

Development and Validation of Nonlinear Models for Helicopter Dynamics

Gary James Gray

Department of Electronics and Electrical Engineering

University of Glasgow

A thesis submitted for the degree
of Doctor of Philosophy
to the University of Glasgow.

ProQuest Number: 13815418

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 13815418

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Thesis
9335
copy 1



Abstract

The need for validated nonlinear helicopter models and methods to validate these models directly is identified. Published validation methods for validating nonlinear dynamic models are reviewed and the need for an integrated approach is established. Sensitivity coefficient based validation techniques are investigated. Single value sensitivity coefficients are found to be useful for parameter and output variable selection. Examination of sensitivity coefficients time histories is found to be a useful addition to parametric validation methods. A model distortion technique is evaluated. The method is tested with simple systems and simulated data as well as a helicopter model and real flight data. The method is discussed. Its application to helicopter dynamics is rejected because of noise problems. A nonlinear one degree of freedom yaw model for an Aerospatiale SA.330 PUMA helicopter is improved and validated using analogue matching and a parameter estimation method which uses a linear search. The importance of physical knowledge of the system being modelled is highlighted in the development of the model. A nonlinear mathematical model of a helicopter main rotor is validated in two specific areas. These are the lag damper and the engine/rotor speed model. The validation techniques used are maximum likelihood parameter estimation with sensitivity coefficient examination and analogue matching of time response data. The importance of good physical knowledge of the system being modelled is again indicated. The structure of the model in the identified areas is validated. The validation methods are brought together in a specification for an interactive model validation computer package. The benefits of an integrated approach are identified and the computer program is specified so as to take advantage of this. Through this package, the user will interact with the model, the available validation methods and the experimental data and will be able to develop and validate dynamic models easily and efficiently.

CONTENTS

Page number

Chapter 1

Introduction

1.1	Introduction	11
1.2	Helicopter modelling	13
1.3	Applications	15
1.3.1	Piloted simulation	15
1.3.2	Flight control systems	15
1.3.3	Design modifications	16
1.4	Background to research	17
1.4.1	System identification based validation methods	19
1.4.2	Rotor dynamics	20
1.5	Nonlinear model validation	20
1.6	Summary of thesis	21

Chapter 2

Nonlinear Model Validation Methods

2.1	Introduction	24
2.2	Analogue matching	25
2.3	Parameter estimation	26
2.3.1	Parameter selection	27
2.3.2	Estimation methods	29
2.3.3	Sensitivity coefficients	30
2.4	Nonlinear validation using parametric techniques	30
2.5	Model distortion	31
2.6	Inverse simulation	32
2.7	Open loop simulation	33
2.8	Conclusions	35

Chapter 3

Sensitivity Methods

3.1	Introduction	38
3.2	Calculation of sensitivity coefficients for nonlinear systems	38
3.3	Single value sensitivity coefficients	42
3.3.1	Experimental application of single value sensitivity coefficients	43
3.4	Sensitivity coefficient time histories	46
3.5	Discussion	49

Chapter 4

Investigation of Model Distortion Technique

4.1	Background	51
4.2	Application of model distortion technique to helicopter dynamics	52
4.3	Theory	53
4.4	Experimental Analysis	56
4.4.1	Second order system with simulated data	56
4.4.2	Noise response	60
4.4.3	Third order system	63
4.4.4	Nonlinear system	65
4.4.5	Linear helicopter model	66
4.5	Discussion	70

Chapter 5

Development and validation of a helicopter yaw model

5.1	Introduction	73
5.2	Flight data acquisition system on research PUMA	74
5.3	Experiments and flight data	75
5.4	Development and Validation of Yaw model	76
5.4.1	Helicopter yaw dynamics - theory	77
5.4.2	Pitch-flap coupling	80

5.4.3	Investigation of the effect of drag on the tailboom	81
5.4.4	Investigation of vortex ring and wind effects	83
5.4.5	Main rotor downwash	93
5.5	Unmodelled cross-coupling dynamics from aircraft motion	101
5.6	Conclusions	101

Chapter 6

Validation of a Main Rotor Model

6.1	Introduction	103
6.2	The rotor model	103
6.3	Parameter estimation based model validation methods	109
6.3.1	Investigation of model structure	110
6.4	Sensitivity methods	110
6.5	Maximum likelihood parameter estimation - Theory	111
6.6	Experimental	113
6.6.1	Computer implementation of parameter estimation	116
6.7	Lag damper model validation	116
6.7.1	Parameter estimation results	119
6.7.2	Derivation of original value for lag damper damping coefficient	122
6.7.3	Lag damper validation - conclusions	123
6.8	Engine model validation	123
6.8.1	Engine/rotorspeed model	123
6.8.2	Parameter estimation	125
6.8.3	Effect of flight data sampling rate	128
6.8.4	Time delays in the system	131
6.8.5	Engine model validation - conclusions	132
6.9	Investigation of an alternative model structure for rotor speed dynamics	133
6.10	Investigation of the effect of blade twist	136
6.11	Conclusions	136

Chapter 7

An Interactive Model Validation Computer Program

7.1	Introduction	139
7.2	Specification for a model validation and development package	139
7.2.1	User interface	139
7.2.2	System simulation	140
7.2.3	Parameter estimation	140
7.2.4	Sensitivity coefficients	141
7.2.5	Animated time responses with respect to parameter variations	141
7.2.6	Parameter selection	143
7.2.7	Model structure investigation	143
7.2.8	Input and output	143
7.2.9	Other validation methods	143
7.3	Implementation of validation program	144
7.4	Applications	147

Chapter 8

Conclusions and Suggestions for Further Work

8.1	Conclusions	148
8.1.1	Validation methods	148
8.1.2	Applications of model validation methods	150
8.1.3	Summary	150
8.2	Suggestions for further work	151

Appendix 1

Model Distortion Method Program Listings	153
---	------------

Appendix 2

Yaw Model Development Program Listings	161
---	------------

Appendix 3

Main Rotor Model Development Program Listings	168
--	------------

List of Figures

	page number
1. Comparison of open loop and closed loop simulation	34
2. Methods of partial differentiation of signum function	39
3. Block diagram of dynamic system and sensitivity cosystem	41
4. Sensitivity coefficients for vertical velocity output for each of the sixty-four parameters in the system A matrix (8 state linear helicopter model) for collective step input	46
5. Section of engine model time response to main rotor collective step with sensitivity coefficients for engine parameters	48
6. Time response of second order system	57
7. Parameter time histories for second order system with simulated data	58
8. Time response of second order system with time varying parameters	59
9. Parameter time histories for second order system with white noise added to the input signal	61
10. Parameter time histories for second order system with white noise added to the simulated data	62
11. Parameter time histories of second order system using simulated data from a third order system	64
12. Parameter time history of second order system using simulated data from a nonlinear system	65
13. Time response of Z_w	66
14. Time responses of Z_w and Z_θ where optimisation is done separately for each parameter and for both together	68
15. Time response of linear helicopter model compared with flight data	70
16. Jack Calibration	76
17. Schematic representation of helicopter showing forces and moments	78
18. Yaw model time response to step input on tail rotor collective pitch in hover with and without pitch flap coupling	81
19. Airflow over helicopter tailboom	82

20.	Aerospatiale PUMA tailboom	82
21.	Time response of yaw model with pitch flap coupling to step input in tail rotor collective pitch in hover	84
22.	Diagram showing tail rotor disk in main rotor downwash	87
23.	Time response of expected tangential inflow arising from wind blowing main rotor downwash over tail rotor for tail rotor collective step input	89
24.	Variation of sum of squared error between flight data and model output for yaw model where downwash is blown over the tail rotor while varying wind parameters	90
25.	Time response of yaw model with wind blowing main rotor downwash over the tail rotor with optimised parameters	93
26.	Variation of parameter INFCON which represents the fraction of main rotor downwash contributing to tangential inflow at the tail rotor	95
27.	Block diagram of helicopter engine, gearbox and rotor speed controller	104
28.	Rotor hub of an Aerospatiale SA330 PUMA helicopter	105
29.	Time response of nonlinear rotor model to the main rotor collective step inputs used in flight experiment 1	114
30.	Time response of nonlinear rotor model to the main rotor collective step inputs used in flight experiment 2	115
31.	Bar chart showing relative sensitivities of output variables to lag damper damping coefficient	118
32.	Time response of blade lag angle for flight experiments one and two with original model and estimated model	121
33.	Time response of blade lag angle for one flight experiment with original model and estimated model from the other experiment	122
34.	Time response of main rotor collective pitch for flight experiment 1 (system input)	125
35.	Time response of model output and flight data of rotor speed for different engine parameter values	127
36.	Sensitivity coefficients of engine parameters T_e and G over time history of system transient	128

37.	Frequency response of low pass filter used on differentiated azimuth data to represent rotor speed	129
38.	Comparison of rotor speed data with differentiated and filtered azimuth data	130
39.	Time response of rotor model where engine parameters have been estimated using differentiated azimuth data	131
40.	Flight data and experimental input from flight experiment 2 with parameter values estimated with data from experiment 1	132
41.	Time response of rotor model for two model structures for rotor speed dynamics	134
42.	Comparison of residuals for the two different model structures for rotor speed dynamics	135
43.	Example of window showing animated time response algorithm output with system time response over experimental data and parameter values	142
44.	Possible appearance of validation program output showing windows based output with various applications running	144
45.	Transputer network for simulating four systems in parallel	146

Tables

1.	Levels of helicopter theoretical model	18
2.	Single value sensitivity coefficients for lag damper parameter $A\zeta$	118.

Acknowledgements

This research has been carried out in the department of Electronics and Electrical Engineering at the University of Glasgow. The author would like to thank Professor David J. Murray-Smith of Glasgow University for his support and supervision.

The support of Dr. Gareth D. Padfield and Mr. Andrew T. McCallum of Defence Research Agency, Bedford is gratefully acknowledged as is the assistance in the computer department of Miss Anne McKinnon, Mr. Stephen Gallacher and Mr. Dougald Campbell of the Department of Electronics and Electrical Engineering, Glasgow University.

The author gratefully acknowledges the support of the Science and Engineering Research Council and the Defence Research Agency (Aerospace Division) at Bedford.

Chapter 1

Introduction

1.1 Introduction

A mathematical model is an abstract representation of a physical system. Such a model can be analysed and experimented upon as if it were a real system but only if one has confidence in the model's representation of the real system. That confidence comes from an very important part of the model building process - model validation.

Model validation is the process in which the model is checked against the real system to ensure its validity as a mathematical representation of that system. If mathematical models are to be used in safety critical applications areas such as the aerospace industry, it is essential that they are adequately validated. Linear mathematical models are derived from nonlinear system equations based on physical principles by linearising around a specific operating point. This means that the range of flight conditions over which these models are valid is usually small. A nonlinear model derived from the physical equations can be valid over a much wider range of operating points and as a result can be much more useful. It is therefore very important that nonlinear models are validated and tested and that proven techniques for validation of nonlinear models are available.

There are various aspects to the model validation process. These can be described using terminology defined by Murray-Smith and Carson¹. They include the following,

- i) Theoretical Validity where the model is of sound theoretical foundation based on known theories.
- ii) Empirical Validity where there is agreement between the model behaviour and all available data concerning the system.
- iii) Pragmatic Validity defines the extent to which the model is suitable for its intended application.

- iv) Heuristic Validity defines the potential of the model to predict system behaviour and to test hypotheses.

Empirical validity is particularly important in development of mathematical models of dynamic systems and should be checked throughout the development and refinement process as well as during the subsequent application of the model.

There are many possible methods of validation for mathematical models. For linear models, the frequency domain can be used with techniques borrowed from system identification. For nonlinear models, it is common to work in the time domain. These time domain nonlinear model validation methods are often based on model refinement techniques where the model is fine-tuned to fit the experimental data and results of this fine tuning are compared with those for different experimental data sets and with theoretical values. Such methods can be either structural or parametric in nature, although, where the model is based on physical equations, there may be limited room for manoeuvre as far as model structure is concerned if theoretical validity is to be maintained. This leaves parametric techniques which include methods based on the sensitivity of specific model outputs to specific parameters. Such parametric techniques are often based on minimisation of a cost function over an experimental time history, or perhaps involve more specific examination of a particular part of the system dynamic response to some input such as the magnitude of an overshoot or the steady-state response.

The experimental data used in the model validation process is very important. It is often desirable to design experiments specifically for a particular validation exercise. This is because it may be desirable to isolate one particular sub system or to excite specific dynamic modes. Additional instrumentation may also be required to monitor variables not normally recorded. The selection of the operating point of the dynamic system at which the experiment is to be conducted is also obviously very important as is the magnitude of the input and response.

Ideally, one half of the available experimental data should be used for the model development and refinement process and the other half should be used to check the robustness of the model. Robustness checks should be done using similar data to that used in the development and refinement stages. The difference between the model output and the robustness data should be small if the model is to be accepted as an adequate representation of the system.

If a model is shown not to agree with some of the experimental data, the validation method can often highlight the deficiencies in the model and provide the basis for possible improvements. These improvements should of course be validated and checked for robustness. In this way, validation is an integral part of the model development process in an iterative modelling/validation loop.

An important application of model validation is to define the boundaries of the system for which the model is valid. This could represent the range of operating conditions of the real system, for example, the flight envelope of an aircraft. Towards the edges of the flight envelope, nonlinearities in the system become more prominent and effects such as aerodynamic compressibility become important and alter the dynamic characteristics of the aircraft. This means that very often, a large increase in model complexity is necessary to increase the range of operating points over which the model is valid. One very important function of model validation is to define this range of operation or validity.

1.2 Helicopter modelling

Mathematical models are used extensively in the aerospace industry because of the high costs of product development and aircraft operation. A validated model of an aircraft can be used confidently to analyse such characteristics as aircraft dynamics, performance and handling qualities cheaply and safely. Real time models can be used in piloted simulation for handling qualities evaluation. This is useful for testing flight control systems or investigating aircraft modifications. One other major application area of nonlinear mathematical models is in the nuclear power industry again for reasons of cost and safety.

In the aerospace industry, helicopters present particular modelling difficulties because of their highly cross-coupled and nonlinear dynamics compared to fixed wing aircraft. This is partly due to the very wide range of flight conditions in which a helicopter is capable of operating. For example, forward speed can vary from zero at hover to over 200 knots (100 m/s). The dynamics associated with the main rotor are highly nonlinear. As well as the nonlinearities of incompressible flow, there are compressibility effects, tip vortices and their effects on the following blade, and blade elasticity in flap, lag and torsion all of which alter the flow of air over the blade and the forces and moments generated by the aerodynamics. Another difference between the helicopter and an aeroplane is the asymmetry originating with the main rotor. In a

fixed wing aeroplane, both wings see the same airspeed. This is not always the case with a helicopter. In forward flight, the rotation of the blades at one side of the rotor adds rotational velocity to the aircraft velocity and at the other side of the rotor, the rotation of the rotor subtracts rotational velocity from the aircraft velocity. This means that the blades on one side of the rotor encounter a higher airspeed than the blades at the other side of the rotor. In a single rotor helicopter, the most common configuration, a large torque is generated which wants to rotate the fuselage of the helicopter in the opposite direction to the rotation of the rotor. This necessitates an anti-torque device like a tail rotor and strongly influences helicopter dynamics in the yaw degree of freedom.

As mentioned above, many of the problems inherent in helicopter modelling originate with the main rotor dynamics. The main rotor generates the lift and thrust for the helicopter and functions as the main control surface for control of the aircraft. The rotor consists of between two and seven blades each of which flaps independently both in the plane of rotation, and perpendicular to the plane of rotation. The pitch angle of each blade is determined by the pilot's controls altering the lift distribution across the rotor and therefore the magnitude and direction of the force and moment generated. The rotor is powered by one or more engines. There is an optimum rotor speed based on the point where the advancing blade tip would begin to experience aerodynamic compressibility effects. It is desirable to maintain the rotor speed at this value. This function is often controlled by an automatic rotor speed control system controlling the fuel flow into the engine. The torque reaction on the fuselage caused by the rotation of the main rotor is compensated for by a smaller vertical rotor with pedal controlled variable pitch blades attached to the tail fin. This tail rotor also provides aircraft yaw control. On some helicopters, there are aerodynamic surfaces attached to the airframe which provide lift in forward flight, or in the case of the tail fin, assist in aircraft yaw control.

In aerospace engineering, mathematical models are often linearised about a particular operating point or flight condition. Such small perturbation models are very useful for fixed wing aeroplane applications, but, because of the highly nonlinear nature of helicopter dynamics, such small perturbation models are valid for only a small range of flight conditions. This means that any helicopter model valid for large excursion manoeuvres involving different flight conditions must be nonlinear. Such a nonlinear model should be derived from the physical equations of mechanics and aerodynamics of the aircraft to give a set of nonlinear differential equations.

1.3 Applications

1.3.1 Piloted Simulation

One of the main applications of a fully validated nonlinear mathematical model is ground based piloted simulation. Feedback from the pilot after a simulation run can give considerable insight into aircraft handling qualities and can highlight the dependence of various flight characteristics on specific system parameters. Piloted simulation does mean that the simulation has to be run in real time and depending on the available computing facilities, this can limit the complexity and the number of the differential equations in the model.

However, a realistic simulation is possible in real time and can have many applications. Piloted simulation is useful for the examination of the effects of design modifications on the helicopter without the expense and time involved in satisfying the engineering and flight safety considerations necessary for experiments involving a real aircraft. The use of piloted simulation to investigate the effect of different rotor features is described in Tomlinson and Padfield². Piloted simulation is also very important for the assessment of handling qualities improvements due to flight control systems.

1.3.2 Flight Control Systems

Helicopters have always demanded more from a pilot in terms of flying skill and the amount of concentration required compared with fixed-wing aircraft. This is because helicopters are inherently unstable for some flight conditions and are subject to a high degree of cross-coupling. The instability imposes a high pilot workload even for the maintenance of a trimmed flight condition in some cases, eg. hover. The cross-coupling between the lateral and longitudinal modes also makes flying more difficult and more tiring for the pilot.

A flight control system can make the helicopter more stable and easier to fly while reducing the effects of cross-coupling and improving the aircraft handling characteristics.

An accurate mathematical model is needed for the design of such a control system. Models developed for control system design need to be linear for the control system

design algorithms. These algorithms maximise handling qualities or performance criteria using optimisation techniques. These linear models are derived by linearising the full nonlinear system equations at specific operating points. With a system such as a helicopter which has strong nonlinearities, system dynamics can change considerably when the operating point or flight condition changes. A controller is designed for one specific linear model. If the model characteristics change with flight condition, robustness becomes a very important factor in the design of the control system. Robust control strategies include gain scheduling which involves varying control system parameters depending on operating point. Validation of the linear models is important to determine how far one has to stray from an operating point before the model derived at that point becomes useless. Currently these linearised models are used for design of low authority stability augmentation systems, but in the future, with better models, full authority fly-by-wire systems should be available.

