribution of the element to the system behaviour and this energy metric thus provides a form of sensitivity measure. An interesting account of the application of this energy-based approach to the reduction of models of integrated hybrid vehicle systems may be found in the work of Louca and Yildir [45], where it was established that a reduced model for a medium-sized military truck could give predictions very similar to those from a full model but running 2.5 times faster.

Reduced-order models are valuable in many fields. For example, when investigating aircraft handling qualities, the model of the vehicle must have appropriate accuracy over a defined frequency range that is important for interactions between the pilot and the vehicle. Similarly, as already discussed, it is often important to ensure that the plant model in control system design applications is accurate for frequencies near the gain crossover frequency, but lower levels of accuracy may be tolerated at frequencies far from that critical range.

As mentioned in Section 3.6, another field in which model reduction is very important is in the development of micro- and nano-electromechanical systems (MEMS and NEMS). Finite element and molecular dynamics models are commonly used for some aspects of MEMS and NEMS systems analysis, but in practical applications some form of model reduction is usually essential for the later stages of the design in order to avoid excessive computational complexity [46]. System identification and parameter estimation techniques have been applied successfully in fitting lower-order lumped dynamic descriptions based on grey-box ideas to the more detailed underlying physically based model (see e.g. [47] and [48]).

Model optimisation methods, involving genetic algorithms and artificial neural networks, have been used successfully and other approaches such as model predictive control (see e.g. [49]) have also been applied with reduced models. Being able to move efficiently from highly detailed physically based models to reduced models and then back to physically based models for further analysis is important. In Section 3.6 it was mentioned that multi-scale modelling concepts have been developed for this application area. One approach involves *sequential multi-scale modelling* methods, in which large-scale models (such as those used for control system design) use low-resolution representations derived through model reduction from detailed higher-resolution descriptions. The simulations at these different levels run independently of each other (see e.g. [50]). Another approach is termed *concurrent multi-scale modelling*, involving a combined model in which the different scales are considered at the same time (see e.g. [51]).

7.5.1 Case study 1: frequency-domain methods applied to aircraft models

One example of a relatively simple model reduction method for MIMO systems involves a frequency-domain complex curve-fitting approach and has been applied to the development of MIMO models for flight control system design [52]. The approach is based on transfer function models and uses a modified least-squares approach to fit transfer functions to the target frequency response data. The method involves minimisation of a sum of the squares of the differences between the absolute magnitudes of the frequency response values for the high-order system and the reduced model over a specific range of frequencies.

A frequency-weighted cost function is used for optimisation, allowing errors in chosen parts of the frequency range to be given particular emphasis. Results for the case of a large transport aircraft model are presented in Figure 7.6 and show frequency responses of a two-input two-output fourth-order model together with the corresponding results for a two-input two-output second order description for the frequency range 0.1–100 rad/s. It can be seen that, for the lower part of the range of frequencies used (<10 rad/s), the response of reduced-order model agrees well with that of the original model. The transfer functions from rudder and aileron

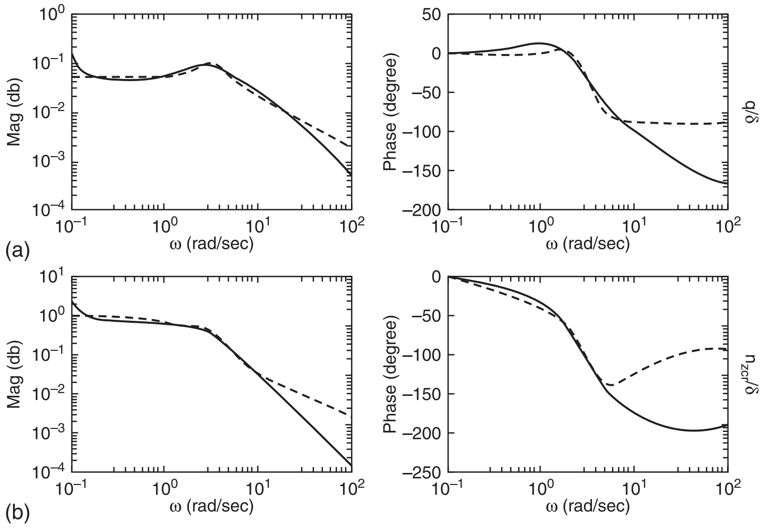


Figure 7.6 Frequency responses of a single-input two-output eighth-order model of a fighter aircraft and an equivalent third-order model over the frequency range 0.1 rad/s < ω < 100 rad/s. In (a) the responses are for pitch rate to elevator stick force, while (b) shows responses for normal acceleration to elevator stick force. The responses of the higher-order model are shown by the continuous curve and the dashed lines show the corresponding response for the reduced-order description (from [7.52])

deflection to yaw rate, sideslip and roll rate all have the same denominator since the high-order model is derived from a state variable description and the eigenvalues of the state matrix determine the denominators for all the transfer functions.

7.5.2 Case study 2: a simulation of a hydro turbine and electrical generator system

A further illustration of model reduction involves the development and application of a model of a hydro-electric generator system [53], which is also discussed in the context of real-time simulation in Chapter 8. The purpose of the model was to provide a basis for the design of a new faster-acting electronic governor for speed control of a 35.75 MW hydro-turbine generator. A number of different types of electronic governor were to be investigated as candidates to replace an existing and slower type of mechanical-hydraulic governor system. An accurate, physically based model of the plant, capable of being operated in real time, was required.

Although some tests on the real system were permitted during the model development process, dynamic tests on the pipeline system, which is a vitally important part of the overall model of the system, were severely restricted for safety reasons. Extensive modelling of the pipeline network had been undertaken previously by engineers employed by the North of Scotland Hydro-Electric Board (NSHEB) and a well-proven finite-element model existed, although this relatively complex and numerically intensive model could not be implemented within a real-time simulation.

A decision was made to develop a lumped-parameter model of relatively low order that could capture the main features of the more complex finite-element pipeline model over the most important part of the frequency range for turbine control. Several lumped-parameter descriptions were tested against the finite-element description using frequency-domain plots to find a model of acceptable accuracy that could be implemented easily, in real-time, using available computing facilities.

Figure 7.7 shows a schematic diagram of the system. At the power station there are four turbines and generators, and below the surge shaft the penstock divides initially into two and then into four pipelines. Any disturbance in terms of head (pressure) or flow will be transmitted along the pipeline, with reflections occurring at discontinuities. Resonance phenomena can be expected at the characteristic frequencies of the complete pipe system.

Equations can be derived by considering momentum balance and continuity of flow for an infinitesimal length of pipe along with a relationship describing the change of fluid density with change of pressure (the bulk modulus) and a

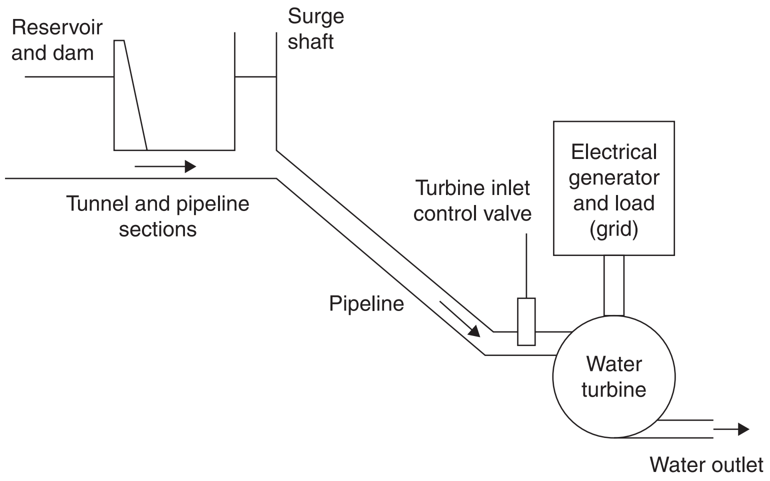

Reservoir and dam
Surge shaft
Tunnel and pipeline sections
Turbine inlet control valve
Electrical generator and load (grid)
Pipeline
Water turbine
Water outlet

Figure 7.7 **Schematic diagram of the hydro-turbine system. Control of the water turbine is through guide-vanes and associated linkages, but is represented for simplicity in this diagram by a simple inlet control valve**

stress-strain relationship for the pipeline [54]. The basic equations are:

$$Ag\frac{\partial H}{\partial x}+V\frac{\partial V}{\partial x}+\frac{\partial V}{\partial t}+\frac{fV|V|}{2D}=0 \tag{7.7}$$

$$V\frac{\partial H}{\partial x}+\frac{\partial H}{\partial t}-V\sin\theta+\frac{V_p^2}{g}\frac{\partial V}{\partial x}=0 \tag{7.8}$$

where *H* is the head of water relative to the exit from the turbine, *V* is the velocity of water, *x* is the position along the pipe, *D* is the pipe diameter, *g* is the gravitational constant and *t* is time. The quantity V_p is the velocity of travelling waves within the pipe and the quantity *f* is the surface friction coefficient of the pipeline section. The cross-sectional area (*A*) relates the flow, *Q*, to the velocity *V* through the equation $Q = AV$. It should be noted that the absolute value sign in Equation (7.7) is introduced to ensure that friction opposes the motion.

After a simplification process in which small terms are neglected, these equations become:

$$Ag\frac{\partial H}{\partial x}+\frac{\partial Q}{\partial x}+\frac{\partial V}{\partial t}+\frac{fQ|Q|}{2DA}=0 \tag{7.9}$$

$$\frac{\partial H}{\partial t}+\frac{V_p^2}{gA}\frac{\partial Q}{\partial x}=0 \tag{7.10}$$

It should be noted that if the surface friction effect in the pipe were represented by a linear function, Equations (7.9) and (7.10) would be similar in structure to the voltage and current relationships for an electrical transmission line section, and useful analogies may be drawn between pipeline and transmission line models. One particularly important analogy involves the concept of hydraulic impedance, which is defined for the pipeline, in a similar way to electrical impedance, as:

$$Z = \frac{H}{Q} \tag{7.11}$$

A lumped parameter model for a simple pipe section was derived by defining a number of internal points along the pipe and transforming the spatial derivatives into first-order difference functions. This lumped model involved a pipe section having one interior point and two end points, with friction effects lumped at the centre point. For situations involving one active turbine (as in the site tests), it was found that five pipe section models, each involving four integrator blocks, provided a pipeline model that was suitable for the intended application. The number of sections used was justified by carrying out an impedance test on this lumped model and comparing the results with an equivalent test using the finite element model based on Equations (7.9) and (7.10). Over the frequency range from 0 to 1 Hz, the reduced model was found to capture the frequency peaks of the impedance diagram of the full finite-element model with sufficient accuracy.

The reduced representation of the pipeline was then integrated into the overall model and external validation was applied using data from dynamic tests on the complete system. Initial testing for the purposes of external validation involved frequency responses over a range that included the first peak in the hydraulic impedance diagram (frequencies up to 0.37 Hz). These tests had to be performed for a number of input amplitudes due to the nonlinearities within the model, especially backlash in the guide-vane linkages and rate and amplitude limits within the main servomotor that controls the guide vanes.

Some adjustments of the model took place following critical assessments of the frequency response test results and the complete simulation was then also subjected to detailed evaluation and testing for conditions involving larger disturbances. This involved system-splitting tests in which part of the local distribution system was configured so that the generator, together with an associated load, was connected to the rest of the grid through a single circuit breaker. Current transfer through that circuit breaker could be monitored and the system could be split when an appropriate transfer had been achieved. This allowed a form of step testing to be carried out. Tests were performed at various loading levels of up to 25 MW with export or import levels at the time of splitting which approximated to a step of the order of 1.5 MW, and results from the tests on the real system and equivalent tests on the simulation model were found to be similar in broad terms.

Certain features of the system behaviour were of particular interest, such as the effect of backlash in the guide-vane linkage mechanism, which has a very direct effect on limit cycle behaviour when the system is operating under isolated load conditions. The backlash not only influences the amplitude and period of the limit cycle, but also the transient characteristics of the closed-loop system. Also of particular importance was the fact that the results showed that no large pressure fluctuations occurred within the system for the operating conditions and governor types considered, and that the damping of disturbances was satisfactory. However, the system-splitting test on the real system resulted in a power change which was only a rough approximation to a step function and repeated trials had to be performed for each test condition for power import and power export, to provide a good basis for quantitative comparisons of the model and the real system.

Although not used in the original application, the polygon type of graphical display discussed in Section 7.3.1 could have relevance in this complex type of situation where particular features of the response are of interest rather than the exact time histories. Figure 7.8 shows a polygon that

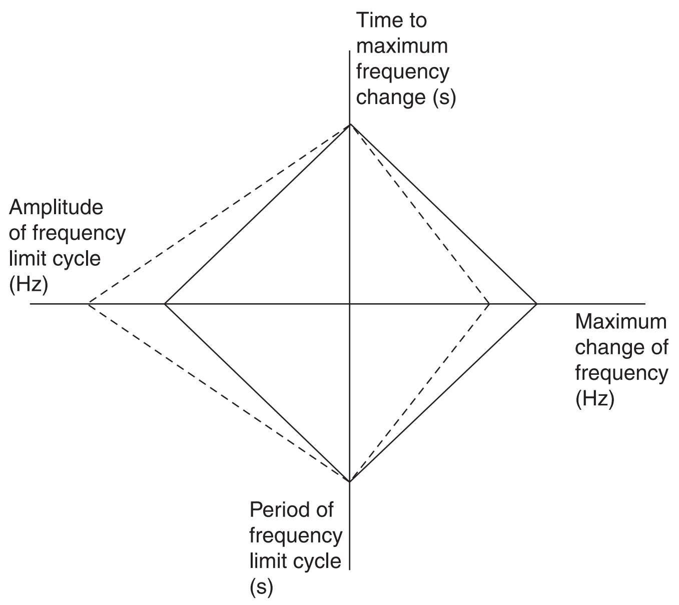

Time to maximum frequency change (s)
Amplitude of frequency limit cycle (Hz)
Maximum change of frequency (Hz)
Period of frequency limit cycle (s)

Figure 7.8 **Polygon diagram for the comparison of experimental and simulation results for the hydro-turbine simulation model**

could provide a basis for comparing key information obtained from testing of this kind. In this case, only four quantities are being compared. These are the peak value in terms of the transient change of frequency (Hz) following the step import or export of power, the time of occurrence of the peak (s), the time to the peak value following the application of the step, the amplitude of limit cycle (Hz peak to peak) observed in the frequency record under steady-state conditions following the step and the period of the limit cycle.

Such diagrams could allow investigation of the sensitivity of key features of the model behaviour for changes in model parameters, or structure, or for different types of governor.

At an early stage in the external validation of the model, an evaluation was made of the real-time simulation with the existing mechanical-hydraulic governor by experienced operators from the power station. They were able to provide subjective feedback which was very helpful and allowed improvements to be made to the model, especially in terms of fine-tuning of some nonlinear characteristics that could otherwise lead to unrealistic limit cycles for some operating conditions. This is one example of the possible benefits of subjective assessment in the external validation of a simulation model. An experienced operator may detect aspects of a simulation, especially when operated in real time, which are not typical of the real physical system but may not be obvious from an examination of plant and simulation records over a short period within a single test. Under some specific conditions, the 'feel' of the real-time simulation may differ significantly from the system itself and establishing the reasons for this can often pinpoint deficiencies in the model.