Once a control system has been designed, it must be tested thoroughly by simulation before it can be put in an aircraft. This testing should be done with as accurate a nonlinear model as is available and it should be tested at the helicopter performance limits as well as the normal operating conditions. It is important that the nonlinear model used has been validated for the flight conditions at which the control system has been tested. The testing of the control system and the nonlinear model together can show up any tendency to instability of the complete system and any excess loads or stresses on any part of the helicopter, for example, the main rotor.

It is also necessary to run a piloted simulation of the helicopter with the control system to test the handling qualities of the system. Again, an accurate model is required, but the restriction of real time operation may mean that the model has to be simplified depending on the available computing power. Any simplified model would of course have to be validated. The real time nonlinear model can be validated using the non-real time full nonlinear model.

1.3.3 Design Modifications

A fully validated nonlinear mathematical model can also be very useful in the product development process. If an engineer is confident about the model's predictive capability and if the model is based on physical equations, simulation can be used to investigate the effect of design modifications to the aircraft without the engineering and safety related expense of physically altering the aircraft and flight testing. In this

way, simulation could be used to optimise physical aircraft parameters based on performance.

1.4 Background to the research

In recognition of the problems inherent in the development and validation of mathematical models of helicopters, research into helicopter modelling, simulation and validation has been ongoing at the Defence Research Agency (Aerospace Division) at Bedford (DRA Bedford) for a number of years. Early research looked at basic modelling for simulation of 'gentle flying'³. Research progressed over the years to HELISIM⁴, a generic model developed at DRA Bedford to represent agile flight. HELISIM was successfully used to establish vehicle and control system design parameters that affected handling in agile flight. Initial validation of HELISIM consisted of analogue response matching of model output with flight data. Although parametric trends were broadly correct, there were some deficiencies in the model which revealed themselves in agile flight simulation. Better validation methods were required to determine whether the fault lay in the model structure or the parameter data set. This led to joint research activity by DRA Bedford and Glasgow University into model validation and development methods.

Model structure has been conveniently be divided into three distinct levels of increasing complexity by Padfield⁵ as shown in table 1. The main difference is in the type of main rotor model.

	Level 1 (Aggregated aerodynamic loads)	Level 2 (Individual blade)	Level 3
Aerodynamics	Linear (2-D) Dynamic inflow/local momentum theory Analytically integrated loads	Nonlinear (Limited 3-D) Dynamic inflow/local momentum theory Local effects of blade/vortex interaction Unsteady 2-D Compressibility Numerically integrated loads	Nonlinear (3-D) Full wake analysis (free or prescribed) Unsteady 2-D Compressibility Numerically integrated loads
Dynamics	(i) Rigid blades 6 dof quasi-steady rotor 9 dof - rotor flapping 12 dof - flap & lag 15 dof - flap & lag & pitch	(i) Rigid blades Options as in Level 1 (ii) Limited number of blade elastic modes	(i) Elastic modes (detailed structural) representation
Application	Parametric trends for flying qualities/ performance studies Within operational flight envelope Low bandwidth control	Parametric trends for flying qualities/ performance studies Beyond operational flight envelope Medium bandwidth appropriate to high gain active flight control	Rotor design Rotor load prediction across a high bandwidth Beyond operational flight envelope

Table 1 Levels of helicopter model (from Padfield⁵)

Most of the real time piloted simulation research to date has concentrated on level one models. This can include nonlinear six degree of freedom rigid body models with analytically integrated blade loadings, dynamic inflow and other dynamic elements such as an engine/rotor speed model and blade actuator dynamics. Level one models are useful within the normal flight envelope where most of the aerodynamic approximations used are valid.

Level 3 represents comprehensive rotor/fuselage dynamics which could be used for high bandwidth flight control system design. As a step towards a level 3 model, the level 2 model has been proposed, the main distinguishing features of which are likely to be nonlinear, unsteady aerodynamics integrated along elastic modal shapes. The level 2 requirements are further discussed by Padfield⁵.

1.4.1 System identification based validation methods

Much of the research being done at DRA Bedford and the University of Glasgow involves model validation based on system identification techniques. System identification methods are used to generate linear models which can be compared with flight data and values for parameters which can be compared with parameters from linearised nonlinear models such as HELISTAB, the generic nonlinear helicopter model developed at DRA Bedford. An important part of the system identification process is the type of the control input used for the flight experiments. A 'good' control input should be easy to apply, excite the desired dynamic modes, and give a good record length. Most of the test inputs used in this research have consisted of binary signals and frequency sweeps. The effectiveness of these and of optimal input design has also been the subject of research at Glasgow University⁶.

The first stage in the analysis of the experimental flight data is preliminary time history interpretation and comparison including a kinematic consistency check. This is done using software developed at DRA Bedford as part of a system identification and model validation package developed through a joint DRA Bedford/University of Glasgow research programme and used as a research tool. This initial stage includes a Kalman Filter/Smoother and state estimator which removes some of the measurement and process noise. The next stage is to define the model structure. Model Structure Estimation is done using equation error parameter identification techniques. This process establishes the degrees of freedom of the model and the initial conditions for the estimation of the system parameters. Step-wise regression is used in either the time or frequency domain. At each step, an independent variable is added to or deleted from the regression equation until a best fit is obtained. The variable chosen for entry into the regression equation at each stage is the one having the highest correlation with the residual (the difference between the model output and the flight data).

This should be followed by Maximum Likelihood output error parameter estimation to give unbiased, minimum variance estimates. This part of the system identification process caused problems when applied to helicopters although it is widely used for fixed-wing aircraft. The probable reason for this is that the validity of the model structure has not been checked. The model structure validity could be affected by incorrect cross-coupling coefficients or unmodelled nonlinearities.

Ongoing system identification research at DRA Bedford and the University of Glasgow is discussed in detail in Padfield, Thorne, Murray-Smith, Black and Caldwell⁷.

1.4.2 Rotor Dynamics

Some of the recent identification and validation research done at DRA Bedford and Glasgow University involves main rotor dynamics, specifically rotor flapping and aerodynamic inflow representation. Inflow is defined as the induced velocity of the air flowing down through the main rotor.

Houston and Black⁸ used a rotor model linearised from the nonlinear level one equations for a PUMA helicopter in the hover with first order inflow as a basis for a parameter identifiability study using frequency domain parameter identification techniques. Identifiability defines whether or not a model can be derived from the available experimental data using a given identification method and is therefore of prime importance in mathematical modelling.

The conclusion of the research was that such linear three degree of freedom models (inflow, blade flap and vertical body acceleration) can be identified from coning and vertical acceleration data but identified model parameters (aerodynamic derivatives) are sensitive to experimental frequency range and a priori estimates used as initial guesses especially with respect to dynamic inflow parameters.

1.5 Nonlinear model validation

In validation studies, the nonlinear physical equations are often linearised about a particular operating point to give a state space description of the model. ie. A set of linear first order differential equations. This format is very appropriate for control system design and with a linear model, frequency domain algorithms can be used. The frequency domain is very convenient for helicopters because much of the unmodelled dynamics is at the rotor frequency or higher. This can be effectively removed in a frequency domain method. However, as mentioned above, parametric identification techniques used on these models can give problems if an output error technique such as maximum likelihood is used. A properly validated nonlinear model would increase confidence in any linearised model derived from it. A validated nonlinear model is

also going to be useful for piloted simulation and could also be used as a basis for further model development research into areas such as dynamic inflow.

As part of the programme of development for HELISTAB, the nonlinear helicopter model developed at DRA Bedford, the development of the mathematical representation of rotor dynamics was considered very important. This is recognised as the main source of nonlinearities in helicopter flight mechanics and is a major area of current helicopter mechanics research. In 1989, when this research began the nonlinear real time helicopter model as used at DRA Bedford represented individual blade dynamics in the flap degree of freedom, the blade lag degree of freedom was not modelled and the engine model including the rotor speed control system had not been validated. The yaw degree of freedom also warranted investigation. This mainly involves the tail rotor which serves both as a means of aircraft yaw control and an anti-torque device to counteract the effect of the main rotor. Initial validation studies⁴ using time response comparisons with the HELISTAB model showed helicopter yaw response to pedal inputs to be another part of the model in need of development and validation.

It was also desirable at this stage to develop and investigate techniques of nonlinear model development and validation. It would be very useful to be able to validate nonlinear models directly i.e. without prior linearisation. Indeed it should be a prerequisite to any application of a mathematical model.

This thesis will investigate various validation techniques for nonlinear mathematical models of helicopters. It will be shown that a variety of parametric techniques can be very effective when used together, and a specification for a comprehensive model validation computer package will be suggested. The results of the validation will also be presented as suggested modifications to the DRA Bedford nonlinear model, HELISTAB in the representation of aircraft yaw dynamics, rotor blade lag dynamics and the engine.

1.6 Summary of thesis

Chapter 1 gives an introduction to model validation, the development of nonlinear models and applications of nonlinear helicopter models. Existing nonlinear model validation methods are reviewed in chapter 2. These methods are for the validation of nonlinear physical models using experimental data.

Chapter 3 introduces and evaluates sensitivity coefficients as a validation tool in the form of single value sensitivity coefficients and sensitivity coefficient time histories. Both were found to be valuable as a basis for validation especially when used together with other methods such as parameter estimation.

In chapter 4, a model distortion technique was investigated. It was found to be unsuitable for helicopter modelling mainly because of problems caused by noise on experimental data. It may however have potential in other applications where noise contamination is less of a problem.

In chapter 5, a helicopter yaw model was developed. This was a one degree of freedom nonlinear physical model for turns in the hover. Various dynamic features were considered for inclusion in the model. Validation methods used were analogue matching, a linear search parameter estimation technique and the application of physical knowledge of helicopter yaw dynamics. It was concluded that main rotor downwash has an effect on tail rotor thrust in the hover. In chapter 6, a model of the main rotor of a helicopter was validated with respect to the representations of the lag damper and the engine/rotorspeed controller. Maximum likelihood parameter estimation and sensitivity coefficient methods were the validation methods used.

The research is brought together in chapter 7. This chapter specifies an interactive computer program which allows the application of all the validation methods investigated in his thesis to a particular modelling problem. This chapter illustrates the benefit of an integrated approach to model development and validation.

Chapter 8 presents the conclusions and some suggestions for further work.

References

- 1 Murray-Smith, D.J. and Carson, E.R. '*The modelling process in respiratory medicine*' in Cramp, D.G. and Carson, E.R. (eds), '*The respiratory system*', Croom Helm, London, pp296-333, 1988.
- 2 Tomlinson, B.N. and Padfield, G.D., '*Piloted simulation studies of helicopter agility*', Vertica, Vol 11, pp 665-684, 1987.
- 3 Wilcok, T. and Thorpe, A.C., '*Flight simulation of a Wessex helicopter - a validation exercise*', RAE Technical report 73096, 1973.

- 4 Padfield, G.D., '*A theoretical model of helicopter flight mechanics for application to piloted simulation*', RAE Technical report 81048, 1981.
- 5 Padfield, G.D., '*Theoretical modelling for helicopter flight dynamics: development and validation*', Proc. 16th ICAS Congress, 1988.
- 6 Leith, D. and Murray-Smith, D.J., '*Experience with multi-step test inputs for helicopter parameter identification*', Vertica, Vol 13, No. 3, pp403-412, 1989.
- 7 Padfield, G.D., Thorne, R., Murray-Smith, D.J., Black, C.G. and Caldwell, A.E., '*UK research into system identification for helicopter flight mechanics*', Vertica, Vol. 11, No. 4, pp665-684, 1987.
- 8 Houston, S.S. and Black, C.G., '*On the identifiability of helicopter models incorporating higher order dynamics*', Fifteenth European Rotorcraft Forum, Paper 67, 1989.

Chapter 2

Nonlinear Model Validation Methods

2.1 Introduction

There are many methods for the validation of nonlinear models. The most common method of validating a nonlinear model with experimental data is analogue matching where the model output is compared graphically with the data in either the time or frequency domain. Parameter estimation is a more accurate technique which is widely used in aerospace. Parameter estimation involves statistically estimating values for system parameters. These estimates can be compared with theoretical values or estimates from other model structures or experimental data sets.

Parameter estimation involves calculation of the sensitivity coefficients, that is the derivatives of the model output with respect to the parameter being considered. Visual examination of sensitivity coefficient time histories provides a lot of information about the system and can be used as part of an integrated validation process or for parameter selection prior to estimation.

For more complicated nonlinear models, various validation methods have been or are being developed. The model distortion method was developed in the nuclear power industry²² and gives a quantitative validity for the model although the methods usefulness for validation of nonlinear helicopter models is doubtful. Inverse simulation and open loop validation are being developed specifically for application to helicopter modelling problems and are promising techniques. Inverse simulation uses the flight path of the helicopter, the system output, as the input to an inverse model, the output of which is the flight control time histories which can then be compared with the actual control inputs. It is of greatest application for modelling of large excursion manoeuvres. Inverse simulation reduces problems due to errors being integrated in a forward simulation and causing divergence of model output from flight data. Open loop validation allows subsystems to be validated by using measured aircraft states as inputs to the system.

This chapter investigates the above techniques and represents a survey of model validation techniques for nonlinear dynamic models with an emphasis on aerospace dynamics.

2.2 Analogue Matching

One of the simplest and most frequently used model validation techniques is analogue matching. This is where the model output is compared graphically with available experimental data by superimposing a plot of one on the other. This can be done in either the time or frequency domain. The criterion for model validity is 'a good fit' between model output and experimental data.

Analogue matching is often used to compare different models of the same system. The model giving the best fit is selected by a visual comparison of the time response of the models with the experimental data. This method was used by deLeeuw and Hui¹ to investigate three degree of freedom and six degree of freedom linear representations of a Bell-205 helicopter. It is often used in non-aerospace applications. For example, time response matching is used to compare different models of solar irradiation on an inclined plane by Feuermann² and different representations of fabric weave in textiles are compared by Clapp and Peng³. Time history comparison is also a useful technique in the initial stages of a model development and validation process as in the initial development work on the nonlinear generic helicopter model developed at the Defence Research Agency (Bedford)⁴ where the time history comparison revealed areas of the model in need of further development. Analogue matching becomes a more difficult problem with multi-output systems. Where many output variables have to be matched, it is often found that improving the fit on one output variable has a worsening effect on the fit for another output variable and it is often not clear which is correct. As the number of output variables increases, it becomes more difficult to get an overall match.

Time history comparison can be a useful validation method but although a poor fit can help identify whether or not a particular part of the model is an accurate representation of the system, it is difficult to identify further possible model developments. Model validation should be part of an iterative development/validation process and as such, the indication of further model developments is an important part of any model validation method.

Generally time response matching is only one part of the validation process and it is used in concert with other methods. Many validation studies include a time history comparison as part of the model validation process^{5,6,7,8,9,10}.

An important subset of time history comparison is comparison of the trim conditions. The trim conditions are the initial conditions of the dynamic system before the experiment and they represent the steady state, that is, an operating point where, with no change in input, the dynamic system variables would theoretically remain at the same values ad infinitum. A mathematical model is trimmed by setting the time differentials of the dynamic states to zero. The experimental trim values can be compared with the theoretical values derived from the model. This validation technique is defined by Mansur and Tischler¹¹ as static validation and is used in conjunction with dynamic validation (defined as time history comparison as described above) in the validation of an Apache AH-64 helicopter model. Static validation can be used to identify measurement offsets and to generally set up the model. An accurate set of initial conditions is important because any errors in the initial conditions will integrate to large discrepancies in long simulation runs.

Response matching is also possible in the frequency domain⁵. Frequency response comparison is widely used in aerospace and especially helicopter dynamics¹². This is because confidence in the model is usually a lot higher regarding the lower frequencies representing the aircraft body motions than in the higher frequencies representing the rotor modes. The model validity can effectively be examined for the body states in isolation from the rotor modes. The same principle can be used to identify possible areas of model development. If, for example, the fit is not as good at the rotor rotation frequency for a helicopter model, the rotor dynamics could be investigated as the next stage of model development.

2.3 Parameter Estimation

Parameter estimation is a common technique in model validation. It involves statistically estimating system parameters and then comparing the results with other parameter estimates from different experiments or with theoretical values for these parameters.

The parameters can be physical quantities in the nonlinear system equations or coefficients in a linearised transfer function or state space model. i.e. where the equations are set out as follows,

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

where,

\dot{x} = time differential of the state vector, x ,

A = matrix of coefficients in set of first order differential equations,

x = state vector,

B = matrix of input coefficients in system equations,

u = system input.

With linear models the parameters can be estimated in the time or frequency domain, while with nonlinear systems, the time domain must be used. For helicopter applications, frequency domain identification has proved very useful because the portion of the frequency spectrum that is of interest can be selected thus excluding any high frequency noise^{7,10}. As well as values for the parameters, the parameter estimation process also produces Cramer-Rao bounds (see Iliff¹³) which provide an indication of the variance of estimated parameters. The value of the Cramer-Rao bound is of more use for comparisons between estimates for different parameters or different experiments than as an absolute representation of variance but nevertheless provide useful information as to how accurate the parameter estimate is and how near it must be to the theoretical value for the model to be valid. The value of the sensitivity coefficients is an important factor in the calculation of the Cramer-Rao bounds.

A comprehensive study of parameter estimation is given in Beck and Arnold¹⁴. Parameter estimation and its application in the aerospace industry along with detailed examples of the parameter estimation procedure is reviewed in Iliff¹³. A complete review of the use of system identification and parameter estimation for model development and validation is given in an AGARD lecture series¹⁵. This lecture series covers the complete process specifically for helicopter applications and includes papers on experimental design, data processing, data quality evaluation, system identification techniques in time and frequency domains, robustness issues, examples and applications of system identification.

2.3.1 Parameter selection

The selection of which parameters to estimate is an important part of the parameter estimation process. It is not always possible to identify all of the parameters from the given data. Identifiability and ultimately parameter selection depends on the

experimental input signal and the available data. Leith, Bradley and Murray-Smith¹⁶ in a validation study of helicopter coupled blade flapping/inflow models showed that identifiability is an important consideration in any validation study.