The eventual approval of the simulation by the NSHEB engineers allowed it to be used in real time as a basis for evaluation of a number of novel analogue and digital electronic governor systems under a wide range of operating conditions prior to their installation and testing on-site, as discussed in the case study in Section 8.3.1.

7.6 Discussion

In some applications, such as helicopter flight mechanics, difficulties can be encountered due to the presence of close coupling of variables and parameters within the system. The fact that such systems are inherently multi-input multi-output in form means also that a number of output variables of the system have to be considered simultaneously and this introduces additional problems. Although quantitative measures of model performance are appropriate for applications of this kind, the use of such criteria reduces the model quality assessment process to consideration of values of a single index, which masks the true complexity of the situation and provides little or no physical insight. Improved visualisation methods are needed for multi-output situations and for models where there can be strong interactions between parameters.

The polygon diagrams discussed in Sections 7.3.1 and 7.5.2 represent a potentially interesting development in terms of visualisation. Such diagrams provide a basis for comparing different results from different models and this means that such an approach has potential for interrelating results obtained from models at different levels of resolution. It can thus provide insight about the consistency of simple descriptions used for conceptual design at the initial stages of a project and highly detailed models that are used at a much later stage. One advantage of the polygon diagram approach to visualisation is that it is extremely flexible in terms of the comparisons that can be made. For example, it allows results of sensitivity analysis to be displayed in a simple and efficient fashion and is applicable to problems in many areas where there is a need to depict relationships among multivariable data.

Figure 7.9 provides a summary of the processes of model development and testing, including internal verification and external validation, and attempts to summarise the main steps involved in that cyclic process. The blocks associated with the real system and system test data are shown by shadowed boxes. Steps concerned directly with the model, its purpose, modelling techniques used, processes of external validation, decisions on model adequacy for the intended application and documentation are represented by simple blocks with a white background.

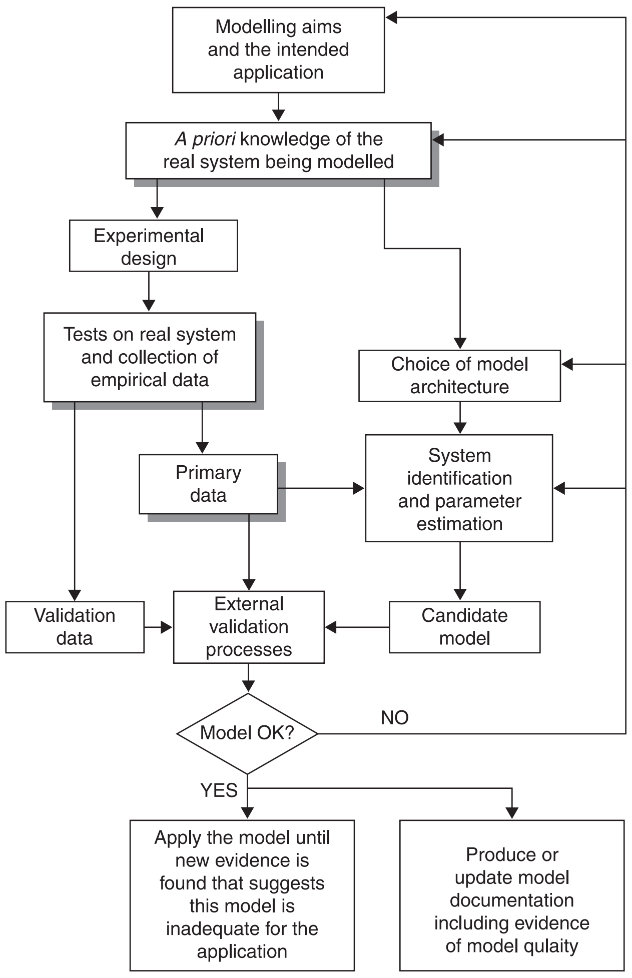

Modelling aims and the intended application
A priori knowledge of the real system being modelled
Experimental design
Tests on real system and collection of empirical data
Choice of model architecture
Primary data
System identification and parameter estimation
Validation data
External validation processes
Candidate model
Model OK?
NO
YES
Apply the model until new evidence is found that suggests this model is inadequate for the application
Produce or update model documentation including evidence of model qulaity

Figure 7.9 **Block diagram of iterative processes of model development showing the stages from formulation of modelling objectives to external validation and testing of model adequacy. Blocks associated with the real system and system test data are shown by shadowed boxes. Blocks relating to the model, its purpose, modelling techniques used, external validation processes and documentation are shown as simple blocks with a white background**

The structure of the diagram emphasises the iterative nature of the procedures, starting from the statement of modelling objectives and the *a priori* assumptions being made, and ending with a model that is credible and fit for purpose in terms of the intended application. Although it represents the processes in terms of comparisons of selected model variables and equivalent measured variables, most features of this diagram could apply equally to comparisons of reduced models with established high-order descriptions.

What is particularly attractive about this form of diagram is that it emphasises the vitally important role of external validation and the importance of prior knowledge about the real system. If a model proves inadequate for the intended application when subjected to the rigorous processes of external validation, there is a possibility of correction through feedback. Feedback pathways lead not only to the blocks representing the model but also, through the block representing knowledge of the real system, to the blocks showing involving experimental design and thus to further tests to collect additional data from the real system.

Confidence in a model prediction depends on the level of confidence in sub-models and each sub-system model could be subjected to the development and testing procedures of Figure 7.9. Exhaustive testing of sub-models allows confidence to be established at that level first and then extended gradually to involve evaluation of the complete system model for a number of experimental situations.

In the development of entirely new systems, experimental data from the complete system are not available at the design stage. However, in some cases, historical data from earlier systems of a similar kind can sometimes be used in reviewing proposals for a model for some new projects and for comparing initial design options. Successful application of this approach depends on good documentation of models of existing systems and of the tests carried out in evaluating them, as discussed in Chapter 9.

7.7 References

[1] Sargent, R.G. (1979) 'Validation of simulation models', in Highland, H.J. (ed.), *Proceedings of the 1979 Winter Simulation Conference*, Vol. 2, pp. 497–503, IEEE Press, Piscataway NJ, USA.

[2] Ören, T.I. (1981) 'Concepts and criteria to assess acceptability of simulation studies: a frame of reference', in Adam, N. (ed.), *Simulation Modeling and Statistical Computing*, Communications of the ACM, Vol. 24, No. 4, pp. 180–8.

[3] Balci, O. (1997) 'Principles of simulation model validation', *Transactions of the Society for Computer Simulation International*, Vol. 14, No. 1, pp. 3–12.

[4] Balci, O. (1997) 'Verification, validation and accreditation of simulation models', in Andradóttir, S., Healy, K.J., Withers, D.H. and Nelson, B.H. (eds), *Proceedings of the 1997 Winter Simulation Conference*, IEEE Computer Society, Washington DC, pp. 135–47.

[5] Balci, O. (2004) 'Quality assessment, verification and validation of modeling and simulation applications', in Ingailis, R.G., Rossetti, M.D., Smith, J.S. and Peters, B.A. (eds), *Proceedings of the 2004 Winter Simulation Conference*, pp. 122–9, Winter Simulation Conference, USA.

[6] Brade, D. (2000) 'Enhancing modeling and simulation accreditation by structuring verification and validation results', in Joines, J.A., Barton, R.R., Kang, K. and Fishwick, P.A. (eds), *Proceedings of the 2000 Winter*

Simulation Conference, pp. 840–8, Society for Modelling and Computer Simulation International, La Jolla CA, USA.

[7] Brade, D. (2003) *A Generalized Process for the Verification and Validation of Models and Simulation Results*, Dissertation submitted for award of the degree Dr rer. nat., Fakultät für Informatik, Universität der Bundeswehr, München, Germany.

[8] Murray-Smith, D.J. (1998) 'Methods for the external validation of continuous system simulation models: a review', *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 4, No. 1, pp. 5–31.

[9] SCS Technical Committee on Model Credibility (1979) 'Terminology for model credibility', *Simulation*, Vol. 32, No. 3, pp. 103–4.

[10] Sargent, R.G. (1982) 'Verification and validation of simulation models', in Cellier, F.E. (ed.), *Progress in Modelling and Simulation*, pp. 159–69, Academic Press, London, UK.

[11] Oberkampf, W.L., Trucano, T.G. and Hirsch, C. (2002) 'Verification, validation and predictive capability in computational engineering and physics', Invited Paper, *Workshop on Foundations for Verification and Validation in the 21st Century*, 22–23 October 2002, Johns Hopkins University, Laurel MD.

[12] Hemez, F.M. (2004) 'The myth of science-based predictive modelling', *Foundations '04 Workshop for Verification, Validation and Accreditation (VV&A) in the 21st Century*, 13–15 October.

[13] Tufte, E.R. (1983) *Visual Display of Quantitative Information*, Graphics Press, Cheshire CT, USA.

[14] Murray-Smith, D.J. (1995) *Continuous System Simulation*, Chapman and Hall, London, UK.

[15] Knudsen, M. (2006) 'Experimental modelling of dynamic systems: an educational approach', *IEEE Transaction on Education*, Vol. 49, No. 1, pp. 29–38.

[16] Smith, M.I., Murray-Smith, D.J. and Hickman, D. (2007) 'Mathematical and computer modeling of electro-optic systems using a generic modeling approach', *Journal of Defense Modeling and Simulation*, Vol. 4, No. 1, pp. 3–16.

[17] Smith, M.I., Murray-Smith, D.J. and Hickman, D. (2007) 'Verification and validation issues in a generic model of electro-optic sensor systems', *Defense Modeling and Journal of Simulation*, Vol. 4, No. 1, pp. 17–27.

[18] Grant, A. and Murray-Smith, D.J. (2004) 'Polygon imaging methods in plant monitoring and model validation applications', in *Proceedings of the Advanced Engineering Design 2004 Conference, Glasgow,* September 2004, Paper C1.07, Orgit, Prague, Czech Republic.

[19] Grant, A.G.N. (2006) *Data to Polygon Transformation and Comparison: A Technique for Model Validation and Related Applications*, PhD Thesis, University of Glasgow.

[20] Balci, O., Nance, R.E., Arthur, J.D. and Ormsby, W.F. (2002) 'Expanding our horizons in verification, validation, and accreditation research and practice', Yücesan, E., Chen, C.-H., Snowdon, J.L. and Charnes, J.M. (eds), *Proceedings of the 2002 Winter Simulation Conference*, pp. 653–63, Winter Simulation Conference, USA.

[21] Cloud, D.J. and Rainey, L.B. (eds) (1998) *Applied Modeling and Simulation: An Integrated Approach to Development and Operation*, McGraw-Hill, New York, USA.

[22] Zeigler, B.P., Praehofer, H. and Kim, T.G. (2000) *Theory of Modeling and Simulation*, Second Edition, pp. 367–89, Academic Press, San Diego CA, USA.

[23] Brade, D. and Köster, A. (2001) 'Risk-based validation and verification levels definition', *Proceedings of the European Simulation Interoperability Workshop*, London, UK, Simulation Interoperability Standardization Organization.

[24] Brade, D., Maguire, R. and Lotz, H.-B. (2002) 'Arguments-based credibility levels', in *Proceedings of the SISO Spring Simulation Interoperability Workshop*, Orlando FL, USA.

[25] Kit, E. (1995) *Software Testing in the Real World*, Addison Wesley, Harlow, UK.

[26] Murray-Smith, D.J. (2001) 'The validation of computer-based models in engineering: some lessons from computing science', *Acta Polytechnica*, Vol. 41, No. 4–5, pp. 45–8.

[27] Padfield, G.P. and Du Val, R.W. (1991) 'Application areas for rotorcraft system identification: simulation model validation', in *AGARD Lecture Series 178, Rotorcraft System Identification*, pp. 12-1–12-39, AGARD, Neuilly sur Seine, France.

[28] Kocijan, J., Girard, A., Banko, B. and Murray-Smith, R. (2005) 'Dynamic systems identification with Gaussian processes', *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 11, No. 4, pp. 411–24.

[29] Tischler, M.B. (1996) 'System identification methods for aircraft flight control development and validation', in Tischler, M.B. (ed.) *Advances in Aircraft Flight Control*, Taylor and Francis, London, UK, pp. 35–69.

[30] Tischler, M.B. and Remple, R.K. (2006) *Aircraft and Rotorcraft System Identification*, AIAA, Reston VA, USA.

[31] Bradley, R., Padfield, G.D., Murray-Smith, D.J. and Thomson, D.G. (1990) 'Validation of helicopter mathematical models', *Transactions of the Institute of Measurement and Control*, Vol. 12, No. 4, pp. 186–96.

[32] Tischler, M.B. (1990) 'System identification requirements for high bandwidth rotorcraft flight control systems', *AIAA Journal of Guidance, Control and Dynamics*, Vol. 13, No. 5, pp. 835–41.

[33] Hamel, P. (1994) 'Aerospace vehicle modelling requirements for high bandwidth flight control', in Cook, M.V. and Rycroft, M.J. (eds), *Aerospace Vehicle Dynamics and Control*, pp. 1–31, Clarendon, Oxford, UK.

[34] Padfield, G.D. (1991) 'SA 330 Puma identification results', in *AGARD Lecture Series 178: Rotorcraft System Identification (AGARD-LS-178)*, Section 10, AGARD, Neuilly sur Seine, France.

[35] Murray-Smith, D.J. (principal author) (1991) 'Robustness Issues', in *AGARD Advisory Report 280, Rotorcraft System Identification, (AGARD-AR-280)*, pp. 213–22. AGARD, Neuilly sur Seine, France.

[36] Manness, M.A. and Murray-Smith, D.J. (1992) 'Aspects of multivariable flight control law design for helicopters using eigenstructure assignment', *Journal of the American Helicopter Society*, Vol. 37, No. 3, pp. 18–32.

[37] Fossen, T.I. (1994) *Guidance and Control of Ocean Vehicles*, Wiley, Chichester, UK.

[38] van Berlekom, W.B. and Goddard, T.A. (1972) 'Maneuvering of large tankers', *Transactions of SNAME*, Vol. 80, pp. 264–98.

[39] Çimen, T. and Banks, S.P. (2004) 'Nonlinear optimal tracking control with application to super-pilots for autopilot design', *Automatica*, Vol. 40, No. 11, pp. 1845–63.

[40] McGookin, E.W. and Murray-Smith, D.J. (2006) 'Comment on "Nonlinear optimal control with applications to supertankers for autopilot design" by T. Çimen and S.P. Banks', *Automatica*, Vol. 42, pp. 2223–5.

[41] McGookin, E.W., Murray-Smith, D.J., Li, Y. and Fossen, T.I. (2000) 'Ship steering control system optimisation using genetic algorithms', *Control Engineering Practice*, Vol. 8, pp. 429–43.

[42] Wilson, H.H. and Stein, J.L. (1995) 'An algorithm for obtaining proper models of distributed and discrete systems', *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 117, No. 4, 534–40.

[43] Walker, D.G., Stein, J.L. and Ulsoy, A.G. (2000) 'An input-output criterion for linear model deduction', *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 122, No. 3, 507–13.

[44] Rideout, D.G., Stein, J.L. and Louca, L.S. (2009) 'Power-based dynamic system model decoupling and reduction using bond graphs', *Simulation Modelling Practice and Theory*, Vol. 17, No. 1, pp. 271–92.