Step-wise regression is an efficient method of selecting parameters. It involves using an equation error technique which minimises a cost function representing the error between experimental data and a dynamic variable using a least squares algorithm in a sequence of steps, each time adding or deleting an independent parameter to the regression equation until a best fit to experimental data is obtained. This is effectively model structure estimation since the addition of independent variables to the regression equation means that the output variable is now dependent on another dynamic variable and the order of the model has been increased by one. As each independent variable is added to or deleted from the regression equation, the statistical confidence in the expression is measured and the optimal model structure occurs when confidence is at its highest. This is not necessarily where the model order is highest because the addition of extra independent variables can decrease confidence if, for example, there are linear dependence problems between the new variable and an existing one. Step-wise regression is an effective technique of parameter selection and model structure estimation and it has been used by Klein¹⁷ with fixed wing aircraft applications and by Padfield, Thorne, Murray-Smith, Black and Caldwell¹⁰ with helicopter models.

For full nonlinear models consisting of the nonlinear physical equations, physical parameters will have a certain confidence associated with their theoretical values. For example, in a real time helicopter model, the parts of the physical equations concerning dimensions or mass moments of inertia will be more accurate than the lift and drag coefficient at the blade tip or any approximations to blade bending or twisting modes. Weaker areas of the model can be concentrated on in the parameter estimation by selecting parameters in which confidence of the theoretical values is low or parameters in a particular part of the model suspected to be in need of further development.

Another method of parameter selection is to use sensitivity information. Sensitivity coefficients give the importance of a specific parameter for a particular output. Single value sensitivity coefficients are used for parameter selection in chapter 3 of this thesis.

2.3.2 Estimation methods

The equation error method of step-wise regression as detailed above, produces estimates of parameter values. These estimates are biased because of measurement noise. To produce unbiased results, it is necessary to use an output error or a maximum likelihood technique both of which are widely used in aerospace.

Output error estimation is based on minimising a cost function which represents the error between model output and experimental data. This cost function is usually least squares and it can include weights for different outputs and a priori parameter information. Maximum likelihood estimation is based on maximising the probability of the model output fitting the experimental data exactly given the parameter values¹⁸. There is no bias due to measurement noise and if a Kalman filter is included in the estimator, process noise can also be accounted for. Maximum likelihood estimation with process noise accounted for is therefore more accurate than output error, but a Kalman filter will only work with a linear system or a moderately nonlinear system using a first order linearised representation¹⁹. Therefore, if process noise is not considered a problem, or if the model is very nonlinear, it is sometimes necessary to use an output error estimator. Output error estimation is a subset of maximum likelihood which does not account for process noise.

The results of the parameter estimation are often compared with theoretical values for the parameter calculated from the original physical equations. When different model structures are being tested, it can be useful to compare estimates of common parameters with each other and with the theoretical value¹. It can also be useful to compare the estimates calculated using different experimental data sets with the same type of experiment.

The technique of comparing estimated and theoretical values is used by Kaletka, Tischler, von Grünhagen and Fletcher¹² in both the time and frequency domains for six degree of freedom helicopter model identification with analogue matching used as a further validation tool and by Gawthrop, Mirab and Li⁵ in the time domain with a nonlinear model of a flexible robot arm.

Parameter estimates can be used for model development if there was uncertainty about the original value and if confidence in the estimated value is higher than in the a priori theoretical value. This technique is used in Blackwell, Feik and Perrin⁶ with a

linearised small perturbation model of a helicopter rotor including blade lag dynamics and rotor speed dynamics and by Gawthrop, Mirab and Li⁵ using parameter estimation with a model of a robot arm. The model with the new parameter values should of course be validated with a different set of experimental data.

2.3.3 Sensitivity coefficients

The sensitivity coefficient is the derivative of the model output with respect to a system parameter. Most parameter estimation methods involve minimising a cost function using a derivative based search method, so the calculation of the sensitivity coefficient is important. Methods of calculating sensitivity coefficients together with an examination of the problem when the model is nonlinear is given in chapter 3.

Sensitivity coefficients can also be used as an integral part of the model validation and development process. Lehman and Stark²⁰ use single value sensitivity coefficients and sensitivity coefficient time histories for nonlinear model validation. Single value sensitivity coefficients are used to validate a linearised model against the nonlinear model from which it was derived by comparison of the sensitivity coefficients. Sensitivity coefficient time histories are used as well as single value sensitivity coefficients to provide clues as to system dynamics and the possible effects of specific parameters on these dynamics.

2.4 Nonlinear Validation using Parametric Techniques

The on-going research programme at Glasgow University and DRA Bedford has included a lot of work with parameter estimation for linear models in the frequency domain. Helicopters however, have highly nonlinear characteristics and models linearised about a particular operating point are valid over only a small range. Despite this, by estimating aerodynamic coefficients and plotting these together with the theoretical values against flight condition (eg. forward speed) the trend of the coefficient can be compared with the theoretical trend for that parameter. The comparison of individual estimates with theoretical values gives a guide to the validity of the linearised models. The trend can also give insight into the nonlinear model from which the linearised models were derived. Any discrepancy between the theoretical and estimated trends can be investigated in the nonlinear model perhaps leading to further development. This method is only useful however if the experiments are over a

small range of flight conditions. Large excursion manoeuvres would render the linearised model invalid¹⁰.

The concept of validating one nonlinear model with several linear models representing different operating points has also been used by Klein¹⁷ in a method known as data partitioning. The method is especially useful where the flight experiment includes a wide range of flight conditions. It involves partitioning the experimental time response data of the variable representing flight condition, in this case angle of attack, to give several sections of flight data each containing data where the angle of attack varied by only a small amount even although the entire experiment taken as a whole involved a large variation in angle of attack. Sections from the same or different experiments but with the same range of angle of attack could be strung together to form several experimental flight records each of which involves only a small variation of angle of attack. The relevant aircraft parameters could then be estimated at each value of angle of attack using step-wise regression and plotted against angle of attack for comparison with the trends of the theoretical values.

A related validation method is the comparison of the model eigenvalues with peaks in the power spectrum of the experimental output data. The eigenvalues of a system represent the natural frequencies of the various dynamic modes of the system and can be easily calculated for a linear model. Fu and Kaletka²¹ use eigenvalue comparison to compare helicopter models with different levels of complexity and found the models to be consistent with each other for the lower frequencies and the higher frequency eigenvalues in the higher order models were where they had been expected.

2.5 Model Distortion

An approach to model validation pioneered by Butterfield and Thomas²² is model distortion. This technique is based on the assumption that model output should fit experimental data exactly. To allow this, the model is distorted just sufficiently for an exact fit. The amount of distortion required is measured and is used as a quantitative estimate of model validity or compared with a threshold distortion limit representing the model validation criterion. In practice, the model is distorted by varying the system parameters with time in such a way as to minimise the error between the original parameters and the optimised values while still satisfying the condition that model output must equal experimental data exactly.

Two implementations of the model distortion method are given by Butterfield and Thomas. The first is a more general time domain method based on nonlinear differential equations. The second is derived from the first and works for linear transfer function representations. The technique was developed in the nuclear power industry and published applications of the transfer function technique include Butterfield and Thomas²³ and Li²⁴. Applications of the time domain method are also given in Butterfield and Thomas²³.

Cameron²⁵ derives a simplified expression for the solution to the time domain model distortion for linear models and suggests that this technique is not particularly useful with noisy experimental data and is perhaps better suited to validating linearised models against full nonlinear representations. An example of this type of validation is given.

A more detailed explanation of the model distortion technique together with the theory and the results of an investigation into the time domain technique vis-a-vis helicopter applications is given in chapter 3 of this thesis. The conclusion of the research is that despite initial promise especially with regard to the parameter time histories produced, the model distortion method is not suitable for validation of nonlinear helicopter models with experimental flight data. The technique does not work with noisy data. This is also concluded by Cameron²⁵.

2.6 Inverse Simulation

The validation of mathematical models over a range of flight conditions wide enough to include Nap-of-the-Earth (NOE) manoeuvres represents a considerable problem. NOE manoeuvres (where the pilot follows ground terrain as closely as possible while manoeuvring as quickly as possible) involve aggressive control inputs across a wide range of operating points. This inevitably means that the aircraft flight path includes flight conditions where the model validity is not good and a computer simulation of the helicopter would start to diverge from the real system. If a pilot is flying a prescribed manoeuvre as opposed to applying a control input for investigation of the transient, the pilot is using feedback and is acting as a controller of the aircraft. In an open loop computer simulation of the system, this controller is not modelled and this will lead to errors in the simulation. Both of the above sources of error together with the large amplitude of the control inputs would lead to significant divergence of the

model output from flight data in the simulation output because of integrals ramping if a conventional validation technique based on system simulation was used.

One solution to this problem is inverse simulation²⁶. This is a technique developed by Thomson at Glasgow University which involves using the defined flight path as the input to an inverse simulation the output of which is the control time histories which can then be compared with the actual control inputs for model validation. The inverse model is derived from the nonlinear differential equations - in this case, the six degree of freedom helicopter model. The pilot is assumed to be a perfect controller with infinite feedback as this is the only way he could follow a prescribed flight path exactly. This assumption does not appear to compromise the technique.

Results with helicopter NOE manoeuvres including a side-step and quick-hop have been encouraging²⁶ and research into this method is continuing.

2.7 Open loop simulation

Many techniques of model validation are based on comparison of flight data with simulated model output for various vehicle dynamic states. If the complete model is driven by the control input time histories, it can sometimes be very difficult to determine the cause of any discrepancy between the flight data and the simulation output. This becomes more of a problem as the dynamic system under consideration becomes more complicated. The highly coupled nonlinear dynamics of the helicopter present particular difficulties.

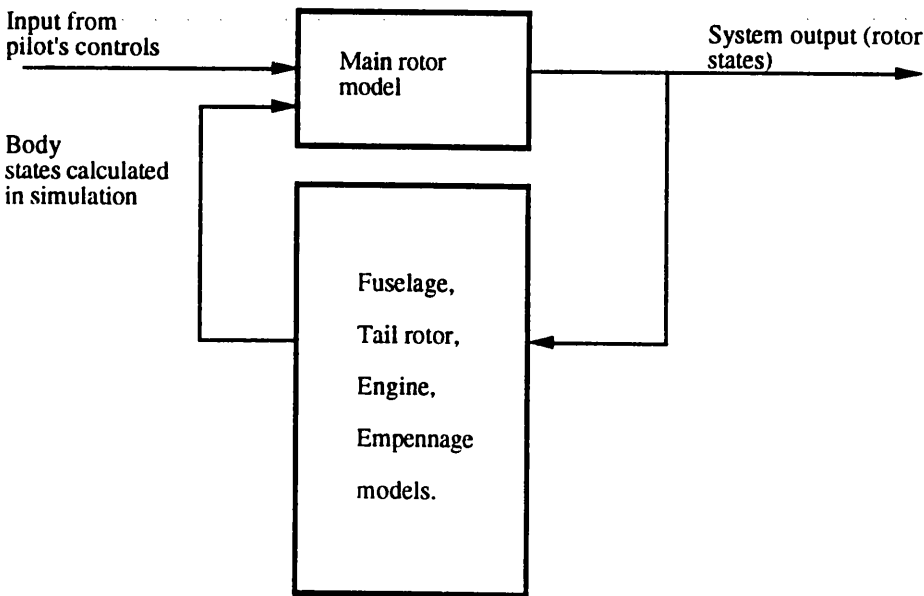
One possible solution is to model the subsystem which is being validated as a complete and independent system (eg. main rotor, tail rotor, engine). This presents problems when dealing with helicopter models because of the high degree of cross coupling - the dynamics of the subsystem are affected by the dynamics of the rest of the helicopter. Even where the aircraft is being actively maintained in one flight condition (eg. hover) by the pilot as happens in many flight experiments, there are small variations in the aircraft body states. For a subsystem model, this must be considered as process noise which limits the effectiveness of any model validation or development on the subsystem.

Padfield and DuVal²⁷ describe the conventional approach where a model of a complete system is driven by the control input time histories as closed loop simulation

where all the states are generated within the simulation (Figure 1a). Open loop simulation is defined as simulation of an independent subsystem using flight data to represent unknown states.

The open loop simulation approach involves using flight data records of state variables to drive the simulation of a subsystem (Figure 1b). This means that the aircraft motions are incorporated into the model and not considered process noise. The only source of process noise in this model will be the model of the subsystem itself, and any discrepancies between model output and flight data will be due to the subsystem model making validation and development much simpler. This technique has been used with helicopter rotor dynamics and compared with closed loop simulation with favourable results²⁸.

a) Closed loop simulation



b) Open loop simulation

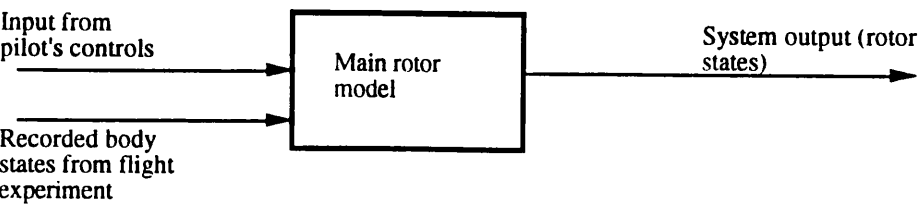


Figure 1 Comparison of open loop and closed loop simulation

2.8 Conclusions

Each of the above techniques represents a valuable model validation method. It is important that the correct technique for the particular problem is used. For example, inverse simulation is a good technique for validation of a model representing large excursion manoeuvres. These model validation methods should not however be regarded as mutually exclusive. It is often desirable to use more than one method on a particular problem. Different techniques can validate different parts of the model or act as a check on each other.

All of the techniques discussed in this chapter are statistical in nature. They could all be used simply by application to the set of equations representing the model being validated together with experimental data and examining and interpreting the results statistically. This would not be the most efficient use of the model validation technique. An essential component of model validation is physical knowledge of the system being modelled. Only then can the correct validation techniques be chosen and the results correctly interpreted. For a complex nonlinear dynamic system such as a helicopter, no mathematical model is going to be perfect. This means that the assumption of model validity will be subjective to some extent no matter which validation technique is used. That subjectivity depends on interpretation of the validation results in terms of the physical model and the ability to explain the results in physical terms. Physical knowledge of the dynamic system being modelled should be considered an essential component of the validation of nonlinear physical models.

References

- 1 deLeeuw, J.H. and Hui, K., '*The application of linear maximum likelihood estimation of aerodynamic derivatives for the Bell-205 and Bell-206*', Vertica, Vol. 13, No. 3, pp369-392, 1989.
- 2 Feuermann, D. and Zemel, A., '*Validation of models for global irradiance on inclined planes*', Solar Energy, Vol 48, no. 1, pp59-66, 1992.
- 3 Clapp, T.G. and Peng, H., '*Buckling of new fabrics. 3. Experimental validation of theoretical models*', Textile Research Journal, Vol 60, No. 11, pp641-645, 1990.

- 4 Padfield, G.D., '*A theoretical model of helicopter flight mechanics for application to piloted simulation*', RAE Technical report 81048, 1981.
- 5 Gawthorp, P.J. and Mirab, H. and Li, X., '*Robot model validation*', Transactions of the Institute of Measurement and Control, Vol. 12, No. 4, pp197-208, 1990.
- 6 Blackwell, J., Feik, R.A. and Perrin, R.H., '*Identification of rotor dynamic effects in flight data*', Fifteenth European Rotorcraft Forum, paper no. 68, 1989.
- 7 Ballin, M.G. and Dalang-Secrétan, M., '*Validation of the dynamic response of a blade element UH-60 simulation model in hovering flight*', Journal of the American Helicopter Society, Vol. 36, No.4, pp77-88, 1991.
- 8 Houston, S.S. and Black, C.G., '*On the identifiability of helicopter models incorporating higher order dynamics*', Fifteenth European Rotorcraft Forum, Paper 67, 1989.
- 9 Jategaonkar, R.V. and Plaetschke, E., '*Maximum likelihood estimation of parameters in nonlinear flight mechanics systems*', Seventh IFAC symposium on identification and system parameter estimation, York, pp167-172, 1985.
- 10 Padfield, G.D., Thorne, R., Murray-Smith, D.J., Black, C.G. and Caldwell, A.E., '*UK research into system identification for helicopter flight mechanics*', Vertica, Vol. 11, No. 4, pp665-684, 1987.
- 11 Mansur, M.H. and Tischler, M.B., '*Modeling methods for high-fidelity rotorcraft flight mechanics simulation*', Sixteenth European Rotorcraft Forum, Paper III.11.2, 1990.
- 12 Kaletka, J., Tischler, M., von Grünhagen, W. and Fletcher, J.W., '*Time and frequency domain identification and verification of BO 105 dynamic models*', Journal of the American Helicopter Society, Vol. 36, No. 4, pp25-38, 1991.
- 13 Iliff, K.W., '*Parameter estimation for flight vehicles*', Journal of Guidance and control, Vol. 12, No. 5, pp 609-622, 1989.
- 14 Beck, J.V. and Arnold, K.J., '*Parameter estimation in engineering and science*', John Wiley and sons, 1977.
- 15 '*Rotorcraft system identification*', AGARD lecture series 178, October 1991.
- 16 Leith, D.J., Bradley, R., and Murray-Smith, D.J., '*The identification of coupled flapping/inflow models for hovering flight*', Seventeenth European Rotorcraft Forum, Paper No. 91-45, 1991.