[45] Louca, L.S. and Yildir, U.B. (2006) 'Model and reduction techniques for studies of integrated hybrid vehicle systems', *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 12, No. 2–3, pp. 203–18.

[46] Ferreira, A. and Sumeet, S.A. (2011) 'A survey of modelling and control techniques for micro- and nanoelectromechanical systems', *IEEE Transactions on*

Systems, Man and Cybenetics – Part C: Applications and Reviews, Vol. 41, No. 3, pp. 350–64.

[47] Borovic, B., Lewis, F.L., Agonafer, D., Kolesar, E.S., Hossain, M.M. and Popa, D.O. (2005) 'Method for determining a dynamical state-space model for control of thermal MEMS devices', *IEEE J. Microelectromechanical Systems*, Vol. 14, No. 5, pp. 961–70.

[48] Chen, J., Kang, S.-M., Zou, J., Liu, C. and Schutt-Aine, J.E. (2004) 'Reduced-order modelling of weakly nonlinear MEMS devices with Taylor series expansion and Arnoldi approach', *IEEE Journal of Microelectromechanical Systems*, Vol. 13, No. 3, pp. 441–51.

[49] Bleris, L.G. and Kothare, M.V. (2005) 'Real-time implementation of model predictive control', in *Proceedings of the American Control Conference*, 8–10 June, Vol. 6, pp. 4166–71.

[50] To, A.C., Liu, W.K., Olson, G.B., Belytschko, T., Chen, W., *et al.* (2008) 'Materials integrity in microsystems: a framework for a petascale predictive-science-based multiscale modelling and simulation system', *Computational Mechanics*, Vol. 42, No. 4, pp. 485–510.

[51] Xu, Y. and Alaru, N. (2008) 'Multiscale electrostatic analysis of silicon NEMS via heterogeneous quantum models', *Phys. Rev. B*, Vol. 77, No. 7, pp. 075313-1–075313-13.

[52] Gong, M., and Murray-Smith, D.J. (1993) 'Model reduction by an extended complex curve-fitting approach', *Transactions of the Institute of Measurement and Control*, Vol. 15, No. 4, pp. 188–98.

[53] Bryce, G.W., Foord, T.R., Murray-Smith, D.J. and Agnew, P. (1976) 'Hybrid simulation of water-turbine

governors', in Crosbie, R.E. and Hay, J.L. (eds), *Simulation Councils Proceedings Series*, Vol. 6, No. 1, pp. 35–44, Simulation Councils Inc, La Jolla CA, USA.

[54] Wylie, E.B. and Streeter, V.L. (1978) *Fluid Transients*, McGraw-Hill, New York, NY, USA.

Real-time simulation, virtual prototyping and partial-system testing

Abstract: This chapter deals mainly with development of virtual prototypes and with testing procedures in which equipment from a real system is evaluated within a simulation environment. This leads to requirements for real-time performance and to 'hardware-in-the-loop' simulation in which some elements of the system are represented only in software. Real-time simulation methods, including multi-rate simulation techniques, are described and examples considered include real-time simulation of a hydro-turbine generator system and an unmanned underwater vehicle.

Key words: partial system testing, real-time simulation, multi-rate simulation, hardware-in-the-loop testing, emulation.

8.1 Virtual prototyping through simulation

The sequential and concurrent approaches to design outlined in Chapter 2 both involve the development of prototype systems. Virtual prototyping is involved, especially in the concurrent approach, as well as the development of a physical prototype. Simulation is central to this and a procedure known as *partial-system testing* is applied, which involves use of a computer simulation model operating in conjunction with hardware from the real engineering system. This is known as *hardware-in-the-loop* simulation and, in this approach, execution of the model must match exactly the timing of events within the real-world system. As well as these cases where a simulation model operates with external hardware, *real-time* situations are also important for applications in which the simulation model runs with a human operator as part of a manually operated system.

8.2 Real-time simulation methods

Real-time operation imposes important demands in terms of numerical integration, since the computational resource available for each integration time step is fixed. Variable-step integration methods are therefore inappropriate for real-time simulation (see e.g. [1]) and concepts of numerical error control for variable-step algorithms become irrelevant. Any fixed-step integration method used for real-time simulation must achieve real-time performance while also keeping numerical errors as small as possible.

Simple *fixed-step* techniques, such as rectangular or trapezoidal integration, are widely used for real-time applications. Analysis of the underlying mathematical models may have to be carried out to find the model eigenvalues for chosen operating conditions before the integration step size can be chosen. Informed decisions may then be taken about any simplifications for the real-time version of the model. Time constants that are small compared with others might be neglected and a dynamic sub-model could be replaced by one without dynamics. Methods for handling discontinuities which are associated with variable-step integration methods clearly cannot be used and other approaches have been suggested (see e.g. [2] and [3]).

In addition to the choice of integration method and time step, questions can arise about the *communication interval* appropriate for a real-time problem. Clearly, for real-time operation, *synchronisation* to an external clock is essential. If the simulation can be run faster than real time on the chosen computer, a delay can be introduced until each clock interrupt occurs. The communication interval may represent the update interval for *external data* being input through analogue-to-digital converters or output to external hardware through digital-to-analogue converters, or being output for graphical display. This communication interval is, usually, considerably larger than the integration step length. For hardware-in-the-loop situations, external data thus changes value only at the communication times and the choice of communication interval must depend on the dynamic characteristics of the external hardware elements. In some applications, such as a flight simulator, there may be a number of different (but related) communication intervals, since different parts of the external hardware may involve different dynamic characteristics. One example of such *multi-rate* simulation methods in a real-time application may be found in the generation of force feedback for the pilot's controls to provide tactile information about the response of a simulated aircraft to the pilot's commands [8.4]. These channels involve smaller communication intervals than those applied elsewhere in the simulator because the human tactile system is sensitive to high frequencies and use of inappropriate communication intervals would be immediately apparent to an experienced pilot through the absence of the expected sensory feedback.

Real-time simulation applications date back to times before digital computers had the necessary computational power. *Analogue computer* technology (see e.g. [1] and [5]) was widely used for real-time problems during the period from the 1950s until the late 1970s. However, as the performance of digital computers improved, the inherent parallelism of analogue computers was combined with the general-purpose computational power of digital processors in a new type of simulation facility known as a *hybrid computer* (see e.g. [1] and [6]). Although relatively expensive, hybrid computers were capable of providing highly effective solutions for complex real-time problems and were applied successfully in a number of areas, especially in aerospace and defence, until the 1980s. An interesting recent development has been a VLSI analogue computer/ math co-processor by Cowan and his colleagues at Columbia University [7]. This is capable of handling nonlinear ordinary differential equations of up to 80th order and occupies an area of integrated circuit of 1 cm^2. It has an interface which provides facilities for automated programming and calibration, and a demonstration version of this modern analogue computer, implemented in 0.25 μm CMOS, has been shown to provide solutions of differential equations up to 400 times faster than a modern workstation running MATLAB® while dissipating only 300 mW [8.7]. Whether or not innovations of this kind will lead to a new generation of commercial analogue/hybrid computers remains to be seen, but this development is certainly of potential interest for specialised real-time applications involving large simulation models.

In tackling real-time simulation problems using methods involving general-purpose digital computers, one must understand the trade-offs between computer hardware performance, software capabilities and model accuracy. For most real-time applications the communication interval and the integration step size should be as small as possible, since the discretisation inherent in digital simulation introduces an effective time delay which can have a destabilising effect. Another factor to be considered in the choice of integration step size and communication interval is the time required for other numerical processes, such as *multivariable function generation*, which may be as significant as the time for numerical integration. In such situations it is pointless to try to improve the integration accuracy unless these other issues are fully taken into account.

Error measures involving quantitative comparisons between model behaviour and the real system are more important in real-time situations than the error analysis and convergence tests applied for other simulation applications. For example, it is important in real-time simulation to ensure that, for a given operating condition, the eigenvalues of the simulation model closest to the imaginary axis in the complex plane are represented accurately. It is also important to check that spurious high-frequency oscillations do not appear which have no counterpart in the real system.

The possibility of linking simulations to rapid prototyping equipment means that new designs in fields such as mechatronics, robotics and automatic control can now be tested on hardware in minutes where, previously, equivalent tasks could have taken weeks. In such specialist applications, where much use is also made of *embedded processors*, commercial systems such as the MathWorks Simulink Coder™ (formerly the Real-Time Workshop™) [8] and the National Instruments (NI) LabVIEW Real-Time™ system [9], allow generation of C or C++ code directly from simulation diagrams. This avoids the time-consuming and error-prone process of converting algorithms from languages such as MATLAB® into C code by hand for real-time implementations. Similar facilities are available for other simulation software systems, including the 20-sim modelling and simulation tool developed by ControlLab Product BV in the Netherlands [10]. Another widely used system for embedded system development and real-time simulation is available from dSPACE Inc [11] and products are also available from other vendors which offer broadly similar solutions.

In addition to the plant which is being controlled, embedded systems involve software elements and hardware elements that include both computer hardware and input/output interfacing hardware. Modelling of the complete system must involve modelling of all three aspects and, ideally, it should be possible not only to simulate each part of the system independently of the others, but also to simulate the computer and interface hardware together with the software and the computer and interface hardware along with the plant. As a final step it should be possible to investigate all the hardware and software elements of the system when operating together. Broenink and his colleagues from the University of Twente in the Netherlands advocate a building-block approach for embedded system design, along with an object-oriented approach to modelling [12]. The specific object-oriented methods proposed involve bond-graph modelling for the system to be controlled, VHDL [13] for the input/output hardware and algebraic techniques for describing the embedded software in terms of communicating processes through the application of the CSP (communicating sequential processes) approach of Hoare [14].

Systems based on general-purpose *digital signal processing (DSP) boards* and digital *field programmable gate array (FPGA) hardware* are providing new possibilities in terms of low-cost interfacing and control, and have considerable potential for new developments in hardware-in-the-loop simulation [15]. The account by Jovanovie *et al.* [16] of an inexpensive prototyping system for mechatronic systems provides an interesting example where the 20-sim modelling and simulation tool (with C-code generation functionality) was used to support system design for an electro-mechanical servo system. An example of FPGA-based real-time simulation may be found in the 2008 paper by Crosbie [15], where use was made of an RT-LAB™ system from Opal-RT [17] for a real-time simulation of a power-electronic system. The RT-LAB™ system provided two dual-core Opteron™ processors operating under the Red Hawk real-time Linux operating system and a FPGA which can provide interfacing to external hardware or perform high-speed simulation of part of the complete system.

In Crosbie's application [15], the FPGA simulates part of a power-electronic system involving two three-phase converters connected by a DC link with simple AC generator/load units at each end of the network. The simulation, which provides a benchmark for evaluating different simulation approaches, can thus represent an AC generator feeding a converter that provides three-phase rectification to produce direct current, which is then supplied to the second convertor for conversion back to alternating form for an AC load. Filters are included on the DC side of each converter. Each converter has six switches operated by timing pulses from controllers involving pulse-width modulated (PWM) control for voltage, current and power flow. The model involves a total of 23 first-order ordinary differential equations in addition to switching logic and other features that further increase the computational demands. In the RT-LAB implementation, one converter was simulated on the FPGA while the rest of the model was implemented on the dual Opteron™ processors. With an appropriate choice of FPGA, simulation frame times as small as 400 ns have been achieved [15]. This suggests that, for applications such as power electronics, the FPGA approach allows simulation of quite complex sub-models on a single chip. The FPGA approach also has significant advantages in terms of cost compared with DSP boards, although the programming task for simulation applications can be significant.

8.3 Hardware-in-the-loop simulation

A good example of hardware-in-the-loop simulation is an aircraft flight simulator (see e.g. [4]) where elements of the cockpit, such as the pilot's controls, may be the same as hardware in the real aircraft. Similarly, training simulators for chemical process plant or electrical power facilities may involve control room displays or other hardware used in the real plant. Often a training simulator is built before the corresponding real plant is completed and the simulator may then be very helpful for the plant commissioning process as well as for training operators.

Hardware-in-the-loop techniques can provide a form of rapid prototyping in which we start with a virtual prototype and move in stages towards a real prototype system involving the hardware and software of the final design. For example, a controller for a chemical process plant could be tested initially by coupling the controller hardware to a real-time simulation of the plant based entirely on software. Once testing of the controller showed that the performance was satisfactory when used with the simulator, the same controller hardware could be tested further by coupling it to the real plant. The benefits of this incremental 'divide and conquer' type of approach include the fact that the performance can be investigated safely for a wide range of normal and fault conditions.

One point, often overlooked, is that differences exist between training simulators and real-time simulators designed to support engineering activities. Although the problems being tackled may have similarities, the behaviour of a training simulator need match that of the real system only at the operator interface and the emphasis is therefore on *emulation*. Thus, some components of the simulation may be input-output descriptions representing reduced-order models, fitted in a purely mathematical fashion, to more complex underlying descriptions. Sub-models may thus have little or no physical basis, but may be entirely suitable for the training application, provided they are used within the limits for which they have acceptable accuracy. Development of a training simulator may therefore be regarded as a top-down process in which details from the real system are added until the simulator's performance is adequate. This is usually inappropriate for a real-time simulation intended for engineering system development and rapid-prototyping applications. In that case, the role of the real-time simulation is usually to test a proposed system up to, or even beyond, the normal design limits and it is important that as much as possible of the model is developed on a physical basis in order to enhance understanding of the real system and the model. One example of a situation in which a physically based simulation model is important is in testing for fault conditions and fault recovery strategies. The intentional introduction of faults in the testing of the real plant is often unacceptable for reasons of safety or plant integrity, and the use of real-time simulation for such investigations becomes essential.

8.3.1 Case study: hardware-in-the-loop simulation for development of a speed-control system for a hydro-turbine and generator

Although not an example of concurrent design, the development and commissioning of the fast-acting speed governor system for the hydro-electric generator system discussed in Section 7.5.2 provides an illustration of partial-system testing and hardware-in-the-loop simulation. This study involved the redesign of a control system for an existing plant to meet an entirely new set of performance requirements and it was a truly multidisciplinary undertaking involving control systems specialists, mechanical engineers and electrical power systems engineers.

The project involved development of a real-time simulation for one generating unit at the Sloy Power Station operated, at that time, by the North of Scotland Hydro Electric Board. The work formed part of a broader investigation concerning the possible replacement of existing mechanical/hydraulic speed governors for hydro-turbines in power stations in Scotland by faster-acting electronic governors. The requirement for faster governors was associated with a changing role for hydro-electric power generation in Scotland as it moved away from base-load supply to provide standby generation capacity that could provide sources of additional power to compensate very rapidly for failures of large nuclear, coal or gas-fired stations elsewhere in the distribution network. The electronic control hardware developed for the project had to be tested initially in conjunction with an accurate real-time simulation of the hydraulic, mechanical and electrical components of the system before testing of the controller on site.

Initial site tests on the complete hydro-turbine and generator system with the existing governor hardware provided results that were used in developing the model, which included water pipeline dynamics (as discussed in Section 7.5.2), a nonlinear representation of the hydraulic and mechanical elements of the turbine, a detailed representation of the generator and simplified models of the associated electrical system for various different network configurations. New controller hardware was coupled to the real-time simulation through an interface that involved all the continuous and logical signals that would appear in the interface between that controller and the real plant.