- 17 Klein, V., '*Estimation of aircraft aerodynamic parameters from flight data*', Progress in Aerospace Science, Vol. 26, No. 1, pp1-77, 1989.
- 18 Maine, R.E. and Iliff, K.W., '*Formulation and implementation of a practical algorithm for parameter estimation with process and measurement noise*', SIAM Journal of Applied Mathematics, Vol. 41, No. 3, pp558-579, 1981.
- 19 Jategaonkar, R.V. and Plaetschke, E., '*Identification of moderately nonlinear flight mechanics systems with additive process and measurement noise*', Journal Guidance, Control and Dynamics, Vol. 13, No. 2, pp277-285, 1990.
- 20 Lehman, S.L. and Stark, L.W., '*Three algorithms for interpreting models consisting of ordinary differential equations: sensitivity functions, sensitivity equations, global optimization*', Mathematical Biosciences, Vol. 62, pp107-122, 1982.
- 21 Fu, K-H and Kaletka, J., '*Frequency domain identification of BO 105 derivative models with rotor degrees of freedom*', Sixteenth European Rotorcraft Forum, paper III.10.3, 1990.
- 22 Butterfield, M.H., and Thomas, P.J., '*Methods of quantitative validation for dynamic simulation models. Part 1: Theory*', Transactions of the Institute of Measurement and Control, Vol 8, No. 4, pp182-200, 1986.
- 23 Butterfield, M.H., and Thomas, P.J., '*Methods of quantitative validation for dynamic simulation models. Part 2: Examples*', Transactions of the Institute of Measurement and Control, Vol 8, No. 4, pp201-219, 1986.
- 24 Li, C.L.R., '*Application of distortion technique for model validation to nuclear power plant- AGR scatter plug*', Transactions of the Institute of Measurement and Control, Vol 8, No. 4, pp220-234, 1986.
- 25 Cameron, R.G., '*Model validation by the distortion method: linear state space systems*', IEE Proceedings-D, Vol 139, No. 3, pp296-300, 1992. Cameron
- 26 Bradley, R., Padfield, G.D., Murray-Smith, D.J. and Thomson, D.G., '*Validation of helicopter mathematical models*', Transactions of the Institute of Measurement and Control, Vol. 12, No. 4, pp186-196, 1990.
- 27 Padfield, G.D. and DuVal, R.W., '*Application areas for rotorcraft system identification : Simulation model validation*', AGARD LS 178, October 1991.
- 28 McCallum, A.T., '*A study of helicopter rotor/fuselage response in low speed manoeuvres: comparison of theory with flight*', Seventeenth Rotorcraft Forum, paper no. 91-48, 1991.

Chapter 3

Sensitivity coefficients

3.1 Introduction

Most of the search techniques used to minimise the cost function in the parameter estimation process (chapter 2) require the derivative of the model output with respect to the parameters being estimated. This derivative is known as the sensitivity coefficient. Sensitivity coefficients can also provide information of direct value with model validation and development process. Single value sensitivity coefficients can be used for validation and for parameter selection prior to estimation. Sensitivity coefficient time histories can be used for model development when examined in context with a parameter estimation process.

3.2 Calculation of sensitivity coefficients for nonlinear systems

With continuous well-behaved functions, a finite difference algorithm can be used to determine the derivative or sensitivity coefficient. This method is the most commonly used for well-behaved nonlinear models. Nonlinear mathematical models based on physical equations however, often contain discontinuous nonlinearities such as a saturation function, a deadband or hysteresis. Theoretically the properties of these nonlinearities can be estimated using parameter estimation techniques. For discontinuous nonlinearities however, a finite difference technique has an amplitude effect similar to filtering the function with a low pass filter with a cut-off frequency which is dependent on the size of the difference. For example, figure 2 illustrates the effect of a finite difference algorithm on a signum function. That is, a function which returns $+a$ for a positive input and $-a$ for a negative input where a is a constant. This function is often used to represent friction. To generate a sensitivity function, it is necessary to differentiate the output of each component with respect to its input (the input could be a function of the parameter). The true derivative is zero with an impulse function where the input is zero. The result of a finite difference approximation to the derivative is a pulse, the shape of which is decided by the value of the difference, δ . As δ tends to zero, the derivative of the function tends to the correct shape, but where δ is not zero, and the input to the nonlinear element approaches zero, the derivative will be inaccurate and serious convergence problems could arise in the minimisation stage of the parameter estimation process.

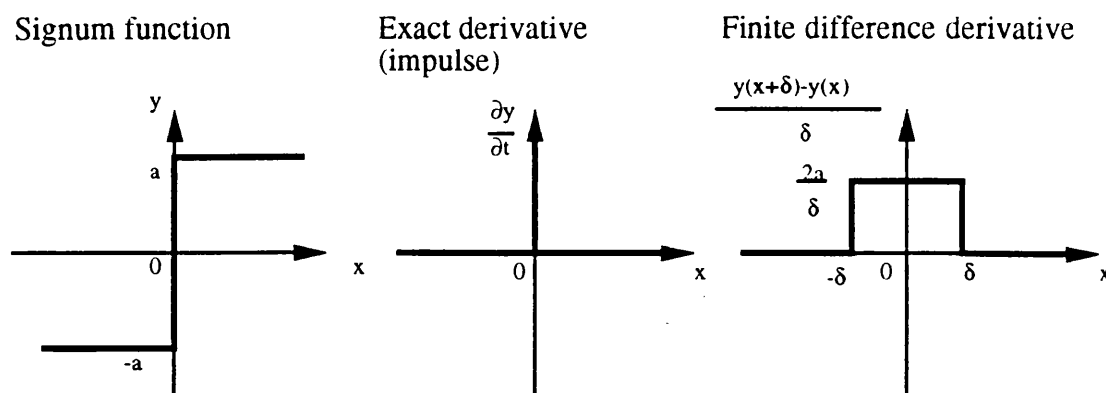


Figure 2 Methods of partial differentiation of signum function

One solution is to use a search method which does not require derivatives. A direct search algorithm is used by Jategaonkar¹ to minimise the maximum likelihood cost function for a dynamic system with discontinuous nonlinearities. The technique is used to estimate a deadband nonlinearity on a control surface actuator of a research aircraft using real flight data. Such search algorithms are often slower than derivative based methods. A comparison by Jategaonkar and Plaetschke² of a direct search algorithm with a quasi-linearised derivative based method showed the direct search method to be considerably slower.

The alternative is to use sensitivity equations derived by analytically differentiating the main system equations. These sensitivity equations form a separate dynamic system which must be simulated alongside the main system. This is best done manually since computational algebra packages tend to produce equations which are very long-winded and while they are useful for checking the manual differentiation, they would add a considerable computation time overhead to the parameter estimation process. This is because the sensitivity cosystem must be simulated meaning that each of the sensitivity equations must be evaluated at each integration step in the system simulation. The sensitivity coefficients generated by the computer algebra package, although algebraically identical to the real equations, contain many more terms than equations calculated by manual differentiation and therefore take longer to evaluate on a computer. The use of sensitivity equations can be defined as the cosystem approach³ because the sensitivity equations form a dynamic system which runs in parallel with the real system and is driven by the dynamic variables from the real system (figure 3).

The sensitivity cosystem approach allows inclusion of discontinuous nonlinearities using a method derived by Vuskovic and Ciric⁴. The cosystem is generated on a function by function basis using partial differentiation. Given the original system with linear and nonlinear functions, some of which are dependent on a parameter q , calculate the sensitivity coefficients of the model output with respect to the parameter q , by differentiating the system equations.

Linear elements of a system are represented by,

$$y_i = \int_0^t w_i(t, \tau) x_i(\tau) d\tau \quad (3.1),$$

and nonlinear elements by,

$$z_k = f_k(\epsilon_k(t)) \quad (3.2).$$

The cosystem expression is derived by differentiating the real system with respect to the parameter q . Where the function w of the linear element is not dependent on q ,

$$\frac{\partial y_i}{\partial q} = \int_0^t w_i(t, \tau) \frac{\partial x_i(\tau)}{\partial q} d\tau \quad (3.3).$$

For linear elements where the function w is dependent on q ,

$$\frac{\partial y_i}{\partial q} = \int_0^t \frac{\partial w_i(t, \tau)}{\partial q} x_i(\tau) d\tau + \int_0^t w_i(t, \tau) \frac{\partial x_i(\tau)}{\partial q} d\tau \quad (3.4).$$

For nonlinear elements,

$$\frac{\partial z_k}{\partial q} = \lambda_k(t) \frac{\partial \epsilon_k}{\partial q}, \quad \lambda_k = \frac{df_k(\epsilon_k)}{d\epsilon_k} \quad (3.5).$$

This is shown digrammatically in figure 3 with a system including two functions - one linear and one nonlinear, both of which are dependent on parameter q .

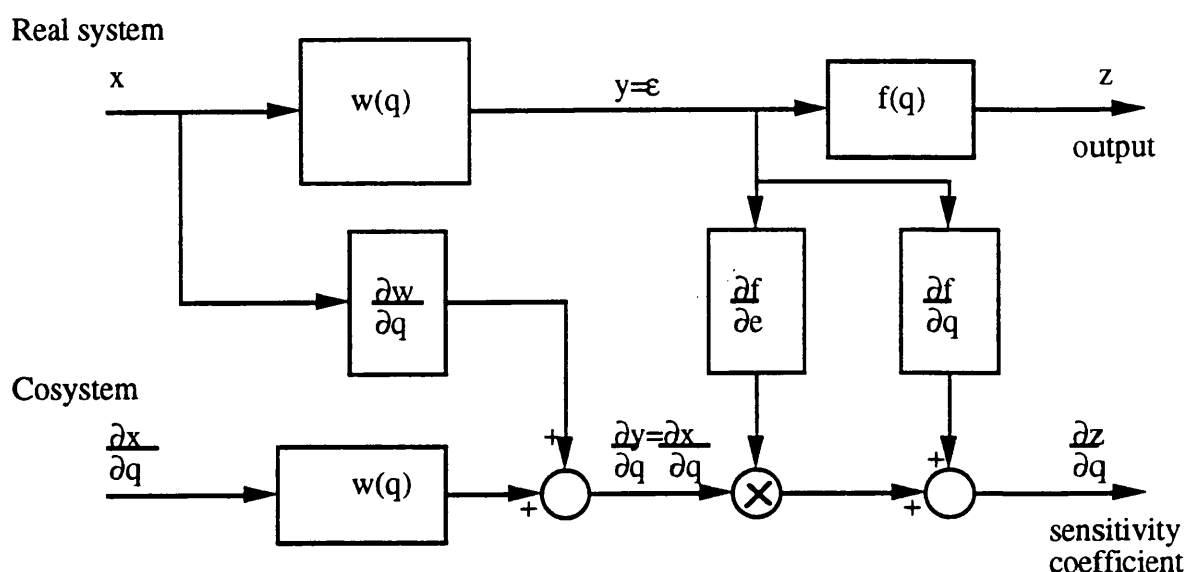


Figure 3 Block diagram of dynamic system and sensitivity cosystem

In practice, a discontinuous nonlinear element is replaced by another discontinuous nonlinear element in the cosystem whose input is the output state of the nonlinear element in the real system. Cosystem outputs are exact derivatives of the model outputs and can therefore be used in the search algorithm as part of the parameter estimation process without the problem of selecting a value for a finite difference and therefore reducing the likelihood of numerical computational problems.

An example of the use of a cosystem is the estimation of a pure time delay in the time domain. Time delays are relatively easy to estimate with linear models in the frequency domain using correlation techniques as used Kaletka, Tischler, von Grünhagen and Fletcher⁵ with helicopter data but with a nonlinear model, it is necessary to work in the time domain. Pure time delays occur in flight data because of delays in data acquisition systems on board the aircraft and also different sampling rates for different variables. Time delays can also be used to approximate dynamic features such as first order lags in control actuators. Results of the time delay estimation must therefore be interpreted with care. The time delay can be estimated by shifting the output in the time domain by the relevant amount and using a finite difference equation to calculate the sensitivity coefficient numerically. This is the method used by Blackwell and Feik⁶. Time delays and other parameters are successfully estimated in a nonlinear model using a maximum likelihood technique. Finite differences are used to approximate the derivatives used in the search in the cost function optimisation. Several examples are given including a fixed wing aeroplane

model incorporating a time delay and a model of an aircraft landing gear with a breakpoint nonlinearity. The importance of the size of difference for the finite difference calculation for a nonlinear model is emphasised in the paper. This could present problems in a more complicated system with other nonlinearities.

The cosystem approach to time delay estimation involves analytically differentiating the time delay function with respect to the delay represented here by parameter p .

Given,

$$y(t) = x(t-p) \quad (3.6),$$

where p is the time delay, the sensitivity coefficient with respect to p is,

$$\begin{aligned} \frac{\partial y}{\partial p} &= \lim_{\delta \rightarrow 0} \frac{x(t-p+\delta) - x(t-p)}{\delta} \\ &= \lim_{\delta \rightarrow 0} \frac{x(t+\delta-p) - x(t-p)}{\delta} \\ &= - \frac{\partial x(t-p)}{\partial t} \\ &= - \frac{\partial y}{\partial t} \end{aligned} \quad (3.7).$$

That is, the sensitivity coefficient for a time delay whose parameter is the size of the delay is the negative of the time derivative of the real system time delay function output. Parameter estimation requires simulation of the model and in a simulation package it is usually quite easy to obtain time derivatives of dynamic variables.

3.3 Single value sensitivity coefficients

As well as their use in parameter estimation search algorithms, sensitivity coefficients can provide useful parametric information for model validation and development. Single value sensitivity coefficients are used by Lehman and Stark⁷ to indicate the importance of individual parameters for particular outputs. It is suggested that sensitivity coefficients could be tabulated as single values for all the possible parameters and all possible outputs so that the relative effect of changing any parameters on any particular outputs could be seen at a glance.

A single value sensitivity coefficient representing the effect of a particular parameter on a particular output can be very useful in parameter selection prior to parameter estimation and can give valuable information about the model and which parts of the model are most relevant for the application or experiment being considered (calculation of any sensitivity coefficient requires a simulation run and is therefore dependent on the chosen input function). Lehman and Stark⁷ also use the sensitivity coefficients to validate a linearised model of eye movement against the nonlinear model. The values of the sensitivity coefficients are compared and if the match is good, the linearised model is considered a valid representation. This validation method also provides information regarding development areas in the linearised model if the model is found not to be valid.

3.3.1 Experimental application of single value sensitivity coefficients

The method of single value sensitivity coefficients was used to select the parameters having the largest effect on model output so that those parameters could be used in the model distortion technique in chapter 4.

From Lehman and Stark⁷, the single value sensitivity coefficient,

$$S_{ij} = \frac{db_i/b_{i0}}{dp_j/p_{j0}}$$

where p and b represent the parameter and system output and p_{j0} and b_{i0} are the nominal values of the parameter and model output.

This still leaves the problem of how to generate the sensitivity coefficients, and more specifically, how is b_i defined.

For each experiment, there is a time response of each sensitivity coefficient, Sixty-four parameters and eight outputs gives five hundred and twelve sensitivity coefficients. Each time response has to be reduced to a single value. There are many ways of doing this but the problem is complicated by the fact that the system is unstable. This means that some of the sensitivity coefficients will be unstable and will

increase exponentially with time. Various possible methods of calculating the sensitivity coefficients were postulated and tested experimentally with a linear helicopter model.

The experiments were based on an eight state model of a PUMA helicopter in the hover with a collective step as an input.

Possible values to take for sensitivity coefficients were maximum sensitivity over a given time span (absolute and squared), sum over a given time span (absolute and squared) and value of sensitivity at some arbitrary time point. The results from maximum absolute, sum of absolute and arbitrary time sample were very similar. The results from the squared categories were as expected i.e. the results of the absolute categories with the larger peaks emphasised.

It was decided that a squared category would be better for this application because a smaller number of parameters was desirable. The sensitivities were calculated and plotted in two formats. One consisted of eight plots (one for each output) each of sixty-four points (one for each parameter). This showed which parameters would have the greatest effect on output if they were altered. The other format was sixty-four plots of eight points each. This showed which outputs were most affected by a change in each parameter.

It was hoped that this method could be used to identify a parameter of the six degree of freedom helicopter model with high sensitivity for the investigation of the model distortion model validation technique (chapter 4). This was therefore chosen as an example to investigate the sensitivity coefficient generation method.

For a collective step in the six degree of freedom linear model, the sensitivity of vertical velocity, w , to each of the coefficients in the A matrix is,

columns 1 to 4

0.0093	0.6582	0.1217	0
0.1666	107.4824	0.5590	0
0.0264	0.0222	1.2125	1.7867
1.0096	3.5846	0	0
0.0200	0.1454	0.6225	0
0.0060	0.0108	0.1752	0
0	2.5686	0	0
0	0	0	2.8105

columns 5 to 8

0.0030	0.1415	0	0.0083
0.0022	0.1846	0	0.0425
0.0076	0.1026	0.0026	0.0149
0.0030	0	0	0
0.0281	0.2774	0	0.0005
0.0242	0.4704	0.6466	0.0602
0.9652	0	0	0
0.1740	0.0355	1.5744	0.2950

The sensitivity information was also generated in the graphical format (figure 4).

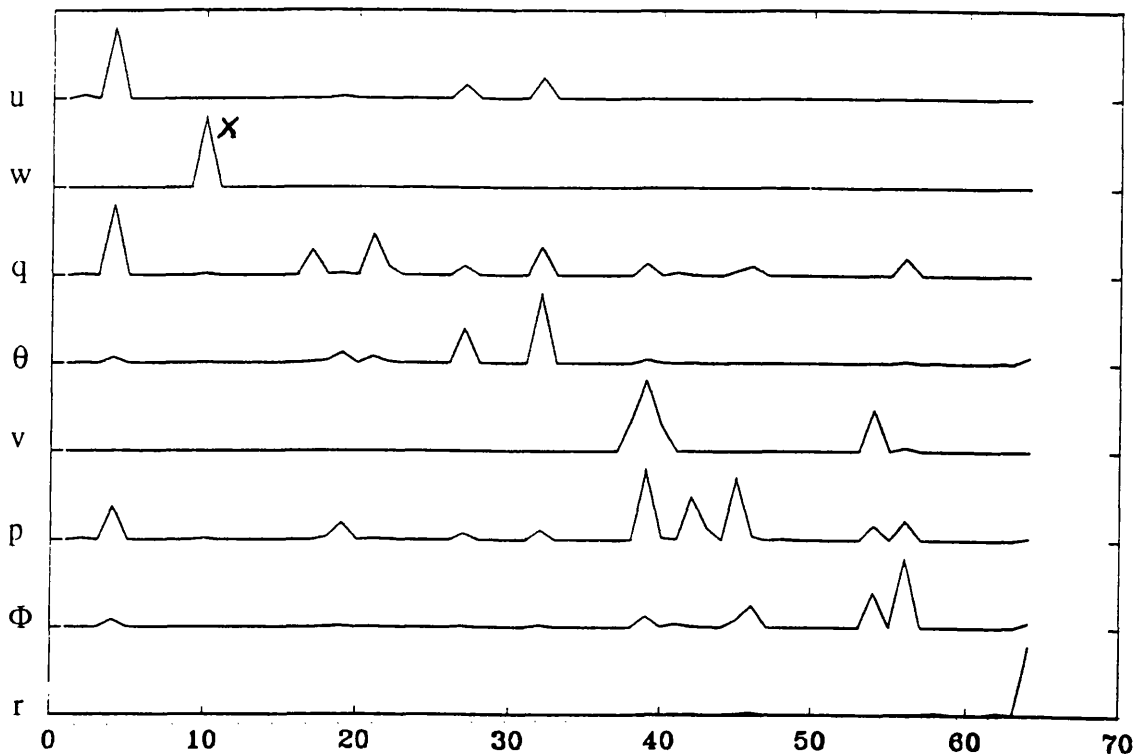


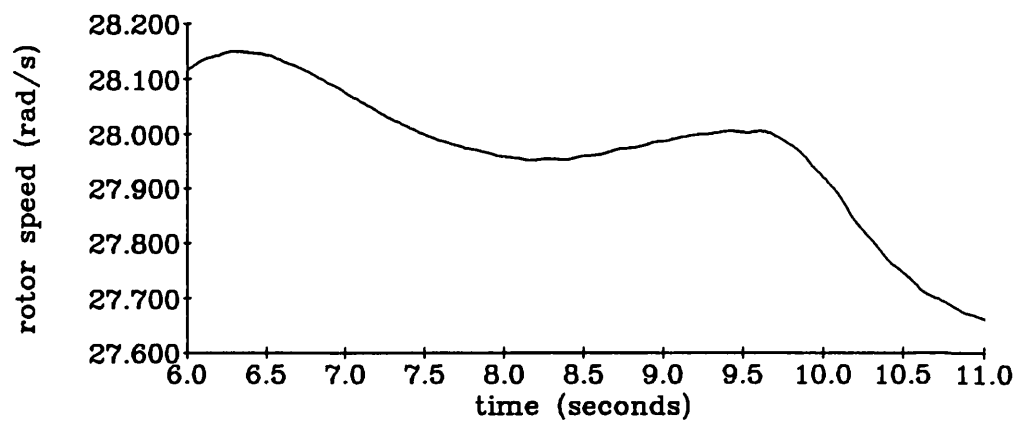
Figure 4 Sensitivity coefficients for vertical velocity output for each of the sixty-four parameters in the system A matrix (8 state linear helicopter model) for collective step input

The letters down the left hand side of the graph represent the eight outputs. These are u , w , q , θ , v , p , Φ and r , (forward velocity, vertical velocity, pitch rate, pitch angle, lateral velocity, roll rate, roll angle and yaw rate respectively). The maximum sensitivity coefficient for each output is normalised to one. This means that different outputs cannot be compared directly on this graph. The numbers along the bottom of the graph represent the 64 parameters in the A matrix of the helicopter model. Peak X is at aerodynamic coefficient number 10 on the second row (w output). This corresponds to the aerodynamic coefficient Z_w . Z_w is the parameter with the largest sensitivity coefficient and therefore was the parameter chosen for the investigation of the model distortion technique (chapter 4).