The real-time simulation was developed initially using a hybrid computer [18], but was subsequently implemented as a digital simulation. It allowed investigation of a number of different situations on the system, including start-up, normal steady-state operation for different electrical loading conditions, transients following sudden electrical load changes, fault situations and system shutdown. The quality of the real-time simulation model was assessed initially by making comparisons with non-real-time models and by comparisons with data from initial site tests.

Once the electronic controller within the hardware-in-the-loop simulation of the combined turbine, generator and control system was judged to be operating appropriately (as outlined in Section 7.5.2), it was moved on site for commissioning tests. On-site testing initially involved relatively benign cases that had been investigated previously using the real-time simulation and then moved, in an incremental fashion, to include situations requiring faster governor action. The project led to a prolonged programme of work involving evaluation, on site, of fast-acting analogue governors [19] and also microprocessor-based digital governor systems ([20] and [21]).

8.4 Multi-rate simulation techniques

Many models involve sub-systems that have a wide range of time constants. Since variable step algorithms are clearly inappropriate for real-time applications, one approach in such cases involves grouping the sub-models according to dynamic properties so that different integration steps can be used in different sub-models. The total number of calculations is then smaller and the simulation is potentially faster. This is known as *multiple frame rate* or *multi-rate simulation*. Although this reduces computational demands, it also raises issues in terms of the overall accuracy and stability of the simulation (see e.g. [15]).

8.4.1 Fundamentals of multi-rate simulation

With fixed-step integration methods, an integration step length of h seconds gives a frame rate of $f = \frac{1}{h}$ frames per second. A simple multi-rate situation is shown in Figure 8.1 where there are two integration step lengths, h_1 and h_2, where is h_2 related to h_1 through an integer N according to the equation:

$$h_2 = Nh_1 \tag{8.1}$$

Thus Segment 1 of the model, as illustrated in Figure 8.1, gives results at a frame rate of:

$$R_1 = \frac{1}{h_1} \tag{8.2}$$

while Segment 2 gives results at the slower rate of:

$$R_2 = \frac{1}{h_2} = \frac{1}{N} R_1 \tag{8.3}$$

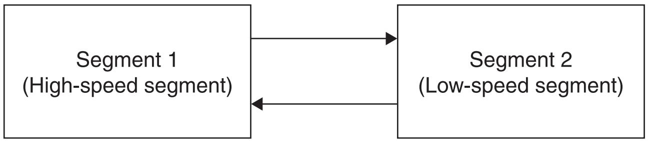


Figure 8.1 **Block diagram for a simulation involving two segments having different frame times. The slow frame time is an integer multiple of the faster frame time and communication takes place at the slower frame rate**

Communication between segments can be handled in several ways. For the case shown in Figure 8.1, the simplest is based on a zero-order hold for transfer from Segment 2 to Segment 1. Segment 1 thus uses the last value received from Segment 2 for *N* steps until the value from Segment 2 is updated. Outputs from Segment 1 being communicated to Segment 2 must be averaged, possibly by simple filtering, over *N* steps before being transferred. Large differences of step sizes may require use of anti-aliasing filters.

Although introduced here using only two segments and two frame rates, the principles of multi-rate simulation can be applied to situations involving more segments and more frame rates. The ratios of these different frame rates are normally integer quantities.

8.4.2 Case study: multi-rate simulation of an unmanned underwater vehicle

Electrical drive systems for surface ships and underwater vehicles involve sub-systems having a wide range of time constants. The complete model may involve mechanical, thermodynamic, hydrodynamic, electrical, electronic and software elements. Time constants for sub-systems involving electronic components may well be many orders of magnitude smaller than the dominant time constants of the six-degrees-of-freedom model of the vehicle involving the hull, propellers and control surfaces. Simulation of the complete system using a single program requires integration step sizes consistent with the smallest time constants, leading to a very 'stiff' simulation problem and, probably, very long solution times, as discussed in Chapter 3 and Section 8.2. This situation is clearly one for which multi-rate simulation offers possible advantages [22].

Development of a simulation for an unmanned underwater vehicle (UUV) capable of running in real time (or in a timescale faster than real-time for multi-run optimisation studies) has been the subject of a recent study ([23] and [24]) and provides an illustration of multi-rate simulation techniques. The vehicle is the one described in Appendix A1 and the specification requires that the model should be capable of representing the power electronic sub-system for events involving time intervals of less than 10 µs, over time periods of minutes or longer while running in real time or in even faster timescales. The system divides naturally into sub-systems with several ranges of frame time. Figure 8.2 is a schematic diagram of the complete underwater vehicle system showing the interactions between these different sub-systems.

In Figure 8.2, the model is split into five blocks involving four different frame rates. The DC to AC converter model involves the fastest frame rate, the feedback controller is a slow-medium speed component, the electric motor model is a fast-medium component and the battery, the vehicle and its control surfaces are all grouped together and have the slowest frame rate. The graphics interface, which provides an animated display showing the vehicle motion in three dimensions, also involves the slowest frame rate. The specific

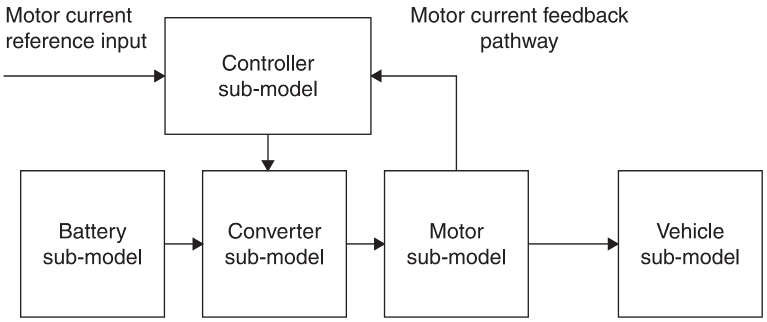


Figure 8.2 **Block diagram representation of UUV model with the electrical drive system**

frame times used for these four areas involve integration periods of 2μs for the converter, 100 μs for the motor, 800 μs for the feedback controller and 100 ms for the battery, vessel and graphical output.

The simulation has been implemented using the Virtual Test Bed (VTB) environment (discussed briefly in Chapter 3), together with the associated VXE graphics software for displaying graphical output and 3-D animations. Both of these software tools were developed at the University of South Carolina [25]. The VTB provides a flexible simulation environment which allows sub-system models developed initially using different simulation tools to be combined. The VTB software also has a model library which contains many mechanical and electrical component models.

Funding for an investigation of multi-rate simulation techniques, including the development of the multi-rate simulation of the UUV, was provided by the U.S. Office of Naval Research (ONR) through grants to California State University, Chico, and the University of Glasgow, and different programming approaches were used in developing the sub-system models because these were developed by different groups in different locations. For example, the electrical and electronic sub-models were the responsibility of a research group at California State University, Chico, and were programmed using C++ code while adaptation of an established MATLAB® six degrees-of-freedom vehicle model to suit this application was undertaken by researchers at the University of Glasgow in the United Kingdom. The DC power source was represented using native VTB battery models.

Although the converter model was coded using C++, it was implemented as a native VTB simulation model and in this sub-model the DC input connection and the AC output connections are natural couplings (as outlined in Chapter 3), while the connections to the controller are signal couplings. The converter involves Euler integration for the input DC filter capacitor and trapezoidal integration for the filter components at the output. The sine and triangular waveforms of the controller are found by a table look-up method. The motor simulation was implemented in C++ as a native VTB simulation using trapezoidal integration. The six-degrees-of-freedom simulation model of the vehicle involved a fixed-step fourth-order Runge-Kutta integration algorithm and the original MATLAB® simulation was translated to C++ and implemented in the multi-rate simulation as a native VTB model. Fin deflection and propeller shaft inputs are applied by the user.

A multi-rate solver was developed initially by bringing together the models to be run at different rates within a single VTB ‘super-model’. The VTB would then run at the rate of the slowest model and the individual model step sizes are integral divisors of the VTB time step. An internal scheduler calls the internal models at the correct times within each VTB time step and returns values to the VTB at the start of the next VTB step. Results can be displayed at the user interface at each VTB time step.

Execution in a timescale faster than real-time, with the VXE 3-D graphical output, has been achieved on a typical laptop computer. The European Simulation Language (ESL) [26] was used to provide comparisons with conventional simulation results. ESL, which was developed at the University of Salford for the European Space Agency, supports multi-rate simulations. One of the important features of ESL, which has recently been integrated into the VTB [27], is that it can automatically locate switching points, accurately integrate up to the discontinuity and then continue from that exact point. The effects of taking fixed-time steps in a fast multi-rate simulation can thus be investigated and errors associated with different step lengths can be assessed.

8.5 Some new developments in real-time simulation

Steady progress continues to be made in the computing power provided within conventional personal computers, workstations and specialised digital signal processors and FPGAs. These developments have a direct and positive effect on the real-time computing field in general and on simulation in particular. Crosbie [15] has also suggested that the IBM Cell Architecture, which involves one PowerPC type of processor coupled to eight synergistic processing elements and a very high-speed bus, might well be appropriate for simulation applications. A version of this processor is used within the Sony Playstation 3® and has been shown to have significant speed advantages. Other developments in terms of specialised hardware, such as developments based on FPGAs and the VLSI analogue computer developed at

Columbia University, may also have an effect on real-time simulation practice in the longer term.

While developments in terms of hardware are important, other developments in terms of simulation methods and simulation software are also significant. The current interest in multi-rate simulation is a good example and there are areas where additional research effort may well produce benefits. One example of this could be systematic investigation of the influence on the overall stability of a simulation model of the methods used to provide communication of data between modules operating at different frame rates. Crosbie [8.15] has also pointed out that *quantised state system (QSS) methods* ([28] and [29]) could be useful for high-speed real-time simulation applications and that further research on this might yield dividends. In that approach the process advances until one state variable reaches the next quantisation level instead of using the conventional idea of a time frame. Advantages are claimed for this quantised approach in terms of accuracy and stability, but few applications have been reported so far.

8.6 References

[1] Murray-Smith, D.J. (1995) *Continuous System Simulation*, Chapman and Hall, London, UK.

[2] Gear, C.W. (1977) ‘Simulation: conflicts between real-time and software’, in Rice, J.R. (ed.), *Mathematical Software III, Proceedings of the Symposium at the University of Wisconsin, 28–30 March*, Academic Press, New York NY, USA, pp. 121–38.

[3] Howe, R.M., Ye, X.A. and Li, B.H. (1984) ‘An improved method for simulation of dynamic systems with discontinuous nonlinearities’, *Transactions of the*

Society for Computer Simulation, Vol. 1, No. 1, pp. 33–47.

[4] Rolfe, J.M. and Staples, K.J. (eds) (1986) *Flight Simulation*, Cambridge University Press, Cambridge, UK.

[5] Ricci, F.J. (1972) *Analog-Logic Computer Programming and Simulation*, Spartan Books, New York NY, USA.

[6] Matko, D., Karba, R. and Zupanćić, B. (1992) *Simulation and Modelling of Continuous Systems*, Prentice Hall, London, UK.

[7] Cowan, G.E.R., Melville, R.C. and Tsividis, Y.P. (2006) 'A VLSI Analog Computer/Digital Computer Accelerator', *IEEE J. Solid–State Circuits*, Vol. 41, No. 1, pp. 42–53.

[8] Mathworks/Simulink Coder™ (online): *www.mathworks.com/products* (accessed 30 June 2011).

[9] LabVIEW Real-Time™, National Instruments Corp, Austin, USA (online): *www.ni.com/realtime* (accessed 30 June 2011).

[10] 20-sim, modelling and simulation package, ControlLab Products BV, Enschede, the Netherlands (online): *www.20sim.com* (accessed 30 June 2011).

[11] dSPACE Prototyping Systems, dSPACE Inc, Novi, USA (online): *www.dspaceinc.com* (accessed 30 June 2011).

[12] Broenink, J.F., Groothuis, M.A., Visser, P.M. and Orlic, B. (2007) 'A model-driven approach to embedded control system implementation', in *Proceedings of the 2007 International Conference on High Level Simulation Languages and Applications (HLSLA '07), part of the 2007 Western Multiconference on Modeling and Simulation (WMC '07), 14–18 January 2007, San Diego, California, USA*, the Society for Modeling and Simulation International, San Diego CA, USA, pp. 137–44.

[13] Yalamanchili, S. (2000) *Introductory VHDL: From Simulation to Synthesis*, Prentice Hall, Upper Saddle River NJ, USA.

[14] Hoare, C.A.R. (1985) *Communicating Sequential Processes*, Prentice Hall, London, UK.

[15] Crosbie, R.E. (2008) 'Advances in high-speed real-time simulation', in *Proceedings of the Second UKSIM European Symposium on Computer Modelling and Simulation*.

[16] Jovanovic, D., Orlic, B., Broenink, J. and van Amerongen, J. (2003) 'Inexpensive prototyping for mechatronic systems', in *Proceedings of the 5th Workshop on European Scientific and Industrial Collaboration (WESIC 2003), 28–30 May 2003, Miskolc, Hungary*, Vol. II, pp. 431–8, University of Miskolc, Hungary (ISBN 963-661-570-5).

[17] RT-LAB™, Opal-RT Technologies Inc, Canada (online): *www.opal-rt.com/product/rt-lab-professional* (accessed 30 June 2011).

[18] Bryce, G.W., Foord, T.R, Murray-Smith, D.J. and Agnew, P.W. (1976) 'Hybrid simulation of water-turbine governors', in Crosbie, R.E. and Hay, J.L. (eds), *Simulation Councils Proceedings Series*, Vol. 6, No. 1, pp. 35–44, Simulation Councils Inc, La Jolla CA, USA.

[19] Bryce, G.B., Agnew, P.W., Foord, T.R., Winning, D.J. and Marshall, A.G. (1977) 'On-site investigation of electrohydraulic governors for water turbines', *Proceedings IEE*, Vol. 124, No. 2, pp. 147–53.

[20] Winning, D.J., Marshall, A.G., Findlay, D.G.E., Aitken, K.H. and Grant, N.F. (1980) 'Controller testing facility on 32.5MW water turbine', *Proceedings IEE*, Vol. 127, Part C, No. 6, pp. 357–9.

[21] Findlay, D., Davie, H., Foord, T.R., Marshall, A.G. and Winning, D.J. (1980) 'Microprocessor-based adaptive

water-turbine governor', *Proceedings IEE*, Vol. 127, Part C, No. 6, pp. 360–9.

[22] Word, D., Bednar, R., Zenor, J.J. and Hingorani, N.G. (2008) 'High-speed real-time simulation for power electronic systems', *Simulation*, Vol. 84, pp. 441–56.

[23] Zenor, J.J., Bednar, R., Word, D., Hingorani, N.G. and McGookin, E. (2007) 'Simulation of an unmanned underwater vehicle (UUV): a multi-rate simulation', *Proceedings of the Summer Computer Simulation Conference 2007 (SCSC 2007)*, San Diego, CA, July 2007, pp. 204–8, Society for Computer Simulation International, San Diego CA, USA.

[24] Zenor, J.J., Murray-Smith, D.J., McGookin, E. W. and Crosbie, R.E. (2009) 'Development of a multi-rate simulation model of an unmanned underwater vehicle for real-time application', in Troch, I. and Breitenecker, F. (eds), *Proceedings MATHMOD 09, Vienna, 9–13 February 2009*, pp. 1951–7, Argesim/Asim, Vienna, Austria.

[25] VTB, modelling and simulation package, the University of South Carolina, *The Virtual Test Bed* (online): *http://vtb.engr.sc.edu/vtbwebsite* (accessed 30 June 2011).

[26] ESL, modelling and simulation package, ISIM International Simulation Ltd (online): *www.isimsimulation.com* (accessed 30 June 2011).

[27] Pearce, J.G (2007) 'Interfacing the ESL simulation language to the Virtual Test Bed', in *Proceedings of the 2007 Western Simulation Multiconference, San Diego, CA, January 2007*, Society for Computer Simulation International, San Diego CA, USA.

[28] Kofman, E. and Junco, S. (2001) 'Quantised state systems: a DEVS approach for continuous system

simulation', *Transactions of Society for Computer Simulation*, Vol. 18, No. 3, pp. 123–32.

[29] Beltrame, T. and Cellier, F.E. (2006) 'Quantised state system simulation in Dymola/Modelica using the DEVS formalism', in *Proceedings of the 5th International Modelica Conference, 4–5 September 2006, Vienna, Austria*, Vol. 1, pp. 73–82, the Modelica Association, Linköping, Sweden (online): *www.modelica.org/events/modelica2006/Proceedings/proceedings/Proceedings2006_Vol1.pdf* (accessed 30 June 2011).

9

Model management

Abstract: The model-driven concurrent approach to the design and development of engineering systems allows early detection and correction of design errors, and offers new opportunities for optimisation at the level of the complete system. Tools for the efficient development and testing of physically based dynamic models are of central importance for this and include forward and inverse simulation, parameter sensitivity analysis, system identification, parameter estimation, optimisation and partial system testing through hardware-in-the-loop simulation. Fitness for purpose, quality, reusability and the integration of simulation methods with other design and analysis tools are vitally important. Libraries of models and models that are generic in structure are also important. Dealing successfully with all of these aspects of modelling and simulation requires the application of good management principles and the development of good documentation and appropriate updating procedures at all stages in a project. It is argued that the education and training of most engineers does not at present give sufficient emphasis to many of these broader issues that are of vital importance for the successful application of simulation methods and model-driven design.

Key words: documentation, tool integration, model reuse, sub-model, model library, model sharing, generic model, internal verification, external validation, education.

9.1 Issues of model management

As pointed out in Chapters 1 and 2, systems involving closely integrated elements from a number of different engineering disciplines have become increasingly important in recent years. Integrated systems applications in aircraft, in the automotive industry, in electrical power generation and distribution, in robotics and in chemical and pharmaceutical process engineering are becoming commonplace. In all of these and many other areas, a multidisciplinary and model-driven concurrent approach to design is also taking over from traditional sequential design procedures and this is now recognised as important for successful integrated system design. Benefits from the model-driven concurrent approach, if applied correctly, include earlier detection and correction of design flaws, and new possibilities for system-level optimisation.

Because of the central role of modelling and simulation techniques in integrated system design, a strategy is needed to ensure that these methods are applied properly and more effort is needed in this area if we are to be successful in meeting design requirements while also reducing development times and costs. Unfortunately, this is not yet true in general. Current practice in system modelling and simulation within many organisations is often lacking in terms of systematic processes, in surprising contrast with accepted procedures within the more general software engineering field where more rigorous testing, documentation and version control are an integral part of project management.

Modelling objectives in different areas can differ greatly and prior knowledge of the real system and understanding of design requirements are both important. A full statement of how the model is to be used is particularly important when an integrated system design and development project involves a number of different teams, perhaps drawn from different engineering disciplines. The purpose of a model influences the type of model needed and, if the goal is to provide further insight about the corresponding real system, the required form of the model may differ from many models conventionally used for quantitative prediction, simulation or system design. Physically based forms of model may be particularly helpful in a multidisciplinary design environment where engineers have widely different backgrounds and where models and computer simulations can provide a natural means of communication about technical issues.

One feature of present-day research on simulation is an increased emphasis on the broader aspects of the model development process (such as bond-graph and other energy-based modelling procedures), enhanced computing environments (especially in terms of the user interface), libraries of sub-models and systematic processes for assessing, correcting and documenting models. More effort must be directed towards further developing and maintaining libraries of validated simulation models and commonly used sub-models. This is important if we are to exploit fully the benefits of model reuse and the development of reliable generic models, and is a significant part of the move towards developing more efficient model-driven design techniques and more effective methods for model management.

Inverse systems receive considerable attention in this book and inverse simulation methods, developed initially for use in handling-qualities studies for fixed-wing aircraft and helicopters, have been shown to be of value in modelling and simulation of complex systems of a more general kind. Different physical insight may result from examining the input needed to allow a specific form of output to be achieved and this is especially significant in areas such as actuator design. It is believed that combining forward and inverse simulation processes within a modelling and simulation strategy can be potentially very valuable.

Confidence in a prediction is a function of the confidence demonstrated in sub-system models as well as in the complete model. This is particularly important where sub-system models can be tested experimentally. Exhaustive testing of sub-models allows confidence to be established at that level first and then extended gradually to less well-defined situations involving testing of the complete system model over a range of experimental conditions. The main aim in using modelling, simulation and prototyping techniques in engineering design and development is to make sure that when the system is built, tested and put into service, there are no surprises. This aim is seldom completely satisfied, but the more comprehensively that models are checked and tested, the more likely it is that the resulting system will be acceptable in service from an early stage.

In the development of entirely new systems, experimental data from the complete system cannot be available at the design stage. In some cases, historical data from earlier systems of a similar kind can be helpful in evaluating the model of the new system. Successful application of this approach depends on good documentation of models of the earlier systems and of the experimental data and tests used to evaluate those previous models.

If simulation and modelling methods are applied in a highly focused fashion, with the right questions in mind, they can help to produce new insight that would be very difficult to obtain in other ways. For example, as pointed out in Chapter 7 in the context of reduced-order models, developments taking place in the field of micro- and nano-electromechanical systems (MEMS and NEMS) are throwing up many interesting and challenging problems relating to modelling and simulation. It must also be recognised that those working in that field are also making important contributions in terms of methodological developments, many of which have significance for all involved in modelling and simulation of integrated systems. In addition, MEMS and NEMS developments often involve problems of multi-scale physically based modelling that have much relevance for engineers in other application areas.

9.2 Tools for model management

Developers of computer-based tools for modelling and simulation must strive to achieve a good balance between efficiency of numerical solutions and user-friendly software tools for model construction, testing, external validation, documentation and maintenance. Good planning of systems for documentation and the management of models is especially important when models may be reused, and well-maintained libraries of reusable models can assist greatly in the development of models for new applications. Good tools for the management of models also allow the user to focus attention on issues of model quality and testing. The potential for integration of simulation models with other design and analysis tools is another important area. Efficient, user-friendly and reliable version handling for models is essential and, as pointed out by Brade [1], this must form part of a more general stepwise procedure for model development in which verification, validation and documentation are of central importance.

9.2.1 Documentation and reuse of models

Models are often developed for engineering applications on a one-off basis for a specific task and new projects, often very similar to earlier ones, frequently involve completely new models. This is not only costly and time-consuming, but it is also true that models developed in this way are not always subjected to rigorous validation and documentation processes.

A poorly documented model of questionable validity is seldom helpful, whereas a model of proven fitness, together with good documentation, can provide an excellent starting point for a new application. This applies even if an established model is to be used in a new way since, although the model may need changes for the new application, examination of a proven model may still be the best starting point. There must, however, be information about how the earlier model was developed and used, about its range of validity, and about the underlying assumptions and constraints.

Items recorded about a model or sub-model should include:

1. the purpose of the model and the intended application;
2. assumptions used in developing the model and any constraints that may result;
3. details of tests on the real system carried out for model development, including model structure estimation and parameter estimation;
4. the computer simulation code for the model, if appropriate;
5. details of internal verification checks carried out to ensure that the computer-based representation matches the mathematical description; and
6. details of external validation processes for the complete simulation model, along with the reasons for accepting or rejecting the model, together with statements about the range of applicability of each accepted model.

As pointed out in Chapter 7, model development does not end when a model or sub-model is accepted for a specific application or for inclusion in a library of models following successful internal verification and external validation. Model development continues throughout the life of the real engineering system or product. Indeed, understanding of the limitations of a given model should grow throughout the application phase of a project. The process is not complete when the decision is made to accept a model for a particular application, and model management systems and documentation procedures must allow for this. *Regressive testing* of models within the iterative process of model development is as important as regressive testing of any other software (see e.g. [2]) and model documentation systems must allow for possible changes and updating throughout the life cycle of the system represented by the model.

Model documentation must take account of the needs of those encountering that specific model or sub-model for the first time. Diagrams are needed and these must be consistent with documentation of the corresponding real system. Brade [1], as well as emphasising the need for more meaningful documentation and criticising the present lack of quality assurance as an integral part of the model development process, discusses the potential and current limits of existing guidelines, such as the Verification, Validation and Accreditation Recommended Practices Guide of the US Defense Modeling and Simulation Office [3]. He believes that a structured approach to the collection of information and documentation during the model development process can provide a stable foundation for verification and validation and opportunities for effective and efficient reuse of models. These ideas are closely linked to the concept of the 'Verification and Validation (V&V) Triangle' which Brade has presented as a central part of his doctoral thesis [1]. The V&V Triangle provides a foundation for planning and implementing all aspects of verification and validation, and provides an overview of desirable verification and validation objectives and possible techniques that can be used to achieve those objectives. This methodology also allows features of an executable model to be traced back systematically to the model requirements specification.

Many tools for modelling and simulation provide only basic facilities for the processes of model entry and the running of simulations. Documentation within models developed from these tools usually consists of comment lines within code or annotations in graphical models. Recently developed simulation and modelling tools put more emphasis on model management issues and allow access to published libraries of sub-models and the creation of new libraries. *Object-oriented methods* are relevant for this and object-oriented software environments may offer some advantages for the development of reusable and readily extendable models.

Brade [1] suggests that, for documentation to be fully effective at reasonable cost, software is needed that automatically records changes made at each stage of model development since it has to be accepted that, if attempted manually, documentation of the model development process is often ineffective, usually incomplete and frequently full of errors.

The *assessment of overall model credibility* remains a problem even after the successful application of internal verification and external validation methods, and this presents problems for documentation of overall model quality and the model testing procedures. V&V methods do not *prove* that a model and the corresponding simulation results are suitable for the intended application; they merely suggest that one can apply the model until new evidence is found that indicates a further problem. Extensive statistical data evaluation of simulation model output may be helpful in identifying problems that were not detected in earlier external validation processes and, in future, more highly automated methods of error detection may prove to be helpful. Brade and Waldner have described a tool of this kind for automatic detection of violations of desired behaviour within a model [4]. Although developments that may lead to a more automated approach to external validation are of considerable interest, the current view is that automated methods of analysis cannot entirely replace human review and 'face' validation techniques, in which experts familiar with the behaviour of the real system carry out more subtle and intuitive testing procedures on the simulation model. Adequate and detailed documentation of the results of these manual testing procedures is, of course, of vital importance.

9.2.2 Model libraries and their organisation

A *library of sub-models* for a specific field of application must be designed not only to meet current requirements but also possible future requirements [5]. A collection of sub-models should be built up, each of which can be tested separately for a range of conditions, documented and made available for wider use. Thus sub-models are best designed, from the outset, as building blocks for a family of applications rather than for a single project with internal verification and external validation processes applied, initially, at the sub-model level.

Establishing a taxonomy of models within a library becomes important if the number of models is large [6].

This commonly involves generic classes and specific sub-classes of models, and the chosen structures should also, ideally, allow movement between energy domains. This could, for example, allow one to change easily from consideration of electrical motors to hydraulic motors in a design application.

Model reuse can reduce the development time of new models and a model library may allow engineers to choose sub-models from a number of possible representations involving different levels of detail. Those who use a library for development of a new model must be offered a limited number of reasonable alternatives for the representation of a particular element so that they can make an informed decision about the sub-model that best meets their needs [7]. Any library of models must therefore be fully supported by adequate documentation in order to allow full confidence in a modelling process based on model reuse to be established.

As with other aspects of model documentation, the information about a library sub-model should include the theory used in the development of the model (with sources referenced properly), the assumptions made, testing and external validation procedures applied, and information about the range of applicability. Although models are always developed with a particular application in mind and therefore involve some subjective elements, they can be made potentially useful for other applications if assumptions and limitations are clearly stated. This information must all be accessible to others. Also, awareness of model limitations inevitably fades with time and good documentation is essential for the developer, even if others are not immediately involved.

There are also broader issues of software design that can facilitate reuse of sub-models. One example is the extent to which the object-oriented approach of general purpose programming languages can be applied in specialised packages for modelling and simulation. Also, some simulation software such as Modelica® [8] and associated packages such as Dymola allow non-causal modelling and facilitate the creation of models in declarative style. This may allow an interface to be defined for a sub-model in terms of a pair of variables which are not specifically associated with an input or an output.

In addition to environments, such as Modelica® [8], which provide standard model libraries and facilities for developing new libraries, other widely used software packages can be extended with tools for physical modelling in various domains. One example is the Simscape™ package [9], which provides the MATLAB®/Simulink® [10] user with access to powerful facilities for the integrated modelling of systems involving a number of physical domains. Simulink® itself includes some standard library sub-models and through Simscape™ there are additional standard libraries involving sub-models for components in specialised fields, including mechanics, electronics and hydraulics. There are also libraries for mechanical transmission systems and for electrical power systems. Using the Simscape™ language, which is based on MATLAB®, sub-models can be created together with equivalent Simulink® blocks for new physical components that do not appear in the standard libraries. Similarly, it is relatively straightforward to create entirely new libraries using the facilities of Simscape™ and to extend existing libraries so that specialist models can be deployed across an organisation or made available to subcontractors in large projects. Signals and parameters can have units within the models in Simscape™ libraries and there are also facilities for the automatic conversion of units which can be important for multidisciplinary projects involving several design teams. Additional facilities that can provide links with other widely used design tools may be included. For example, in the case of the mechanics library (SimMechanics™), the facilities include translators for SolidWorks®, Autodesk®Inventor® and Creo Parametric® (formerly ProENGINEER®) to allow use of well-established CAD tools in the definition of models.

In using a model library, the model developer has to keep the intended application firmly in mind. A model is assembled in an iterative way using the three-layered structure discussed in Chapter 3 and selecting appropriate sub-models at each layer. The layered approach also has significance in terms of model documentation and Breunese *et al.* [6] describe an approach that could allow more useful forms of documentation to be developed. Those authors advocate an approach based on the layered type of model structure outlined in Chapter 3, involving a top layer in which the system is considered as a set of physical components or real-world objects, a second layer in which the system is viewed as a set of physical concepts and the third level that depends on mathematical relationships. These three layers can be seen as representing three different viewpoints that are all important in the context of modelling. As discussed in Section 3.3, the detail necessary for descriptions at the physical-concept level depends on the application. Factors to be considered might involve, for example, the range of frequencies or magnitudes over which a high level of fidelity is required for certain variables of the model.

Within an organisation, guidelines are needed for formalising the information that must be provided before a model or sub-model is accepted for inclusion in a library. This should be consistent with the guidelines for documentation discussed in Section 9.2.1.