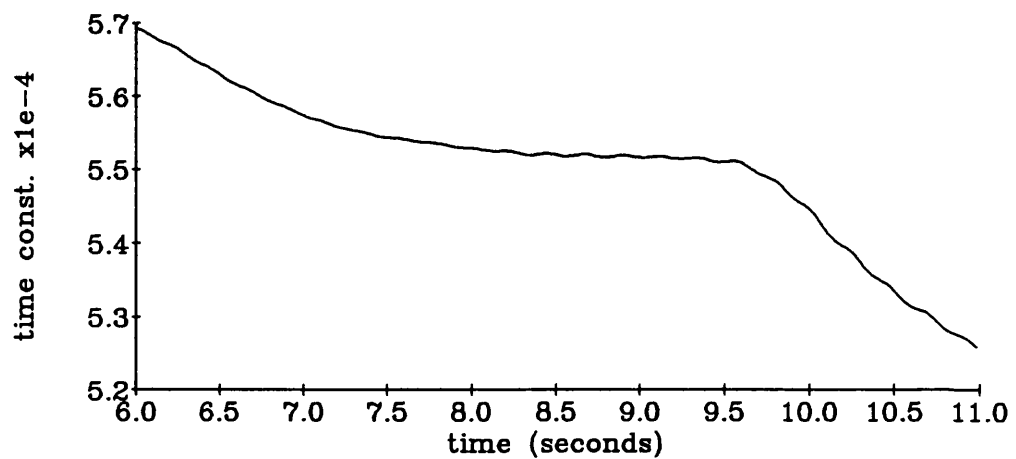
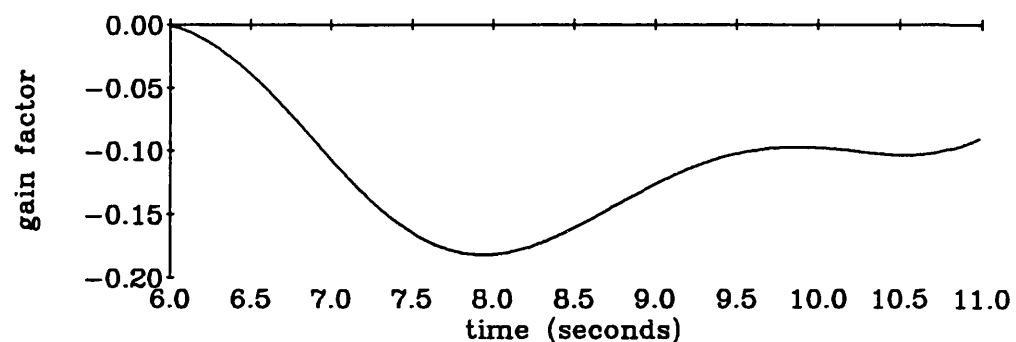
3.4 Sensitivity coefficient time histories

Time response plots of the sensitivity coefficients are a very useful tool for model analysis, validation and development. Lehman and Stark⁷ use time histories of

sensitivity coefficients to provide clues as to the model dynamics and the effects of the different parameters on the dynamics. This is particularly useful with the nonlinear model because otherwise it can be very difficult to obtain this information. When compared with the model time response and the experimental data, the effect on the model output of varying the parameter can be seen clearly. If, for example, one part of the model time response is a bad fit to flight data, like an initial transient for example, it is possible that examination of the sensitivity coefficients will reveal a parameter with increased sensitivity at that point indicating that the model could perhaps be improved by altering this parameter. This is illustrated by an example from chapter 5 of this thesis in figure 5. The model output is helicopter rotor speed and the parameters under consideration are engine controller gain and time constant. The part of the time response shown is the transient after a step input. The first six seconds of the record (not shown) represent trimmed flight. It can be seen that improving the fit on the second peak in the time response would involve adjusting the time constant and modelling the controller with a gain alone would be inadequate.



System time response



Sensitivity coefficient time histories

Figure 5 Section of engine model time response to main rotor collective step with sensitivity coefficients for engine parameters (see chapter 6)

Time history plots of sensitivity coefficients are also useful for detecting any linear dependence between parameters. That is where the effect of altering one parameter is very similar to the effect of altering another. Linear dependence causes numerical problems in the minimisation part of the parameter estimation process and should be avoided. Linear dependence can be spotted easily from graphical plots of sensitivity coefficients as two of the plots having very similar shapes⁸.

3.5 Discussion

For nonlinear systems, derivation of analytic sensitivity coefficients is preferable to a finite difference approximation. With severe nonlinearities, selection of the difference becomes very important and if sensitivity coefficients are not accurate representations, convergence problems can result in any parameter estimation procedures. Analytic differentiation using the cosystem approach gives exact differentials which can be used with confidence in parameter estimation or for model validation.

This method of single value sensitivity coefficients is very useful for selecting parameters most likely to have the largest effect on the model output and has possible applications in many parametric model validation methods. It is most useful for linear systems however, as removing parameters from consideration in nonlinear models based solely on this sensitivity information requires care.

The use of single value sensitivity coefficients for validation should be considered for inclusion in any model validation procedure where one model is being validated against another. For example, when a linearised model is being validated with a full nonlinear model. Sensitivity coefficient time histories give valuable insight into a validation problem especially regarding the effects of individual parameters. They should be considered an integral part of any parametric model validation method.

References

- 1 Jategaonkar, R.V., '*Parametric identification of discontinuous nonlinearities*', Seventh IFAC symposium on identification and system parameter estimation, York, pp167-172, 1985.

- 2 Jategaonkar, R.V. and Plaetschke, E., '*Maximum likelihood estimation of parameters in nonlinear flight mechanics systems*', Seventh IFAC symposium on identification and system parameter estimation, York, pp167-172, 1985.
- 3 Murray-Smith, D.J., '*Parameter sensitivity analysis in system modelling*', Departmental report CSG/70/001, Dept. Electronics and Electrical Engineering, University of Glasgow, 1970.
- 4 Vuskovic, M. and Ciric, V., '*Structural rules for the determination of sensitivity functions of nonlinear stationary systems*', in Radonovic, L. (ed.), '*Sensitivity methods in control theory*', pp154-165, Pergamon press, 1964.
- 5 Kaletka, J., Tischler, M., von Grünhagen, W. and Fletcher, J.W., '*Time and frequency domain identification and verification of BO 105 dynamic models*', Journal of the American Helicopter Society, Vol. 36, No. 4, 1991.
- 6 Blackwell, J. and Feik, R.A., '*Identification of time delays in flight measurements*', Journal of Guidance, Control and Dynamics, Vol. 14, No. 1, pp132-139, 1991.
- 7 Lehman, S.L. and Stark, L.W., '*Three algorithms for interpreting models consisting of ordinary differential equations: sensitivity functions, sensitivity equations, global optimization*', Mathematical Biosciences, Vol. 62, pp107-122, 1982.
- 8 Beck, J.V. and Arnold, K.J., '*Parameter estimation in engineering and science*', Chapter 7, pp349-351, John Wiley and sons, 1977.

Chapter 4

Investigation of Model Distortion Technique

4.1 Introduction

In 1986, Butterfield and Thomas¹ proposed a model validation technique based on distorting a nonlinear model by varying its parameters in order to achieve an exact fit of model output to experimental data. This model distortion method is based on the premise that 'any model can be made to follow any observed transient by introducing enough distortion: the less the required distortion, the better the model'. This distortion can be measured and can provide a quantitative criterion of validity of the model. If the required distortion is within acceptable limits with respect to approximations made in the mathematical model, the model can be considered a valid representation of the system.

This model distortion technique was developed for use in the modelling of nuclear power plant. The nuclear industry is a field where safety is of paramount importance and therefore where model validity is critical. The technique involves varying the system parameters in time in such a way that the model output fits the experimental data perfectly. This produces a time history for each parameter from which a parameter variance can be calculated. In a model derived from physical principles, this variance can be compared with the uncertainty in the value of the physical quantity.

In use, given that the mathematical model to be validated consists of a set of nonlinear differential equations, the set of equations is solved for any unknown parameters at each time point independently. Unmeasured states are considered parameters and their time histories are also estimated. Usually, there are more parameters than equations and the problem is overdetermined. This means that the solution takes the form of a constrained optimisation of some cost function where the constraint is the model output equals experimental data.

Results of two applications of this time domain model distortion technique have been published. Butterfield and Thomas² illustrate the method using a simple version of the method for two parameters in a first order linearised model of a hydraulic oil supply system. A small number of points is used but the resulting parameter time history is oscillatory unlike the system time response. A rolling average is used to smooth the

parameter time history producing a peak at the start of the time history followed by a steady state constant value. It is suggested that this parameter time variation indicates a deficiency in the model and that perhaps the parameter should be time varying thereby illustrating the potential of the model distortion technique for model development.

Cameron³ derives explicit formulas with some assumptions for the necessary distortion for application of the model distortion technique to linear state space systems. This reduces the considerable amount of computation inherent in the model distortion technique. The example given is the validation of a linearised model against the nonlinear model from which it was derived. A small number of time points is given and the variance of the resulting parameter time history is used as a quantitative measure of model validity. The resulting time histories are oscillatory despite the almost first order appearance of the system dynamic time response. It is proposed by Cameron³ that this technique is of more use with applications such as validating a linearised model against the full nonlinear model rather than with experimental data since it is suggested that the model distortion technique is only of use in cases where the output data is noise free. This paper³ was published in May, 1992, after the completion of the research described in this chapter, and it confirms the conclusion of this work regarding noise. Cameron's paper³ also suggests that the oscillatory nature of the parameter time responses detailed in this chapter was not a peculiarity of that particular example.

4.2 Application of model distortion technique to helicopter dynamics

This technique was specifically designed for nonlinear mathematical models developed from physical principles. The physical parameters of such a model have uncertainties associated with their values. These uncertainties can be compared with the variance of the parameter time histories produced by the model distortion technique. It was realised that such a technique could have applications in nonlinear helicopter model validation.

The resulting parameter time histories also had possibilities where flight experiments involved large excursion manoeuvres with consummate large variations in flight condition or operating point. The resultant parameter time histories could perhaps be compared with the expected variations of the parameters for that manoeuvre through these operating points.

It was concluded that it would be useful to evaluate this model distortion technique for applications to validation of nonlinear helicopter models. Previously, the technique had been used with simpler models of processes in a nuclear power plant. A helicopter presents significantly different problems. Specifically, a larger range of frequencies, more dynamic variables or states, and considerable high frequency noise.

4.3 Theory (from Butterfield and Thomas¹)

The model is described as a set of nonlinear differential equations of the form,

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

where,

x = state vector (vector of system dynamic variables),

p = vector of system parameters,

u = experimental input signal.

Normally, this equation would be integrated over time to give a time response of the model in response to the measured experimental input. This can then be compared with the experimental output data. In the model distortion technique however, equation 4.1 is solved at each time point for the parameter vector p . It is therefore necessary to differentiate the flight data to get the left hand side of equation 4.1. Any states which are not measured and cannot be derived by differentiation of experimental data are defined as parameters for the first run of the optimisation and their time history is used as experimental data in subsequent optimisations to determine the parameters in the rest of the model. Usually there are more parameters than equations - the problem is overdetermined. So it is necessary to postulate a cost function,

$$\text{Cost} = \frac{(p_i - p_0)^2}{w^2}$$

where,

p_i = parameters to be identified,

p_0 = a priori estimates of parameter values,

w = parameter weighting factors based on nominal parameter value,

and minimise C with the constraint that model output equals experimental data exactly. This constrained optimisation problem can be solved using Lagrange multipliers⁴.

The problem is to minimise C , where,

$$C = \sum_{i=1}^k (p_i - p_0)^2 \quad (4.2)$$

subject to

$$\phi_i = z_i - \dot{x}_i = 0 \quad i = 1, q.$$

Derive function,

$$F = C + \sum_{i=1}^q \lambda_i \phi_i,$$

and solve,

$$\begin{aligned} \frac{\partial F}{\partial p_i} &= 0 \quad i=1, k \\ \phi_i &= 0 \quad i=1, q. \end{aligned}$$

$$F = \sum_{i=1}^k (p_i - p_0)^2 + \sum_{j=1}^q \lambda_j (z_i - \dot{x}_j)$$

therefore,

$$\frac{\partial F}{\partial p_i} = 2(p_i - p_0) + \sum_{j=1}^q \lambda_j \left(\frac{\partial \dot{x}_j}{\partial p_i} \right) = 0 \quad i=1, k \quad (4.3).$$

where,

- C = cost function of unknown parameters,
- k = number of parameters including unknown states,
- p_i = normalised parameter value,
- p_0 = a priori estimate of parameter (normalised),
- z_i = experimental flight data for equation 4.3,
- x_i = model output for equation 4.2,
- q = number of dynamic variables for which flight data is available
- Φ_i = residual (flight data - model output) for equation i ,
- F = Lagrange function,
- λ = Parameter used in method of Lagrange multipliers.

Equation 4.3 is solved at each time point to give p , and when the parameter vector p has been calculated at every time point, the parameter time histories can be constructed.

The resulting parameter time histories can be analysed to determine the validity of the model. The variance of the parameter over the length of the experiment can be calculated and compared with the expected variance i.e. the confidence of that parameter in this particular model. If it is less than expected variance for each parameter then the model can be said to be valid for this particular experiment. The shape of the parameter time history could also reveal information about the model. For example, if the flight condition changes significantly during the experiment and a parameter value changes at the same point.

The model distortion algorithm is a constrained minimisation. The method used to solve this was Lagrange multipliers. This reduces the problem to solving a set of nonlinear algebraic equations. This must be done at each point in the time history, typically 400 points in these experiments. PRO-MATLAB was the computer package chosen to solve this problem because of the ease of manipulating data and presenting results graphically. There was also a routine available called 'fsolve'⁵ in PRO-MATLAB which was ideal for solving the equations. It solves systems of nonlinear equations using Newton's method with a few modifications including a linear search to ensure global convergence. The computer used was a DEC VAXstation 2000.

4.4 Experimental Analysis

4.4.1 Second order system

In order to gain a basic understanding of the method, initially, the model distortion technique would be tested with a simple dynamic system - a linear second order system with two parameters, natural frequency (ω), and damping factor (ξ). These two parameters would be estimated using the model distortion technique with initial estimates of p_0 slightly offset from the correct parameter values in order to investigate the effect of these starting values. The noise response of the method would also be tested by adding white noise to the simulated data and to the input signal.

This second order equation was modelled and simulated to give simulated experimental data. The model can be described as a transfer function thus,

$$\frac{Y(s)}{U(s)} = \frac{\omega^2}{s^2 + 2\xi\omega s + \omega^2}. \quad (4.4)$$

The parameters were chosen arbitrarily at $\omega=0.3$ and $\xi=2$ to give a good response for the time period being considered. The mathematical model of the system was coded in differential equation form thus,

$$\dot{x}_1 = x_2, \quad (4.5)$$

$$\dot{x}_2 = -2\xi\omega x_2 - \omega^2 x_1 + \omega^2 u \quad (4.6).$$

A step function was used as the input (u) and simulated data was generated for x_2 and x_1 . The simulated data is shown in figure 6.