9.3 Multi-formalism in simulation and modelling

As mentioned in Chapter 3, it is often useful, especially in large and complex projects involving several design teams, to be able to use *different tools and languages* to build models of different sub-systems. Individuals may then develop and apply a sub-model using tools available within their own discipline and introduce that as an element within some larger model. The Virtual Test Bed (VTB) [11], discussed in Chapter 3 and mentioned again in the context of real-time simulation in Chapter 8, facilitates integration of sub-models developed using widely used tools. Entities can thus be exported from other software environments and incorporated into more complex system models within the VTB.

In one approach available with the VTB, there is no need for translation since the process is based on the use of 'wrappers'. Wrapped models based on sub-models created using software tools such as MATLAB®/Simulink® or ESL retain their original behaviour after being imported into the VTB environment.

It is also possible to bring sub-models into the VTB environment from Modelica® or VHDL-AMS using the Modlying software application [11], which involves a translation process. Modlying first uses software known as the Multi-Translator to convert the given sub-model into a universal XML-based specification. That XML specification is then translated into the form required by the VTB.

A third option for integrating sub-models from other software environments uses the COM interface provided by some software tools (such as MATLAB®) to integrate models into the VTB using a co-simulation approach (see e.g. [12]).

9.4 Generic models

Generic models are a development of the concept of a model library. A structure that is *generic* allows reuse of simulation software for a wide range of different projects with relatively minor reorganisation. Established examples of such a generic modelling approach can be found in application areas such as automotive engineering (e.g. [13] and [14]), gas turbines (e.g. [15] and [16]), electro-optic sensor system models ([17] and [18]) and spacecraft [19].

One example that illustrates the use of generic models very clearly is the European Space Agency (ESA) Generic Project Test Bed (PTB), which involves creation of reusable simulator architectures for spacecraft design [19]. In addition to spacecraft sub-system models, the architecture allows for ground-station models and also some aspects of the environment. The PTB is capable of real-time simulation and hardware-in-the-loop operation.

Making a model generic, even in a specialist application area, can present difficulties. The essential requirements of a generic description must be identified first and a suitable framework established to give the necessary flexibility to allow a number of more specific needs to be satisfied by that generic representation.

As already discussed, a system may need to be represented at several different levels of detail at different stages of a design project and this must also be possible with the generic approach. This means that sub-models, representing specific parts of the complete physical system, may be needed at a number of levels of complexity, ranging from purely functional forms at the initial stage to highly detailed and fully validated models in the later stages of the project. The models at different levels of resolution need to be mutually calibrated in some way. Ideally, the structures for the different levels of model will be directly related and the models at different resolutions will form an integrated group. The relationship between the different levels of each sub-model within the generic structure must be fully understood by users.

The most important benefit of the generic approach is likely to be a faster and less costly development process for new models compared with the traditional approach, which involves the development, on a one-off basis, of a specific new model for each new design task. Other benefits are likely because the development and application of a generic model demands a more systematic and rigorous approach to issues of model validation, together with better documentation.

9.5 Validation of library sub-models and generic models

Issues of model quality and model validation cannot be separated from other processes of model development. The modelling of a real system is an iterative process in which testing, evaluation and tuning are of central importance and, whatever the context, it is essential to ensure that the model being used is appropriate for the purpose. An application based on a model that does not have the necessary quality is bound to lead to difficulties.

Clearly, the elements within model libraries must have information about their purpose and limitations. Without good documentation, such libraries are of little value. In some types of commercial or defence-related applications, libraries may involve precompiled sub-models for which source code is not provided. This necessitates the use of a simulation language having special features and one example of such a language is the experimental OOSlim object-oriented simulation language ([20] and [21]). Good documentation is clearly very important in such cases as there is no opportunity for users to investigate directly the details of internal organisation of the sub-models.

Some general issues of validation for generic models are considered in [18] where the validation of a generic electro-optic sensor system model [17] is discussed. The generic model is, in this case, intended for use in the design of specific types of electro-optic systems such as infrared search and track systems, missile warning systems and thermal imager systems. Although this generic model appears quite specialised in terms of the application, the central ideas and methodology used in its development are applicable to generic models in many other fields.

The approach adopted for the development of the generic model for electro-optic sensor systems involved developing models of some specific electro-optic sensor system applications as an integral part of the process of developing the generic model. Particular configurations of the generic model could then be evaluated and tested against these self-contained specific models for the same application. As confidence in the generic model increases, new modules may be added to the generic structure, but such modifications have to be comprehensively tested for the particular configuration of the model investigated in the earlier tests and this should be based on regressive testing methods.

In applying a generic approach to model development, a need may arise for a model of a new application, not previously considered, using an available generic structure. This introduces new challenges which encourage reuse of established sub-models but further test the generic philosophy. If the approach fails at any point with a new application, then either a flaw has been found in the engineering design or a limitation has been found in the generic model. In the latter case, the generic model has to be modified and its capabilities extended.

9.6 Educational issues

Engineers are usually introduced to mathematical modelling and encounter computer-based modelling and simulation methods early in their university education. However, topics relating to model management are completely neglected in many courses and most students seldom have to give serious thought to what constitutes a simulation model that is fit for purpose. Indeed, all issues of model quality are often glossed over in a superficial fashion and the teaching often stops with the formulation of equations from physical laws and principles or with linear models obtained experimentally by system identification and parameter estimation methods. Also, students often do not make the vitally important link between design success and model quality and fail to appreciate that correction for model inadequacies at a late stage in a project can lead to major additional costs and delays in completion.

In the words of Hardy Cross, a former Professor of Civil Engineering at Yale, '. . . an important duty of teachers is to force students repeatedly back into the field of reality and, even more, to teach them to force themselves back into reality' [22]. In a modelling and simulation context, students must develop an understanding of the limitations of models and this has to begin at an early stage in their education and training. As part of that process, they must be introduced to the idea that models need to be properly managed and they must get into the habit of documenting models and the details of all the model testing processes that they apply.

The guide produced in 2007 by the UK Royal Academy of Engineering [23] and mentioned in Chapter 1 puts particular emphasis on the importance of integrated system design skills for all engineers. It presents a challenge to universities to develop courses that not only teach the fundamentals of an engineering discipline but also give their graduates the abilities to apply sound engineering principles to complex design problems involving more than that one specialist area.

Degree courses must cross traditional boundaries and include not only a sound foundation of theory but also realistic design exercises involving real engineering applications that have the potential to break down human barriers and encourage creative and innovative solutions. Modelling and simulation methods are of central importance for this approach to engineering education, just as they are for practical integrated system design.

Student exposure to modelling should range from initial scoping of problems, involving back-of-the-envelope investigations, through combined experimental and simulation studies of laboratory-scale hardware, to group project activities with students who have other specialisations. This may involve, in the later years of their courses, design projects based on truly integrated systems and may include virtual prototyping, embedded system design and hardware-in-the-loop simulation with due attention given to the use of model libraries and generic models. Careful management of the modelling process becomes vital to the success of the project and needs to be maintained throughout the design and development phase, through to prototype testing or commissioning. Students must start thinking in these terms from an early stage in their training and this means that they must be exposed to modelling and simulation repeatedly and creatively from the earliest stages of their engineering education.

9.7 References

[1] Brade, D. (2003) *A Generalized Process for the Verification and Validation of Models and Simulation Results*, Dissertation for degree of Dr. rer. nat., Universität der Bundeswehr, München, Germany.

[2] Kit, E. (1995) *Software Testing in the Real World*, Addison Wesley, Harlow, UK.

[3] Defense Modeling and Simulation Office (1996) ‘Department of Defense Verification, Validation and Accreditation (VV&A) Recommended Practices Guide’ (co-authored by Balci, O., Glasgow, P.A., Muessiny, P., Page, E.H., Sikora, J. and Youngblood, S.), Defense Modeling and Simulation Office, Alexandria VA, USA.

[4] Brade, D. and Waldner, C. (2003) ‘Automatic detection of behavioural specification violations’, in *Proceedings of the 03 European Interoperability Workshop, Stockholm, Sweden*, Simulation Interoperability Standardization Organization.

[5] Cloud, D.J. and Rainey, L.B. (eds) (1998) *Applied Modeling and Simulation: An Integrated Approach to Development and Operation*, McGraw-Hill, New York, NY, USA.

[6] Bruenese, A.P.J., Top, J.J., Broenik, J.F. and Akkermans, J.M. (1998) ‘Libraries of reusable models: theory and application’, *Simulation*, 71, 1, pp. 7–22.

[7] Top, J.L., Bruenese, A.P.J., Broenink, J.F. and Akkermans, J.M. (1995) ‘Structure and use of a library for physical system models’, in Cellier, F. and Granda, J.J. (eds), *Proceedings of the International Conference on Bond Graph Modeling and Simulation, Las Vegas, U.S.A.*, SCS Simulation Series 27, No. 1, pp. 97–102.

[8] Fritzson, P. (2004) *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*, IEEE Press, Piscataway NJ, USA.

[9] Simscape™ software, MathWorks Inc (online): *www.mathworks.com/products* (accessed 30 June 2011).

[10] MATLAB®/Simulink® modelling and simulation software, MathWorks Inc (online): *www.mathworks.com/products* (accessed 30 June 2011).

[11] VTB modelling and simulation package, the University of South Carolina, *The Virtual Test Bed* (online): *http://vtb.engr.sc.edu/vtbwebsite* (accessed 30 June 2011).

[12] McKay, W., Monti, A., Danti, E. and Dougal, R. (2002) 'A co-simulation approach for ACSL-based models', *Huntsville Simulation Conference 2002, 9–10 October*, Huntsville A, USA (online): *www.scs.org/confernc/hsc/hsc02/hsc/papers/hsc035.pdf* (accessed 1 August 2011).

[13] Sayers, M.W. (1999) 'Vehicle models for RTS applications', *Vehicle System Dynamics*, Vol. 32, No. 4–5, pp. 421–38.

[14] Andreasson, J. (2007) *On Generic Road Vehicle Motion Modelling and Control*, PhD Thesis, Royal Institute of Technology, KTH, Stockholm, Sweden. Also available online at *http://kth.diva-portal.org/smash/get/diva2:11627/FULLTEXT01* (accessed 1 August 2011).

[15] Visser, W.P.J. and Broomhead, M.J. (2000) *GSP. A Generic Object-Oriented Gas Turbine Simulation Environment*, Report NLR-TP-2000-267, National Aerospace Laboratory, Amsterdam, the Netherlands. Also available online at *http://www.nlr.nl/id~4296/lang~en.pdf* (accessed 1 August 2011).

[16] Alexiou, A. and Mathioudakis, K. (2009) 'Secondary air system component modelling for engine component

performance simulation', *ASME Journal of Engineering for Gas Turbines and Power*, Vol. 131, No. 3.

[17] Smith, M., Murray-Smith, D.J. and Hickman, D. (2007) 'Mathematical and computer modelling of electro-optic systems using a generic modelling approach', *Journal of Defense Modeling and Simulation*, Vol. 4, No. 1, pp. 3–16.

[18] Smith, M., Murray-Smith, D.J. and Hickman, D. (2007) 'Verification and validation issues in a generic model of electro-optic sensor systems', *Journal of Defense Modeling and Simulation*, Vol. 4, No. 1, pp. 17–27.

[19] European Space Agency, Modelling and Simulation, *Generic Project Test Bed* (online): *www.esa.int/TEC/Modelling_and_simulation/SEMTRH8LURE_0.html* (accessed 28 May 2011).

[20] Ostroversnik, M. and Murray-Smith, D.J. (1998) 'Real modularity in dynamic system simulation languages', *Systems Analysis-Modelling-Simulation*, Vol. 31, No. 3, pp. 69–182.

[21] Ostroversnik, M., Murray-Smith, D.J., Zupančič, B. and Strmčnik, S. (2000) 'OO PHYSBE model – a benchmark for modular object-oriented dynamic system simulation tools', in *Proceedings of the 3rd MATHMOD, Vienna, 2–4 February*, Vol. 2, pp. 595–8, Argesim/ASIM, Vienna.

[22] Cross, H. (1952) *Engineers and Ivory Towers* (edited by Goodpasture, R.C.), McGraw-Hill, New York NY, USA.

[23] Elliott, C. and Deasley, P. (eds) (2007) *Creating Systems that Work: Principles for Engineering Systems for the 21st Century*, the Royal Academy of Engineering, London, UK.

Further discussion

Abstract: This chapter brings together ideas presented and discussed in earlier chapters and discusses some areas for further research. It is concluded that tools for the efficient development and testing of physically based dynamic models are of central importance for a model-based concurrent approach to the design of integrated engineering systems. These tools include methods for forward and inverse simulation, model management, sensitivity analysis, system identification, parameter estimation, model optimisation and partial-system testing. Topics for further research within some of those areas are identified.

Key words: model management, research, model limitations, model quality.

10.1 A summary of some strategic issues in the modelling and simulation of integrated systems

Complex engineering systems, involving closely integrated elements from different engineering disciplines, became increasingly important during the last two or three decades of the twentieth century, initially in the aerospace and defence sectors but increasingly in other areas as well. Now integrated systems applications are becoming commonplace in fields such as the automotive industry, marine engineering, electrical power generation and distribution, and chemical and pharmaceutical process engineering. Within all these areas a multidisciplinary and model-driven concurrent approach to design is taking over from traditional sequential design procedures. Benefits arising from this more concurrent approach, if applied correctly, include earlier detection and correction of design flaws, and new possibilities for optimisation at the level of the complete system.

Because of the central role of modelling and simulation techniques in integrated system design and within computer-aided engineering in general, a strategy is needed to ensure that model-based design methods are applied appropriately so that system development times and costs can be reduced. Simulation techniques do already offer advantages in the development of new products through providing a faster and more cost-effective approach compared with the use of physical prototypes alone. However, current practice in system modelling and simulation within many organisations still lacks coherence and clarity in terms of strategy. The *ad hoc* approach often adopted contrasts strongly with accepted procedures in software engineering where rigorous processes of testing, documentation and version control are an integral part of the development cycle in most organisations. As pointed out in Chapter 9, universities have an important part to play in properly preparing young engineers for careers in which modelling and simulation methods will be of central importance and major changes are needed in the approach to modelling and simulation within most present-day engineering courses.

One key issue is that, in the modelling and simulation of any complex system, it is essential for all involved to have a full understanding of how the model is to be applied. This is especially important when a project includes several teams and involves people from different engineering disciplines.

The purpose of a model may also influence the type of description needed. It is usually important in engineering applications to retain as much information as possible about the corresponding real system and physically based models may therefore be preferred. This is especially true in a multidisciplinary design environment where engineers have widely different backgrounds and experience, and where models and computer simulations provide an important means of communication about technical issues.

The types of applications mentioned in this book show that, in addition to displaying nonlinear behaviour, most credible physically based models of engineering systems involve significant uncertainties in the early stages of their development. Important simplifications may also have to be introduced, often for reasons of computational complexity, if the model is to be useful for a design application or in the development of a real-time system simulator. Internal verification, external validation and testing of such models then become important issues.

Although experimental modelling techniques and the ideas of system identification and parameter estimation are usually associated with the modelling of an existing system, these methods can also be useful within the processes of design. When used as a tool for refinement of physically based models, system identification techniques also need to provide an indication of the accuracy of parameter estimates and of the validity of the model structure. In parametric models, questions of accuracy can be closely linked to issues of numerical identifiability and thus to experimental design. However, in many cases, especially with nonlinear parametric models, establishing the accuracy of estimated quantities is not straightforward. In the case of non-parametric models, useful insight concerning the range of validity of an overall model can be gained from the use of frequency-domain measures such as coherence.