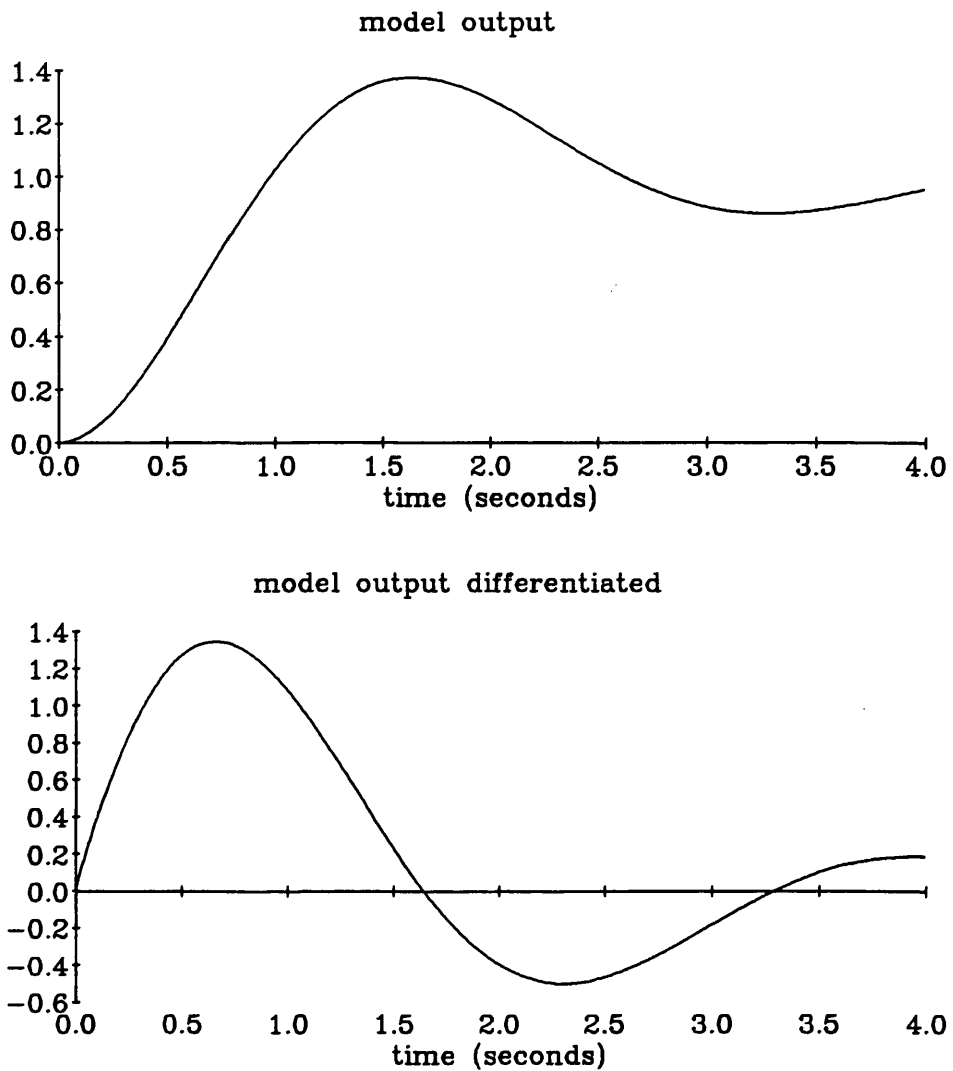


Figure 6 Time response of second order system

The x_2 data was differentiated with respect to time and the model distortion technique was applied to equation 4.6 with the intention of obtaining parameter time histories for natural frequency and damping factor. A priori estimates for the cost function were given as $\omega_0=2.1$, and $\xi_0=0.27$.

Since simulated data was being used, it would not be unreasonable to expect the experimental output to show constant values for the two parameters ω and ξ throughout the time history of the experiment. The resulting time histories in fact showed ω and ξ to be oscillatory (figure 7).

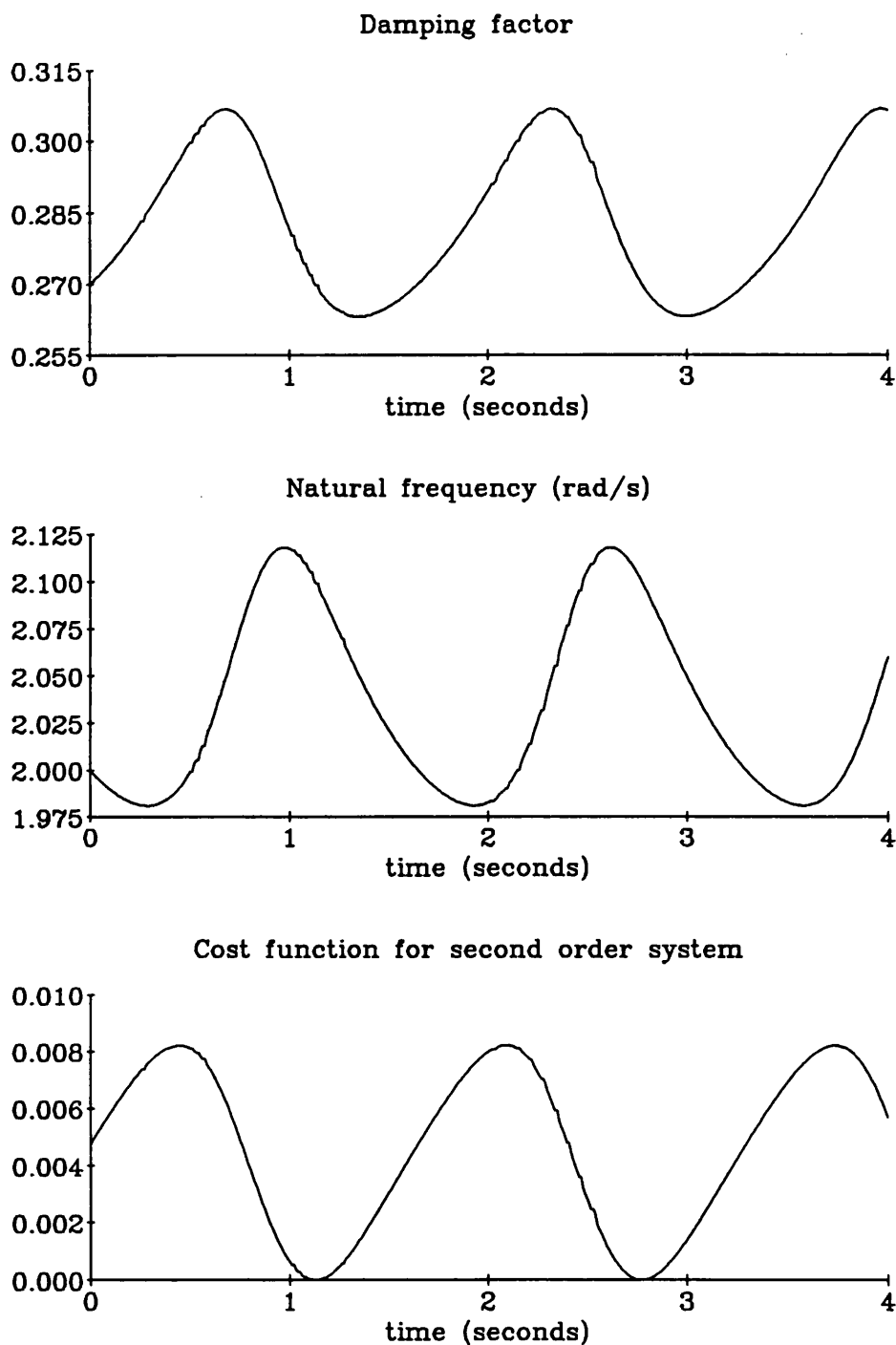


Figure 7 Parameter time histories for second order system with simulated data

The parameters oscillated between the correct parameter value (known because this was simulated data) and the parameter estimate used in the optimisation. The frequency of the oscillation appeared to be the same as the frequency of the time history of the experiment. This was verified by running the experiment using a

different value of natural frequency for the simulated system. The period of the oscillations in the resulting parameter time history changed accordingly. The oscillations were always present - if the initial guess values were the correct value for the system, the parameter time histories showed the same shape of oscillation but with a very small amplitude. The cost function (figure 7) showed that when the cost function was at a maximum (it was also periodic), the values of ω and ξ from the time histories were correct. These points corresponded to the turning points of the original time history suggesting that at these points, only the time response of exactly the same dynamic system would have the same amplitude after the same time since the turning points represent the 'limits' of the time response curve. Other points on the time response curve could occur in the responses of other second order systems with different values for the parameters. The points 'chosen' for the resulting time histories represent those points which minimised the cost function in the model distortion algorithm.

The experiment was verified by simulating the second order system with the time varying parameters from figure 7 and the same input. The output is shown in figure 8 and is identical to that in figure 6 as expected.

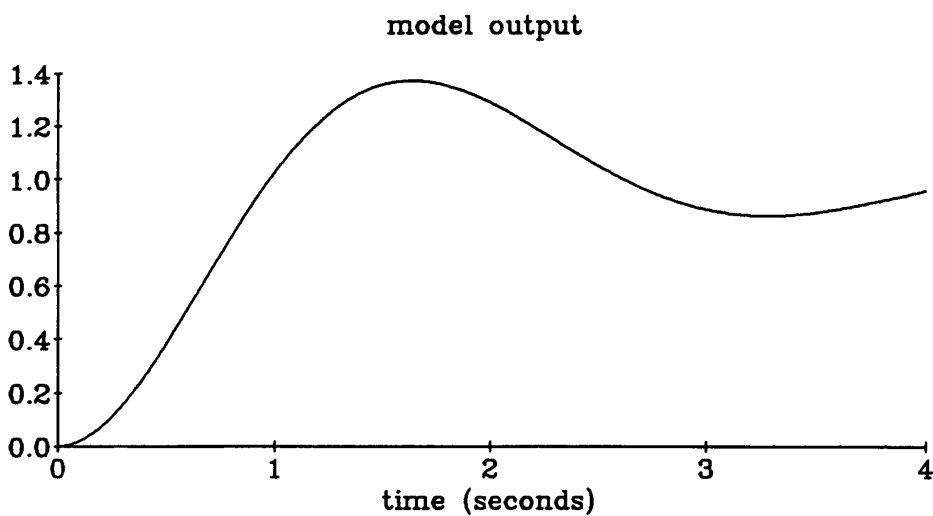


Figure 8 Time response of second order system with time varying parameters

Oscillations were also evident in the results from the two published applications of the time domain model distortion method^{2,3}. In each of these cases, the number of points was too low to discern if there was any regular shape to the periodicity.

4.4.2 Noise response

The noise response of any model validation algorithm is very important. The model perturbation algorithm was tested with the second order system and two types of additive noise. On the input signal and as measurement noise on the simulated data.

White noise was added to the input signal before it was used to generate the simulated data. The resulting parameter time histories (figure 9) were similar to those for a noise free signal but only for the first two or three oscillations (about five seconds). After this point the signal became very noisy. At this point the time response is settling down and the differential of the time response which is the signal that the optimisation is using is tending to zero. This becomes more apparent as the transient decays to zero in the steady state. This means that in the constraint equation (equation 4.6), the input noise has a much larger affect than the dynamic response. The parameter time histories therefore reflect the noise more than the response. The stationary points where there is only one solution to the constraint equation are still evident however as areas in the time history (figure 9) where the noise amplitude is at a minimum. The increasing noise amplitude (most obvious in the cost function) as time increases suggests that this technique should only be used with dynamic response and not steady state experimental data.

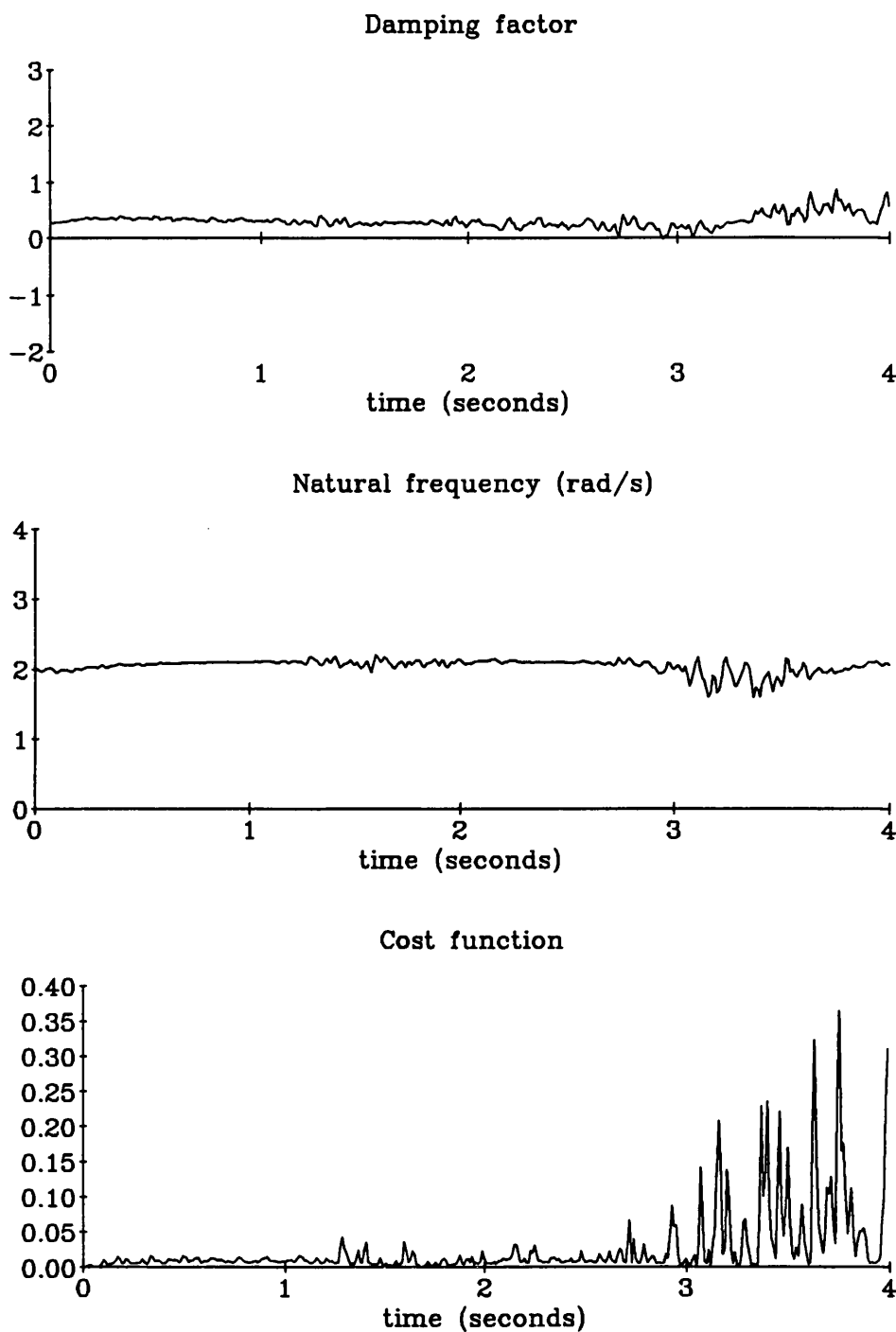


Figure 9 Parameter time histories for second order system with white noise added to the input signal

Simulated measurement noise at a signal to noise ratio of 0.5% was also added to the simulated data before it was differentiated. It is important to test the robustness of any validation technique with respect to measurement noise - especially in aerospace

applications. This measurement noise is more realistic since real experimental data will be contaminated with measurement noise. The results showed a noisy periodic time history (figure 10).

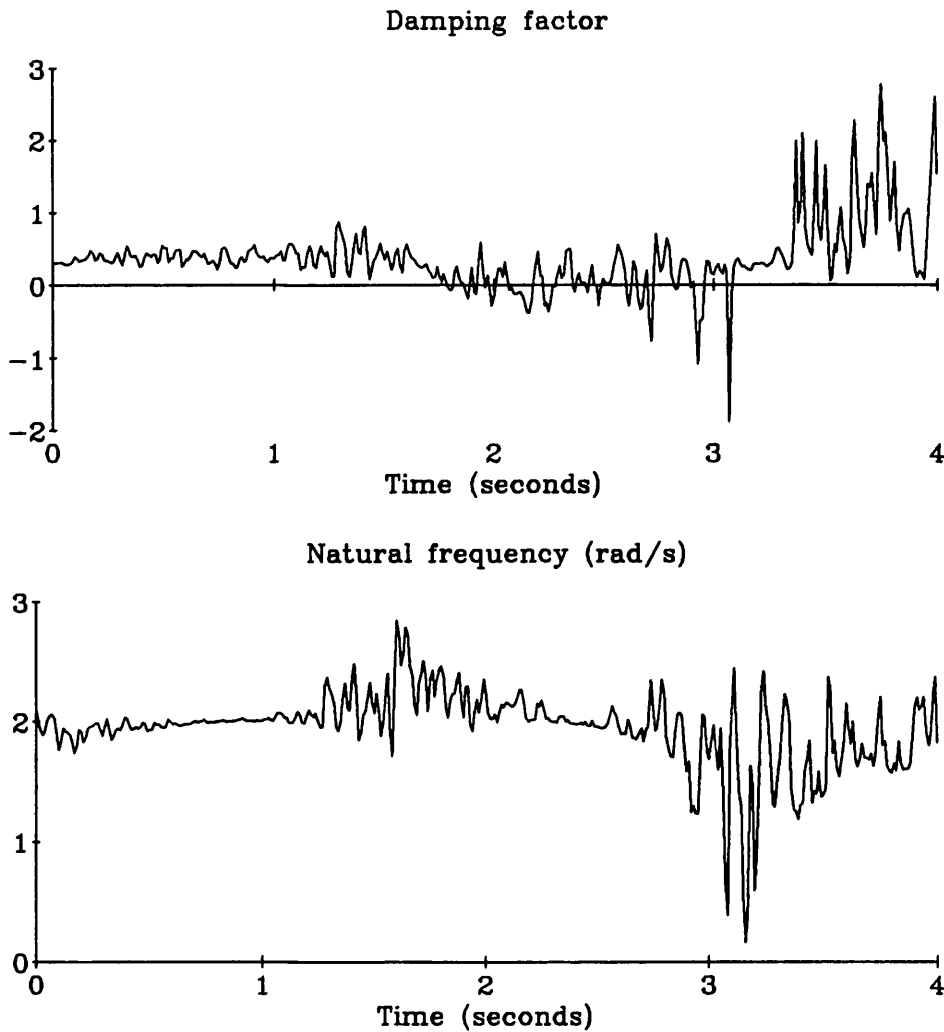


Figure 10 Parameter time histories for second order system with white noise added to the simulated data

Again, the noise free sections correspond to the stationary points where there is only one solution to the cost equation as above.

The model distortion method was also tested on variations of the second order system. In one experiment, a third pole was added to the system generating the simulated data forcing the algorithm to estimate parameter time histories for a second order system

using simulated data generated by a third order system. In the other, a nonlinearity in the form of a velocity squared term was added to the second order system used for generation of the simulated data.

4.4.3 Third order system

In helicopter mathematical model development, unmodelled higher order or nonlinear dynamics are a common source of noise. An example is the high frequency dynamic modes caused by rotor blade flexibility. A model validation method which could identify unmodelled dynamics would be very useful. To test the model distortion technique for this model structure estimation capability, the same second order system was used but with simulated data generated from a system with the same dynamics as the second order system and an extra pole at $s=-0.5$. Its transfer function was

$$\frac{Y(s)}{U(s)} = \frac{\omega^2}{s^2 + 2\xi\omega s + \omega^2} \frac{a}{s+a}. \quad (4.7)$$

The resulting parameter time histories are shown in figure 11.