Following the successful application of identification and parameter estimation methods, simulation tools can be used in the evaluation of the resulting models and for the assessment of competing hypotheses, even in cases where model uncertainties remain. Such an approach can lead to the formulation of new experiments and to a further stage of model refinement, if that is considered necessary for the intended application.

All models have limitations and one aim in applying external validation procedures must be to define properly and attempt to understand those limitations. Practical validation investigations can cover only a finite, and often relatively small, number of test cases. Attempts should therefore be made to establish confidence in a model so that its performance can be recognised as being reasonable for the specific objectives associated with the application, rather than trying to establish its validity or quality in a general way. An additional practical problem is the fact that quantitative measures of model credibility are hard to define for complex multi-input multi-output models and, as models become more complex, there are increasing problems of visualisation of model and system behaviour. New methods are needed for displaying results efficiently for multi-output situations and for models where there can be strong interactions between parameters. It is believed that the type of polygon diagram discussed in Chapter 7 may offer interesting opportunities for new types of display. One advantage of that approach to visualisation is that it is very flexible in terms of the comparisons that are possible. For example, these diagrams can allow results of sensitivity analysis to be displayed in a simple and efficient fashion, and are also appropriate for use with deterministic measures of performance, such as overshoot magnitudes or frequencies of oscillations.

In addition to specific modelling techniques and simulation methods mentioned previously, some emphasis has also been given in this book to the reuse of sub-models and to ideas associated with generic models. These topics are particularly important in the modelling of complex systems, especially when the model is being developed using physical principles. Many of the topics mentioned in earlier chapters are still the subject of ongoing research and development work, and it is appropriate to highlight a few that may be particularly significant in future.

10.2 Research and development work on modelling and simulation methods for integrated system applications

Traditionally, most research in the field of continuous system simulation has been concerned with improved numerical methods and with the development of enhanced simulation environments. These topics are still important and areas where there is significant research of this kind include the development of improved methods for the numerical solution of ordinary differential equations and differential algebraic equations, efficient treatment of discontinuities, improved methods for model reduction and the development of improved methods for user-computer interaction at run time.

However, one important feature of present-day research on simulation is an increased emphasis on some of the broader aspects of the model development process (such as bond-graph and other energy-based modelling procedures, issues of causality in simulation models, model validation methods and systematic processes for assessing, correcting and documenting models that are used in engineering design). These topics are all important in the move towards more efficient model-driven design methods.

Issues of experimental design, which are already recognised as being very important for the successful identification of systems, are also of value elsewhere within the model development process and especially in the external validation of simulation models. Assessing the adequacy of a model for a specific use is a difficult task and the problem of upgrading or tuning a model which is shown to be inadequate for a specific application raises many questions. Terms such as 'model testing' or 'model evaluation' are probably more appropriate than the word 'validation', which may give a false impression of model capabilities. As mentioned earlier, theories can be proved to be wrong but cannot ever be proved to be right and the 'unknown unknowns' mean that there can never be a simple conclusion in the processes that we conventionally call 'model validation'. This is not a problem that can be 'solved' through further research, but there is nevertheless valuable insight that can be gained from further research and development work in this general area, and this is an area still attracting much attention in the aerospace and defence sectors.

Inverse systems also receive significant attention in this book and are the subject of ongoing research. Areas that require further attention include improving the computational efficiency and speed of inverse simulation methods, and ensuring that inverse simulation tools are more user-friendly and can be applied successfully by non-specialists.

Issues of robustness and convenience of the user interface also arise in the context of evolutionary optimisation methods. For example, genetic algorithms are potentially important for the automation of optimisation procedures in model development. However, these methods could, at their present stage of development, only be applied routinely in an industrial design environment if there was a significant period of training for those involved. Similarly, the successful application of the genetic programming approach depends critically on choosing appropriate functions within a function library and this requires good understanding of the likely physical phenomena in the system under investigation. Therefore, at present, genetic programming is unlikely to provide any kind of fully automated approach to the development of complex models, and more research on this topic is needed.

Good management of models is essential for the success of design projects and user-friendly methods for building up system and model documentation are very important. Some human-factor issues that arise in model management at present could well be reduced or eliminated by the use of layered structures in documentation systems where an individual could seek information at different levels of detail and for different levels of prior knowledge. Development of appropriate management systems for this might well necessitate research collaboration between design engineers, computer scientists, psychologists, educationalists and management specialists. Potential benefits of improved systems for model management within a model-based design process could be a reduction in design risks, especially in terms of cost inflation during the design and development process, and possible late delivery.

It is generally accepted that an integrated approach to design should, ideally, involve use of generic forms of description and reusable sub-models. Much remains to be done in many application areas to make these ideas more widely applicable. As outlined in Chapter 9, established examples of such a generic modelling approach can already be found in a number of application areas, but there are many fields where little has been achieved in this respect. Object-oriented methods are relevant both for this and for assisting with sub-model reuse, and some specific software environments may offer significant advantages for the development of reusable and readily extendable models. Further research and development work is clearly needed on the principles of generic models. The practical advantage of adopting a generic approach also needs to be evaluated through additional applications in new areas.

Appendix A1: models of an unmanned underwater vehicle (UUV)

A1.1 An outline description of the basic nonlinear model of the vehicle

The equations of motion of an underwater vehicle may be represented either in body-fixed or earth-fixed frames of reference (see e.g. [A1.1]). Using standard notation, the general body-fixed vector representation involves the nonlinear equations [A1.1]:

$$\mathbf{M}\dot{\boldsymbol{\upsilon}} + \mathbf{C}(\boldsymbol{\upsilon})\boldsymbol{\upsilon} + \mathbf{D}(\boldsymbol{\upsilon})\boldsymbol{\upsilon} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{A1.1}$$

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\boldsymbol{\upsilon} \tag{A1.2}$$

Here the matrix $\boldsymbol{M}$ is the inertia matrix, with added mass effects included. The matrix $\mathbf{C}(\boldsymbol{\upsilon})$ involves Coriolis and centripetal terms, again with added mass effects included. The matrix $\boldsymbol{D}(\boldsymbol{\upsilon})$ represents damping terms, $\boldsymbol{g}(\boldsymbol{\eta})$ is the vector of gravitational forces and moments, τ is the vector involving all externally applied forces and moments and $\boldsymbol{J}(\boldsymbol{\eta})$ is the transformation matrix which relates the body-fixed and earth-fixed coordinate systems.

The body-fixed frame involves the six translational and rotational velocity variables, as defined by the vector $\upsilon(t) = [u(t), v(t), w(t), p(t), q(t), r(t)]^T$, relative to a constant velocity coordinate frame which moves with the ocean current velocity vector $\boldsymbol{u}_c$.

The six components in the global reference frame are defined by the vector $\boldsymbol{\eta}(t) = [x(t), y(t), z(t), \phi(t), \theta(t), \psi(t)]^T$. Here the angles $\phi(t)$, $\theta(t)$ and $\psi(t)$ are related to the body roll, pitch and yaw variables through the Euler transformations. The external forces and moments given by the vector $\boldsymbol{\tau} = [X, Y, Z, K, M, N]^T$ include gravitational, buoyancy, hydrodynamic and propulsive terms and the inputs providing the external forces and moments for control of the vehicle are generated by control surface deflections at the rudder ($\delta_r(t)$), port bow plane ($\delta_{bp}(t)$) and starboard bow plane ($\delta_{bs}(t)$), the stern plane ($\delta_s(t)$), and forces proportional to the propeller speed ($n(t)$) and buoyancy adjustment ($B(t)$) ([A1] and [A2]).

A set of six nonlinear equations for surge, sway, heave, roll pitch and yaw motion can then be derived. The parameters for the specific case of the U.S. Naval Postgraduate School (NPS) AUV II vehicle are as given by Fossen [A1] and these values are, in turn, based on information provided by Healey and Lienard [A2]. This vehicle has a length of 5.3 m and mass 5454.54 kg, and the total number of mechanical and hydrodynamic parameters for the model is well in excess of 100. Relevant software developed by Fossen and his colleagues at the Norwegian University of Science and Technology is available for downloading [A3].

The UUV model may be converted into standard state space form from Equations (A1.1) and (A1.2) to give:

$$\begin{bmatrix} \dot{\eta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} J(\eta)v \\ M^{-1}\{-(C(v) - D(v))v - g(\eta) + \tau\} \end{bmatrix} \tag{A1.3}$$

This equation is in the standard state-space form for a nonlinear model, i.e.:

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u}) \tag{A1.4}$$

where $\boldsymbol{x}$ is the state vector and $\boldsymbol{u}$ is the input vector [A4].

Changes to the model given by Healey and Lienard [A2] had to be introduced to make the simulation model behave appropriately for the operating conditions and manoeuvres of interest. These changes in the model of Healey and Lienard are associated with the fact that the published model equations were intended originally for use in the development of nonlinear control systems for manoeuvres involving specific ranges of forward speed [A2]. Use of the same model for open-loop and manual control investigations over a greater speed range led to difficulties and reinforces points made elsewhere in this book about problems likely to be encountered when a model developed for one application is used in other investigations.

For the purposes of modelling, it has been assumed that the UUV has one propulsive force, although the vehicle has two propellers. The thrust produced is represented by the *bilinear thruster model* suggested by Fossen [A1]. This involves a relationship for propeller thrust of the form:

$$T = T_{nn}|n|n + T_{nu}|n|u \tag{A1.5}$$

where:

$$T_{nn} = \rho D^4 \alpha_1 \tag{A1.6}$$

$$T_{nu} = \rho D^3 \alpha_2 \tag{A1.7}$$

$$\alpha_1 = 0.12 - 0.5\alpha_2 \tag{A1.8}$$

and where D (propeller diameter) = 0.3 m, $\alpha_2 = -0.16$ and the corresponding value of $\alpha_1 = 0.019$. These values were chosen to provide the surge velocity profile required for the NPS AUVII [A2]. Since the efficiency of the propeller is likely to be of the order of 70 per cent, the force developed by the propeller along the longitudinal axis of the vehicle is given by:

$$X_{prop} = 0.7T = 0.7(T_{nn}|n|n + T_{nu}|n|u) \tag{A1.9}$$

Since the two propellers operate together in a counter-balancing fashion, the rotational effects of the propellers are assumed to be negligible in the model.

The model has been further extended by my colleague Dr E. McGookin to include the load on the motor shaft due to rotation of the propeller in the water [A4]. This representation involves considering the propeller as a rotating disc. The input is the torque to be applied to the propeller to balance its drag and inertia components. Both this torque and the drag torque can be combined to give an equation of motion for the propeller of the form:

$$T_{SHAFT} = I_o\dot{n} + \frac{1}{2}\rho C_D n^2 R_{DISC}{}^3 \tag{A1.10}$$

where R_{DISC} is the radius of the disc used in the rotating disc approximation referred to above, the parameter C_D is the associated drag coefficient, I_0 is the moment of inertia of the propeller, n is the rotational speed of the propeller and ρ is the density of water. Therefore, the load torque on the propeller, T_{LOAD}, is calculated as:

$$T_{LOAD} = -T_{SHAFT} = -I_o\dot{n} - \frac{1}{2}\rho C_D n^2 R_{DISC}{}^3 \tag{A1.11}$$

This expression, which provides the load on the shaft, involves only one uncertain parameter which is the drag coefficient, C_D. However, a disc rotating about its longitudinal axis has a drag coefficient of 1.369×10^{-3} [A5] and, in the absence of any more precise value, this has been adopted. Note that this representation is approximate because it takes no account of added mass or the shaft dynamics [A4].

Figures A1.1 and A1.2 show how the vehicle responds when it starts from an initial condition with all state variables set to zero and a propeller input of 1500 rpm applied at time $t = 0$ s. As may be seen in Figure A1.1, the surge velocity increases from zero to a steady value of about 3.8 m/s. At time $t = 10$ s, a rudder deflection of 20 degrees is applied as a step function and an immediate response that can be seen in terms of the yaw rate and yaw variables. The sway velocity also starts to change at that time and reaches a steady value of about 0.9 m/s. It should be noted that because there are no inputs applied to the other control surfaces, the pitch changes in response to the propeller and the rudder inputs, and reaches a steady value of about 20 degrees, with an associated steady heave velocity of about −0.05 m/s. This means that in the

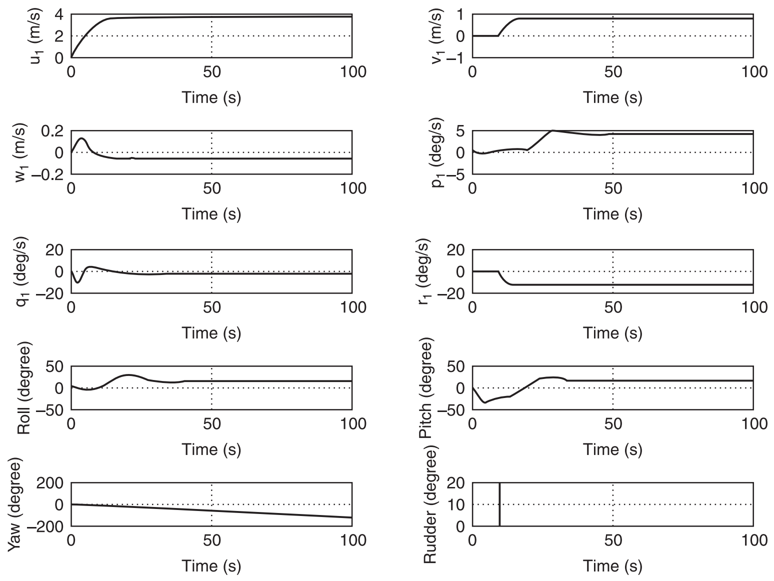


Figure A1.1 **Simulated response of vehicle from an initial condition with all state variables set to zero and a propeller input of 1,500 rpm applied at time *t* = 0 s**

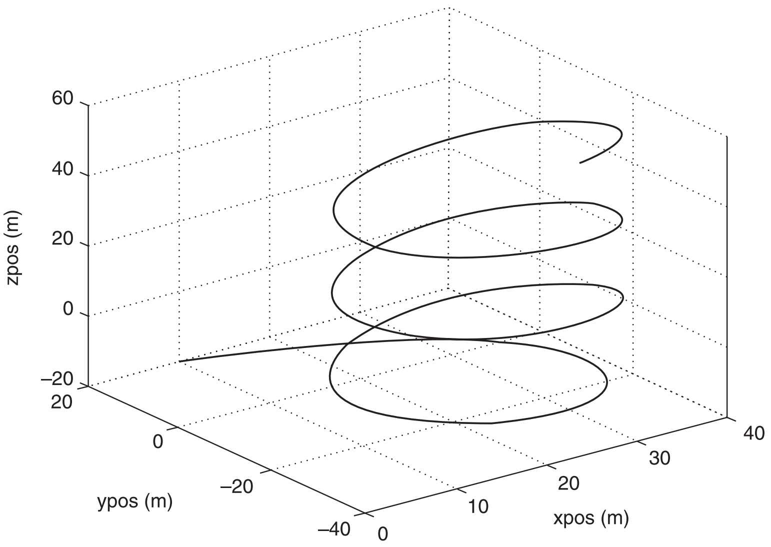


Figure A1.2 Earth-axis representation in terms of position for transient responses shown in Figure A1.1

earth-axis system the vehicle is moving upwards at a constant rate from about time t = 20 s and this may be seen in Figure A1.2 where the vehicle trajectory is seen to take the form of a rising spiral.

A1.2 A linearised model describing diving motion

A linearised dynamic model describing diving motion of the underwater vehicle model involves a third-order system with the following equations [A1]:

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \dfrac{M_q}{I_y - M_{\dot{q}}} & -\dfrac{\overline{BG_z}W}{I_y - M_{\dot{q}}} & 0 \\ 1 & 0 & 0 \\ 0 & -u_0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} \dfrac{M_\delta}{I_y - M_{\dot{q}}} \\ 0 \\ 0 \end{bmatrix} \delta_s \tag{A1.12}$$

where M_q, $M_{\dot{q}}$, and M_δ are hydrodynamic constants, I_y is moment of inertia vehicle about the *y* axis, *W* is the weight of the vehicle, $\overline{BG_z}$ is the vertical distance between the centre of buoyancy and the centre of gravity of the vehicle, and u_0 is the forward speed of the vehicle. The state variables q, θ and z have the conventional meanings, as given in Section A1.1 above, and represent pitch rate, pitch angle and vertical displacement respectively. The variable δ_s represents the stern plane deflection.

In simplified form, this equation becomes:

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & 0 \\ 1 & 0 & 0 \\ 0 & -u_0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} b_1 \\ 0 \\ 0 \end{bmatrix} \delta_s(t) \qquad \text{(A1.13)}$$

and for typical operating conditions, using parameter values derived from [A1] and [A2], $a_{11} = -0.7$, $a_{12} = -0.3$, $u_0 = 1.832$ m/s and $b_1 = 0.035$. Figure A1.3 shows a typical response from this model in terms of the change of pitch angle θ following a step change of stern plane deflection δ_s.

The relationship between the stern plane deflection input, δ_s, and pitch angle, θ(s), may be expressed [A1] as a transfer function:

$$\frac{\theta(s)}{\delta_s(s)} = \frac{b_1}{s^2 - a_{11}s - a_{12}} = \frac{\dfrac{M_\delta}{I_y - M_{\dot{q}}}}{s^2 - \dfrac{M_q}{I_y - M_{\dot{q}}}s + \dfrac{\overline{BG_z}W}{I_y - M_{\dot{q}}}} \qquad \text{(A1.14)}$$

and similarly, for the pitch rate:

$$\frac{q(s)}{\delta_s(s)} = \frac{sb_1}{s^2 - a_{11}s - a_{12}} = \frac{\dfrac{M_\delta}{I_y - M_{\dot{q}}}s}{s^2 - \dfrac{M_q}{I_y - M_{\dot{q}}}s + \dfrac{\overline{BG_z}W}{I_y - M_{\dot{q}}}} \qquad \text{(A1.15)}$$

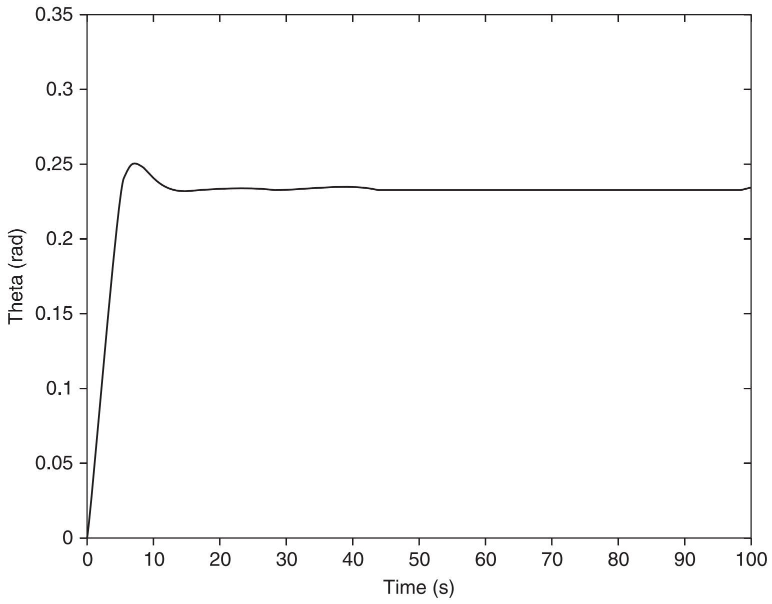


Figure A1.3 **Typical response of the linearised UUV model showing the change of pitch angle θ following a step change of stern plane deflection δ_s**

and for the depth:

$$\frac{z(s)}{\delta_s(s)} = \frac{-u_0 b_1}{s(s^2 - a_{11}s - a_{12})} = \frac{-u_0 \dfrac{M_\delta}{I_y - M_{\dot{q}}}}{s\left(s^2 - \dfrac{M_q}{I_y - M_{\dot{q}}}s + \dfrac{\overline{BG_z}W}{I_y - M_{\dot{q}}}\right)} \tag{A1.16}$$

The transfer functions for the pitch and pitch rate variables involve the characteristic equation:

$$s^2 - a_{11}s - a_{12} = s^2 - \frac{M_q}{I_y - M_{\dot{q}}}s + \frac{\overline{BG_z}W}{I_y - M_{\dot{q}}} = 0 \tag{A1.17}$$

and the natural frequency for pitching motion is therefore given by:

$$\omega_n = \sqrt{a_{12}} = \sqrt{\frac{\overline{BG_z}W}{I_y - M_{\dot{q}}}} \tag{A1.18}$$

with relative damping factor for pitching motion given by:

$$\zeta = \frac{-a_{11}}{2\sqrt{a_{12}}} = \frac{-M_q}{2\sqrt{\overline{BG_z}W(I_y - M_{\acute{q}})}} \tag{A1.19}$$

Thus the period of pitching oscillations will be affected directly by the effective moment of inertia term (I_y – $M_{\acute{q}}$) and inversely by the weight of the vehicle *W* and the distance between the centre of buoyancy and the centre of gravity ($\overline{BG_z}$). A reduction of the effective moment of inertia or an increase in either of the other two factors will thus increase the natural frequency of the oscillations. The damping of the pitch response depends directly on the hydrodynamic coefficient M_q and inversely on the terms (I_y – $M_{\acute{q}}$), *W* and ($\overline{BG_z}$).

A1.3 Model of the electrical drive system

The model of the underwater vehicle, outlined above, was combined with a model of an appropriate electrical drive system. Figure A1.4 is a schematic diagram of the complete UUV system model, showing the interactions between the electrical sub-models and the vehicle model [A4].

The electrical sub-model of Figure A1.4 includes a battery which is connected to a DC to AC inverter, involving a three-phase six-switch network producing a variable-frequency AC waveform. This inverter is, in turn, coupled to an induction motor. The controlled switches in the inverter are

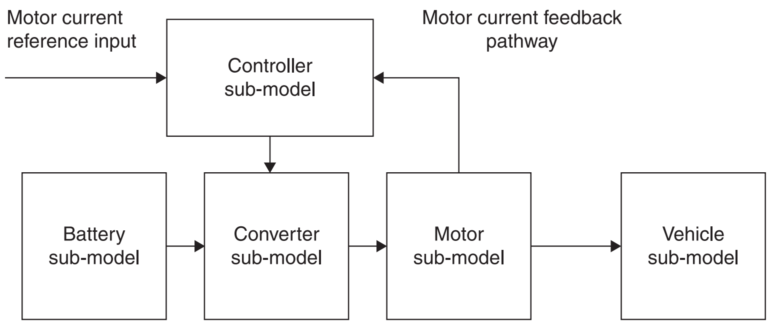


Figure A1.4 **Schematic diagram of the complete UUV system model showing interactions between the sub-models**

operated by a pulse-width modulated controller switching at a frequency of 5kHz and the controller incorporates proportional plus integral (PI) control for pulse timing of the converter switches in order to maintain a set current level in the motor. Switch timings are determined through comparison of sinusoidal and triangular waves. The relative amplitudes of the two waveforms are adjusted by the feedback system and switching occurs when the sine and triangular waves intersect. High-frequency harmonics in the AC output from the converter are filtered out and the result is then supplied to the induction motor as input. The motor is connected directly to the propeller.

A1.4 References

[A1] Fossen, T.I. (1994) *Guidance and Control of Ocean Vehicles*, Wiley, Chichester, UK.

[A2] Healey, A.J. and Lienard, D. (1993) ‘Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles’, *IEEE*

Journal of Oceanic Engineering, Vol. 18, No. 3, pp. 327–39.

[A3] MSS. Marine Systems Simulator (2010) (online): *www.marinecontrol.org* (accessed 19 September 2011).

[A4] Zenor, J.J., Murray-Smith, D.J., McGookin, E.W. and Crosbie, R.E. (2009) 'Development of a multi-rate simulation model of an unmanned underwater vehicle for real-time applications', in Troch, I. and Breitenecker, F. (eds), *Proceedings of the MATHMOD '09 Symposium*, Full Papers CD Volume, pp. 1950–7, Argesim, Vienna, Austria.

[A5] Hoerner, S.F. (1968) *Fluid Dynamics Drag*, Hoerner Publications, New York NY, USA.

Appendix A2: numerical methods for the solution of ordinary differential equations

A2.1 Introduction

Although it is assumed that readers of this book have some knowledge of numerical analysis methods, a brief summary of some of the essential concepts of numerical integration is provided here. Readers requiring more detailed information about numerical analysis methods and especially the numerical solution of ordinary differential equations should use one of the many texts on the fundamentals of continuous system simulation (e.g. [A1] to [A5]). Online user handbooks and help systems provided with most modern software provide information about methods available within specific simulation tools.

Single-step integration methods, which are the basis of many widely used algorithms, depend on approximating derivative terms in a Taylor series by evaluating first derivatives at a number of points within the current integration step. If $x(t)$ is the value of the variable x of interest within a given first-order ordinary differential equation at the start of the integration step and $x(t + h)$ is the value at the end of the step, then:

$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2!}x''(t) + \cdots + \frac{h^n}{n!}x^n(t) + \cdots \quad \text{(A2.1)}$$

where x', x'' and x^n are the first, second and nth derivatives of $x(t)$. The value of the variable x at the end of the time interval h is thus given by the value of x at the start of the interval plus components from each of the derivatives of x.

Many methods of numerical integration are based on Taylor series approximations which involve finite-difference representations for the second-order and higher derivatives. Methods may be classified, broadly, depending on whether they are *single-step* or *multi-step* techniques and whether they involve *explicit* or *implicit* formulae. In the explicit approach, one pass through the formulae produces all the required values, whereas in the implicit case, iterative methods must be used to find a solution. *Semi-implicit* methods are also available where the iterative approach applies only to a sub-set of the values to be found. The *order* of the method depends upon how the Taylor series is truncated and this truncation gives rise to the *truncation error*.

A2.2 Single-step integration methods

Single-step integration methods are widely used and involve approximating the derivative terms in the Taylor series by the first-derivative at a number of points within the integration interval and taking a weighted average of those values. The first-order Runge-Kutta formula (also known as the 'forward Euler' method) is a simple example and is obtained from a Taylor series with terms involving h^2 and above removed. For this the derivative is evaluated at the start of the step and is applied over the whole interval to provide the increment to be applied to x to give the approximate value at the end of the step. An implicit algorithm that is similar to this is the backward Euler method where the derivative is a function of the variable x at the end of the integration interval rather than at the beginning. These ideas can be applied to more complex approximations. For example, the second-order Runge-Kutta formula involves removal of the Taylor series terms after the third (i.e. terms involving h^3 and above), but in this case the derivative is evaluated at the start of the step and at another point within the integration interval. Implicit methods inevitably introduce additional computational demands compared with the explicit approach, but may have advantages in terms of accuracy.

It should be noted that for methods that have a starting point that is the start of the integration step, the initial value of the relevant variable (x) and the initial value of the time (t) are both known. This means that no information is required about values in previous steps and such algorithms are said to be *self-starting*. One benefit of a self-starting algorithm is that the integration step size can be adjusted without reference to the step size in previous intervals, leading to *variable-step* integration methods.

The truncation error depends on the integration step size h and the error may be controlled through the choice of this quantity. However, *numerical stability* issues also have to be taken into account. As well as being more accurate, implicit integration methods are also usually more stable than explicit methods. Most single-step methods that involve automatic variation of step-length give error bounds that are more restricting than the corresponding bounds in terms of stability. Explicit formulae can often give rise to problems if the step size is too great and automatic variation of step size can result in very slow speeds of solution. It should be noted that, although variable step length methods are commonly used for many applications, *fixed-step* integration is important for real-time simulation, as discussed in Chapter 8.

A2.3 Multiple-step methods

Multiple-step integration methods use values from previous integration steps as well as values from the current step. First derivative evaluations from previous steps are stored and provide a basis for estimating higher-order terms in the Taylor series. One example of the multiple step approach is the Adams-Bashforth method, which can involve either explicit or implicit formulae. In most situations, use of an implicit formula with a multiple-step method provides improved accuracy compared with the corresponding solution using an explicit formula.

One refinement of the multiple-step approach involves use of a *predictor-corrector* approach. In such methods an explicit formula (the *predictor*) is used to obtain a fairly close approximation, which is then refined using an implicit *corrector* formula.

Multiple-step methods are not self-starting as they require values from previous integration intervals. Starting can be achieved by using a single-step method for the first few integration intervals before switching to the multiple step approach. One benefit of the multiple-step approach is more accurate assessment of the error, since stored information from past steps can be used to provide improved error estimates.

Variable-step integration presents more problems with multiple-step integration methods than with single-step methods, since stored values from previous steps may relate to a period when a different step size was being used. Interpolation is needed to allow new values to be found that are suited to the new integration interval.

A2.4 Problems of stiffness

Difficulties arise with models that involve both very fast and very slow dynamics (i.e. models that have a wide range of eigenvalues). Such models are said to be *stiff* and a number of specialised integration algorithms are available for such problems (see e.g. [A4]). The degree of stiffness in a model depends on the ratio of the largest to the smallest eigenvalue and, in a simple model involving a cascade of first-order sub-models, this is represented simply by the ratio of the largest to the smallest time constants. One widely used approach to the solution of stiff problems is *Gear's method*, which is a predictor-corrector algorithm [A6].

A2.5 References

[A1] Gear, C.W. (1971) *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice Hall, Englewood Cliffs NJ, USA.

[A2] Bennett, B.S. (1995) *Simulation Fundamentals*, Prentice Hall, Hemel Hempstead, UK.

[A3] Matko, D., Karba, R. and Zupančič, B. (1992) *Simulation and Modelling of Continuous Systems*, Prentice Hall, Hemel Hempstead, UK.

[A4] Murray-Smith, D.J. (1995) *Continuous System Simulation*, Chapman and Hall, London, UK.

[A5] Lee, H.J. and Schiesser, W.E. (2004) *Ordinary and Partial Differential Equation Routines in Fortran,* *Java®, Maple®, and MATLAB®*, Chapman and Hall/ CRC, Boca Raton FL, USA.

[A6] Gear, C.W. (1984) 'Efficient step size control for output and discontinuities', *Trans. Society for Computer Simulation*, Vol. 1, pp. 27–31.

Index

Page numbers in italic indicate sources of further information