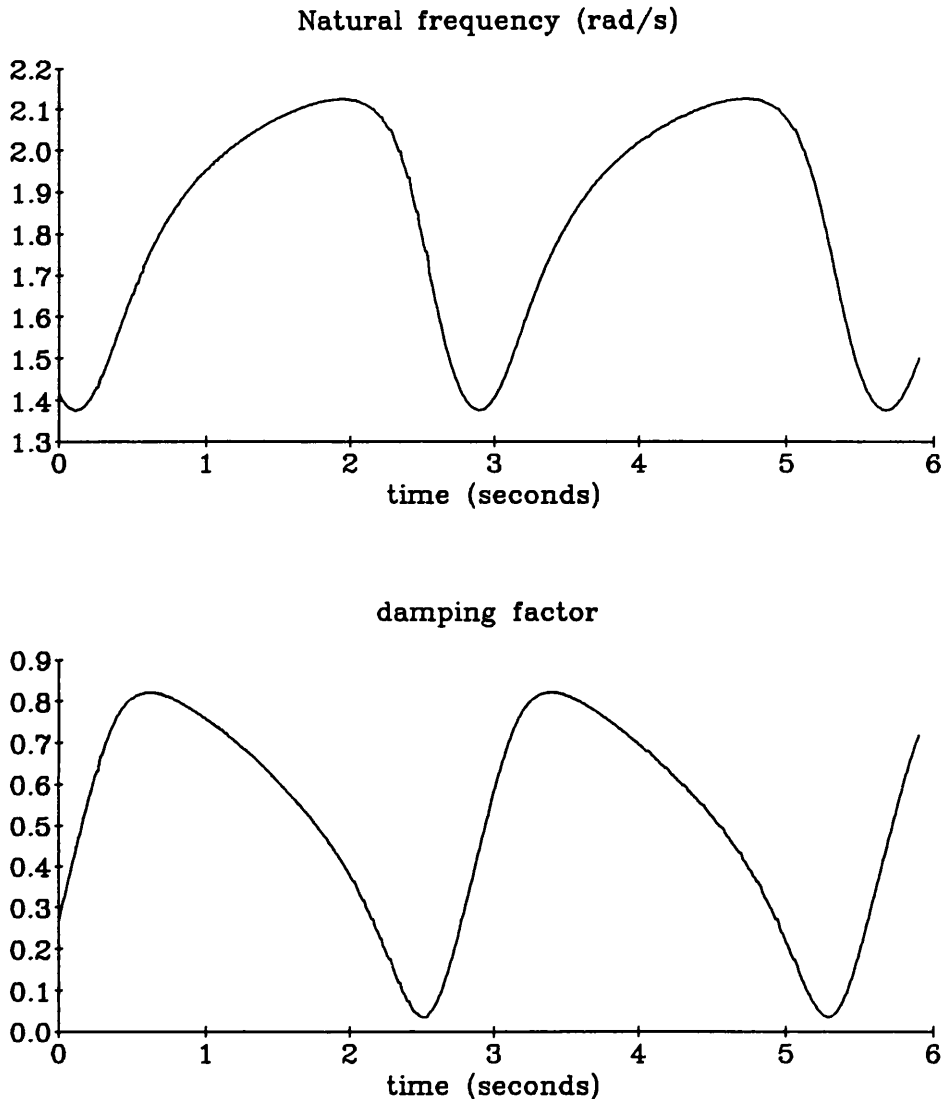


Figure 11 Parameter time histories of second order system using simulated data from a third order system

The parameter time histories were again periodic but with larger amplitude (and variance) than for the second order system. This was because the third order system time response approximated a second order system time response with different values for natural frequency and damping factor. This meant that the a priori parameter estimates were considerably off target. A least squares parameter estimation algorithm was used on the output data from the simulated third order system to estimate natural frequency and damping factor of a second order system. The resulting values of $\omega=1.4$ and $\xi=0.3$ corresponded with the parameter time histories. The larger variance in the parameters suggests less confidence in the validity of the model using the original

validation criteria of the model distortion method. This agrees with the initial second order parameter values being wrong for this third order data.

4.4.4 Nonlinear system

Nonlinearities are a common feature in helicopter models. It was decided therefore to test the model distortion method with a nonlinearity. The nonlinearity consisted of a velocity squared term in one of the system equations (equation 4.7). The parameter time histories are shown in figure 12.

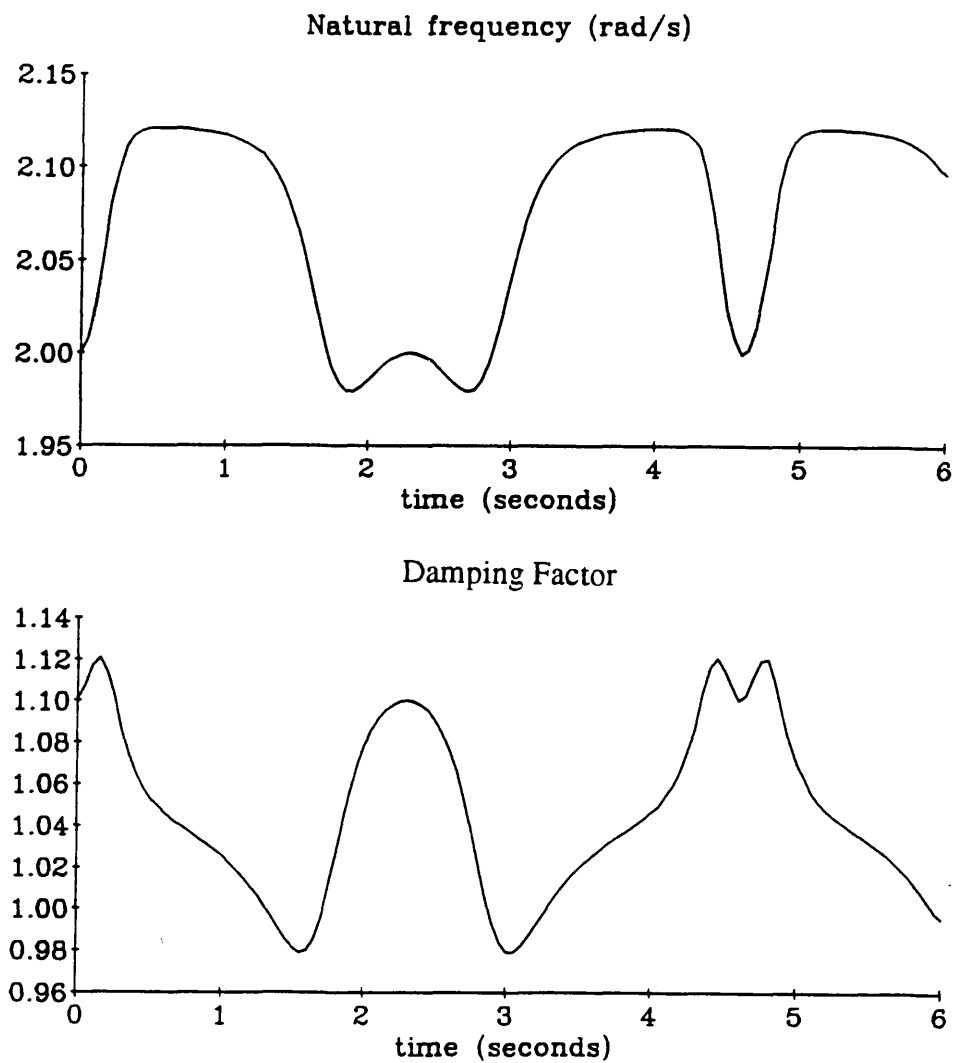


Figure 12 Parameter time history of second order system using simulated data from a nonlinear system

The parameter time histories were again oscillatory - with a large variance and oscillations. The shape of the curve was different from the linear case.

4.4.5 Linear helicopter model

A more realistic test of the algorithm is to use a more complicated model. It was decided that a suitable model would be an eight state linear representation of a helicopter. This provided the option of using genuine flight data from a flight experiment. The model was a six degree of freedom rigid body helicopter model with eight states giving sixty-four possible parameters in the system A matrix. It would not be possible to use all these coefficients as parameters, so some method had to be devised to decide which parameters should be estimated in the optimisation.

It was decided to use sensitivity information to discover which parameters would have the largest effect on the model output if they were changed. The method is detailed in appendix I.

Z_w was selected as the parameter for the Butterfield method with simulated data. Z_{w0} was set to be off by 10%, and it was possible to deduce from the parameter time history (figure 13), that the estimate was low.

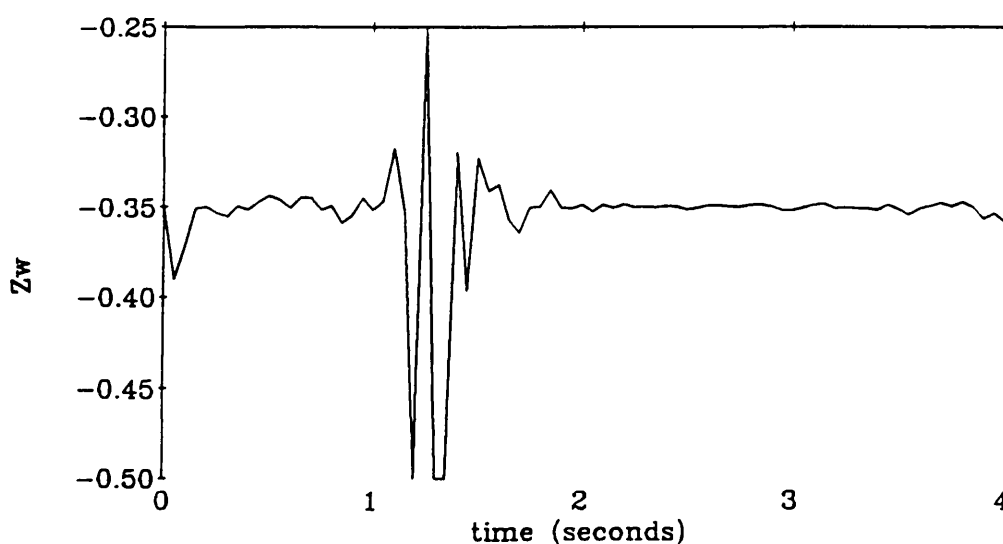


Figure 13 Time response of Z_w

The spike is at the beginning of the transient response and represents the difficulty of matching this part of the model output to the simulated data exactly even with a small a priori parameter estimate. In all the optimisations involving real flight data, the first 5 seconds of data corresponded to trimmed flight. The corresponding part of the time history of the parameters was extremely noisy. For this reason, a time axis of 5.5 to 10 seconds is used in the time response plots.

Real flight data was available for a PUMA in hover as was an eight state linear model. Z_w was again estimated using the model distortion technique. In order to justify using sensitivity coefficients to select the parameters to be optimised, Z_θ was also estimated using the method. Z_θ has a much lower sensitivity coefficient than Z_w , that is, the effect on model output of changing Z_w is much greater than the effect of changing Z_θ . Three optimisations were done, Z_w on its own, Z_θ on its own, and Z_w and Z_θ together.

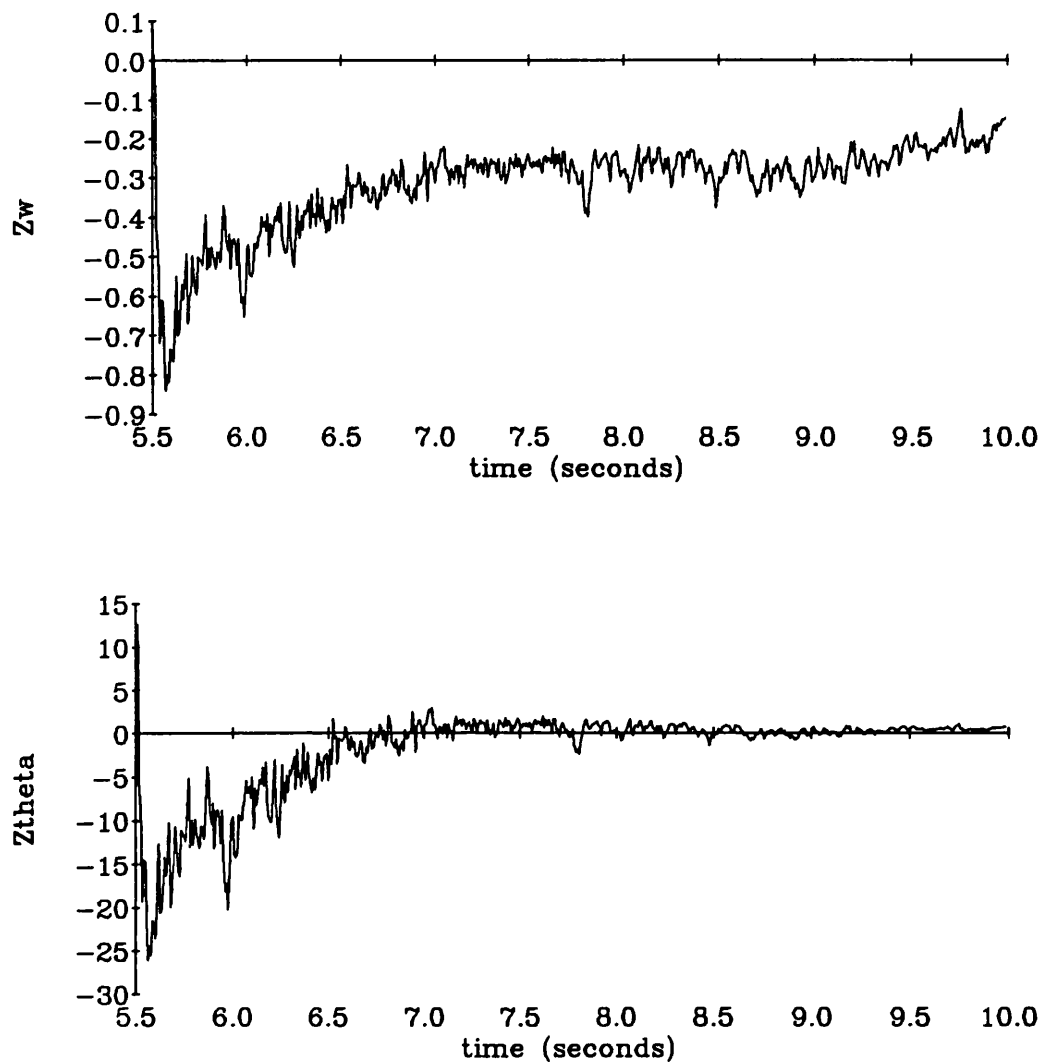


Figure 14a Time response of Z_w and Z_{θ} where optimisation is done separately for each parameter

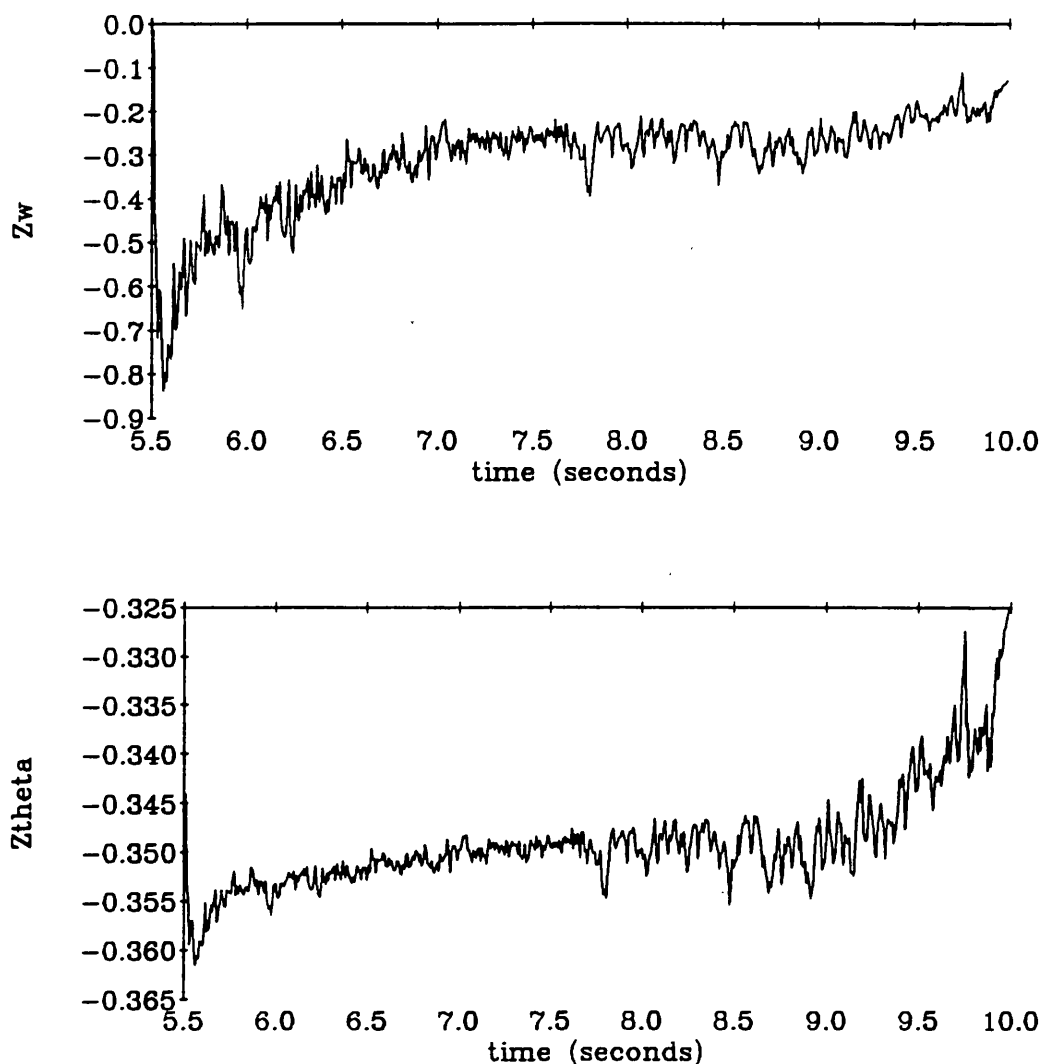


Figure 14b Time response of Z_w and Z_q where optimisation is done for both parameters together

The results (figure 14) show that Z_{θ} varies considerably to give a perfect fit as expected for a parameter with a very low sensitivity coefficient, whereas, the Z_w estimate is reasonably accurate (Z_w has a high sensitivity coefficient). These observations were confirmed by the plots for Z_w and Z_{θ} together which showed Z_{θ} staying at the $Z_{\theta 0}$ values and Z_w adjusting to fit the model to the flight data and having a time history very similar to that generated by estimating Z_w on its own. The start of the transient between $t=5.5$ and $t=6$ seconds gives different parameter values from the rest of the time history. The probable cause of this is that there was a peak in the flight data at the start of following the transient which is not modelled in the linear

six degree of freedom model. At this peak, the parameter time histories also appear noisy (high variance) indicating that the model is not good at this point in the time response (figure 15). This suggests a model structure problem.

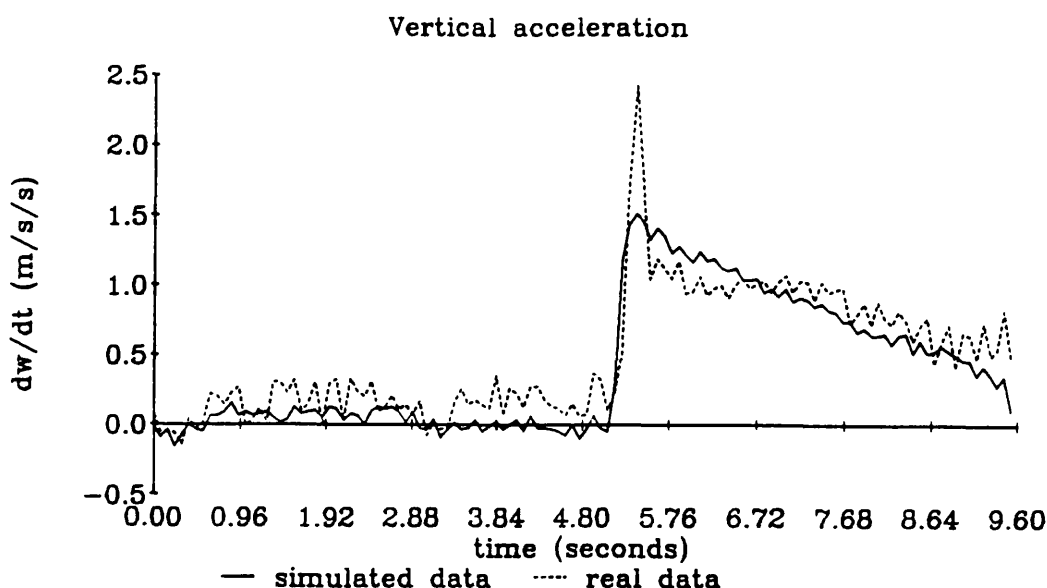


Figure 15 Time response of linear helicopter model compared with flight data

4.5 Discussion

For all the experiments, the optimisations took a long time to run on the computer - typically ten hours of processing time for a simple second order system at 400 points. This represents a significant disadvantage of the model distortion technique compared to other validation methods. The experiments with the simple linear model and simulated data showed that the variance of the resulting parameter time histories could be used as a measure of model validity and although the shape of the time history may be able to give some information regarding parametric values, it is unlikely that model structure information could be determined from the results of this model distortion method. The experiments with the third order system and the nonlinear system produced time histories which showed from the parameter variances that the model was no longer a good representation, but there did not seem to be any correlation between the time history shape and the nature of the modelling inaccuracy.

Where real flight data was used, it was realised that some parts of the model time response would be more accurate than others and it was hoped that this would be reflected in the parameter time histories. This phenomenon was evident to a certain extent with the real data and the linear model where the parameter time history showed a peak corresponding to a discrepancy between the flight data and model output at the start of the transient. It is doubtful however whether the parameter time history revealed any more information than a comparison of model time response with flight data.

Noise presented a major problem especially where the time response of the system is near a steady state or trimmed condition. This is probably because the constraint equation used in the model distortion technique uses the time derivative of the state time history and in a steady state or trim condition, this derivative will effectively be zero. Any additive noise in the system can be effectively thought of as having a constant amplitude. Whereas this noise amplitude may only be 10% of a signal during a transient, approaching the steady state, the noise will dwarf the derivative of the data and the parameter time history will reflect this as can be seen in the steady state in the results from the simple model experiments and the first 5.5 seconds (representing trimmed flight) of results from the experiments with real flight data. Cameron³ concluded that the model distortion method is useful only with noise free data. This agrees with the above findings.

The conclusion must therefore be that this method is unsuitable for helicopter applications because of the noise levels on helicopter flight data and the problems with the technique as the system response approaches steady state or trimmed flight.

References

- 1 Butterfield, M.H., and Thomas, P.J., *'Methods of quantitative validation for dynamic simulation models. Part 1: Theory'*, Transactions of the Institute of Measurement and Control, Vol 8, No. 4, pp182-200, 1986.
- 2 Butterfield, M.H., and Thomas, P.J., *'Methods of quantitative validation for dynamic simulation models. Part 2: Examples'*, Transactions of the Institute of Measurement and Control, Vol 8, No. 4, pp201-219, 1986.

- 3 Cameron, R.G., '*Model validation by the distortion method: linear state space systems*', IEE Proceedings-D, Vol 139, No. 3, pp296-300, 1992.
- 4 Sokolnikoff, I.S., and Redheffer, R.M., '*Mathematics of Physics and modern engineering*', 2nd Ed., McGraw-Hill, 1966.
- 5 *PRO-MATLAB user's guide*, The MathWorks, Inc., MA., pp3-84 - 3-88, June, 1986.

Chapter 5

Development and Validation of a Helicopter Yaw Model

5.1 Introduction

Most helicopters have one main rotor and a tail rotor. The main rotor provides the thrust vector to counteract the aircraft weight and provide a motive force for flight. A result of the engine torque driving the main rotor is a torque reaction in the fuselage which tries to rotate the aircraft in the opposite direction. The most common way to counteract the torque reaction is to use a smaller rotor mounted vertically on the helicopter tail fin producing a sideways force at the end of a long moment arm (the tail boom) to balance the rotor torque. The thrust generated by this tail rotor must be controllable because the main rotor torque varies with flight condition and while manoeuvring. The thrust is controlled by varying the pitch of the tail rotor blades collectively using the pedals in the cockpit. This also gives the pilot yaw control of the aircraft. The tail rotor typically takes 10% of the total engine power.

The fact that in trimmed flight with no yaw rate a sizable amount of tail rotor thrust is still required to balance the main rotor torque means that aircraft yaw control to left and right is asymmetrical. Turning in one direction requires tail rotor thrust to be increased towards its maximum where blade stall may start to cause problems. Turning in the opposite direction means reducing tail rotor thrust eventually to the extent that the rotor enters the vortex ring condition. This is where the velocity of the air flowing through the rotor is the same as the free stream velocity of the air flowing around the rotor causing vortices to be generated and shed alternately from each side of the rotor dramatically reducing the thrust produced by the rotor. These turning rate limits are also affected by the free stream velocity of the air around the rotor i.e. the velocity of the aircraft.

The tail rotor blades are usually hinged in the flap degree of freedom. This hinge is often engineered so that the blade pitch is coupled to the blade flap. Blade pitch coupling reduces the steady state and transient flapping by introducing a cyclic variation in blade pitch. The coupling is usually negative but it has been found by Gaffey¹ that positive coupling is also effective.

The tail rotor affects the aircraft dynamic response in degrees of freedom other than yaw. The aircraft yaw mode is coupled to roll through the Dutch roll mode. This is an oscillatory mode with a period of 1-2 seconds which occurs in all aircraft. The tail rotor produces its own torque reaction in the fuselage in aircraft pitch mode and the high rate of rotation of the tail rotor means that the effect of gyroscopic precession on the aircraft body states cannot be ignored.

Modelling or prediction of the behaviour of the tail rotor is difficult. In its 'trim' position, the tail rotor is balancing the main rotor torque which varies with flight condition. The aerodynamic environment at the tail rotor is very difficult to predict because of its proximity to the tail fin and the effects of main rotor downwash including main rotor blade tip vortices. This makes accurate prediction of the force generated by the tail rotor very difficult.

One result of this is that many aspects of tail rotor design have evolved as much through experimentation and trial and error as through performance predictions using mathematical models. Design features, such as which side of the fin to put the rotor and its direction of rotation, have been determined by experience and experimentation². The most efficient configuration is now accepted to be a tractor arrangement where the tail rotor wake strikes the fin and rotation such that the blade nearest the main rotor is going up.

5.2 Flight data acquisition system on research PUMA

Instrumentation on the PUMA helicopter at the Defence Research Agency at Bedford is capable of recording all aircraft rates and attitudes, control positions and blade motions^{3,4}.

The data acquisition system involves a variety of sensor types. Blade flap and lag angle are measured by potentiometers on the respective hinges. Body accelerations and angular rates of rotation are measured by linear accelerometers and rate gyros situated in a sensor package at the centre of gravity of the helicopter. Rotor speed is measured using an optical sensor and a disk connected to the rotor shaft with a white line painted on it. One effect of this sensor is that there is only one record of rotor speed for each revolution of the disk giving a discrete signal with a low sampling frequency (one per revolution or about 4.4 Hz). The effect of this low sampling rate is investigated in chapter 6. There are also pressure sensors and strain gauges on an

instrumented main rotor blade providing information which allows calculation of aerodynamic loads and blade deformation. A nose boom on the helicopter allows sensors for airspeed, angle of attack and sideslip to be located out of the downwash from the main rotor.

The analogue signals from the various sensors on the main rotor go to an amplifier mounted on top of the rotor hub. The amplification gives an improved signal to noise ratio. The data then passes through slip-rings into the helicopter. The signals are then multiplexed to one channel, passed through an analogue to digital converter and stored on magnetic tape as a digital signal.

A Central Processing Unit controls the multiplexer. There is a maximum data rate of 128 K samples per second for all the data. The multiplexer controller can be programmed to allocate sample rates to different variables thereby giving a higher priority and sampling rate to certain measurements. For example, blade flap and lag angles are sampled at 1024 Hz, rotor speed is sampled at 64 Hz.

The data are stored digitally on tape and also transmitted in real-time to a ground station for fatigue damage accounting and observation of maximum load limits. The data can then be taken from the tape and put into a different format for analysis on a computer.

5.3 Experiments and flight data

In May 1988, a set of flight experiments were flown in Italy in exceptionally good still air conditions with the RAE PUMA helicopter. The experiments included hover turns out of ground effect with step inputs to the pedals to port and starboard of magnitude 4° , 8° and 13° . The experiments are referred to as events 3,4,5,7,8 and 10. During the experiments, the pilot worked to keep the aircraft level and in the hover. It was proposed that the flight data from this experiment, which included aircraft yaw rate, could be used to validate and possibly improve the tail rotor representation in HELISTAB.

Blade collective pitch data was not available due to a failure of the slip ring assembly on the helicopter. It was necessary therefore to derive the tail rotor collective pitch from the tail rotor jack data. The relevant calibration record was found and the gradient information calculated using a least squares fit to a straight line (figure 16).

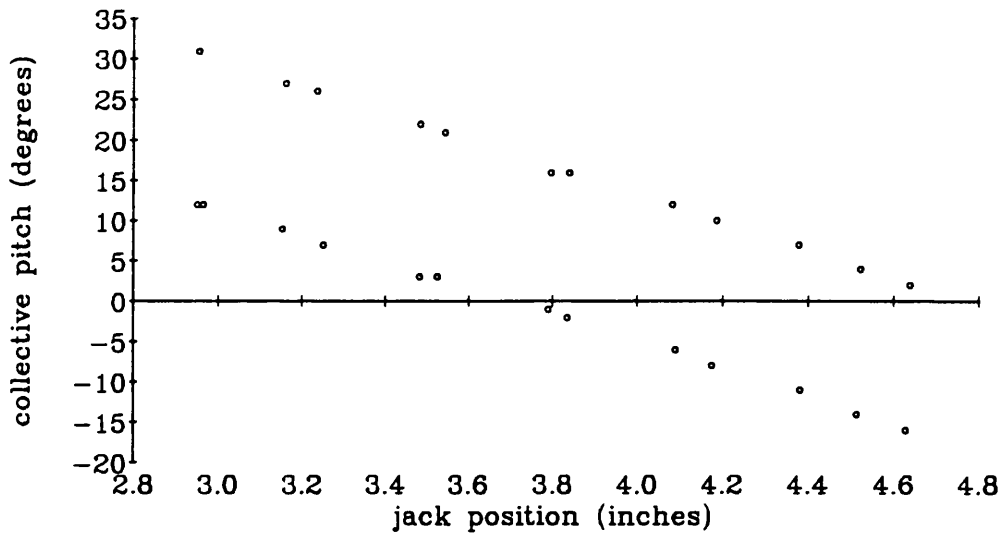


Figure 16 Jack Calibration showing ground based measurements of tail rotor pitch actuation mechanism. Testing in both directions shows backlash.

$$\theta_{0t} = -0.30 \times \text{Jack} + \text{offset} \quad (\theta_{0t} \text{ is in radians}) \quad (5.1)$$

This method could not however, be used to find the offset. The offset was calculated by looking at the trim at the start of each of the flight records and getting the trim position for the tail rotor jack. The trim collective pitch was derived using the trim equations of the helicopter and the offset was calculated. This also ensured that the model was trimmed correctly. This is important because any errors would be integrated up in the simulation.

5.4 Development and Validation of Yaw model

HELISTAB⁵ is a real-time nonlinear helicopter model developed at the Defence Research Agency at Bedford. From initial validation studies using analogue matching⁶, the aircraft yaw response to pedal inputs had been identified previously as a weak point of the model. The model of the tail rotor in HELISTAB uses instantaneous inflow optimised iteratively with no pitch flap coupling.

It was realised that tail rotor dynamics were a crucial part of any yaw model. Available flight data included step responses to pedal inputs with the pilot maintaining a hover (hover turns). This meant that it would be possible to model the system as a one degree of freedom yaw model developing and validating the tail rotor dynamics

without the additional complication of other dynamic modes from the rest of the aircraft.

Various model validation/development techniques were applied to this problem with the aim of evaluating these techniques as well as improving the helicopter yaw model.

The constraint of real time simulation means a simple low order model of the yaw dynamics and particularly the tail rotor is necessary. Several hypotheses were examined for model improvement within this constraint.

- a) Possibility that the tail rotor enters the vortex ring condition.
- b) Incorporating pitch-flap coupling in the model.
- c) The effect of wind blowing the main rotor downwash over the tail rotor.
- d) Aerodynamic drag on the side of the helicopter tail boom.
- e) The effect of the main rotor downwash flowing over the tail rotor in hover.

Validation techniques used were analogue matching of time responses, parameter estimation using a linear search and analogue matching of time responses for different parameter values.

5.4.1 Helicopter yaw dynamics - theory

The following equations are taken from Padfield⁵ and describe the basis for a simple one degree of freedom mathematical representation of yaw dynamics on the aerospace PUMA helicopter. The model is a nonlinear representation with instantaneous inflow calculated using an iterative loop with a Newton Raphson minimisation.

Figure 17 gives a diagrammatic representation of the helicopter.

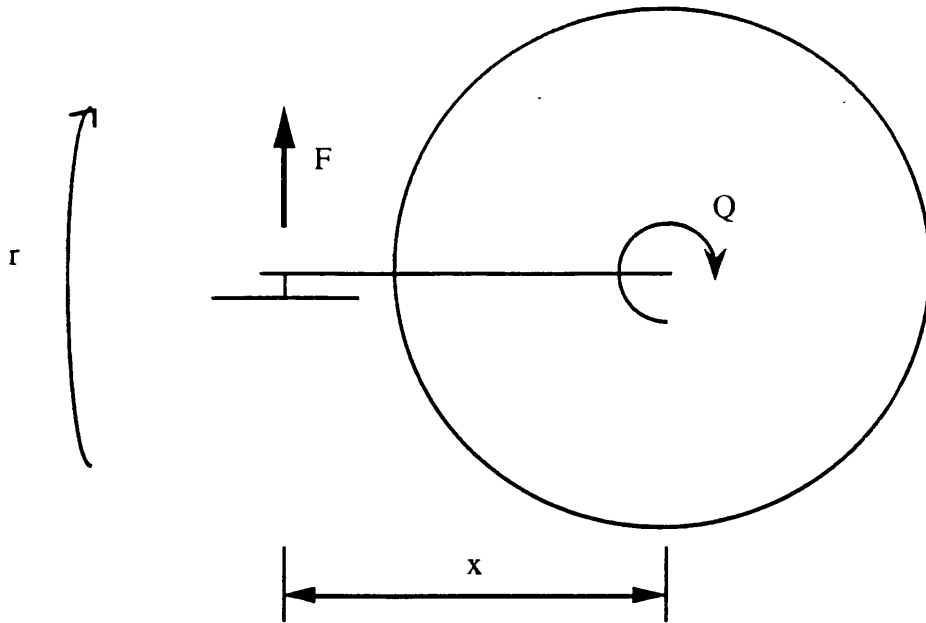


Figure 17 Schematic diagram of helicopter

The yaw equation for a one degree of freedom yaw model is

$$\dot{r} = \frac{xF - Q}{I_{zz}} \quad (5.2),$$

where,

Q = main rotor torque,

x = distance form tail rotor to main rotor,

F = thrust force produced by tail rotor,

r = aircraft yaw rate,

I_{zz} = moment of inertia about vertical z axis.

The tail rotor thrust is calculated thus,

$$F = \rho (\Omega R_T)^2 \sigma a_{0T} (\pi R_T^2) \frac{C_{TT}}{a_{0T} \sigma} F_{blk} \quad (5.3),$$

where,

ρ = air density,

Ω = tail rotor rotation speed,

R_T = tail rotor radius,

σ = tail rotor solidity,

a_{0T} = lift curve slope of tail rotor,

C_{TT} = tail rotor thrust coefficient,

F_{blk} = Fin blockage factor.

The fin blockage factor is an empirical parameter which represents the effect on the thrust of the tail fin.

$$F_{blk} = 1 - \frac{3}{4} \frac{S_{FN}}{\pi R_T^2}$$

where,

S_{FN} = area of tail fin.

The thrust coefficient and the tail rotor inflow are calculated together in an iterative loop.

$$C_{TT} = \frac{a_{0T} \sigma}{2} \left(\frac{\theta_{0T}}{3} \left(1 + \frac{3}{2} \mu_T^2 \right) + \frac{(-\mu_z - \lambda_{0T})}{2} \right) \quad (5.4),$$

and

$$\lambda_{0T} = \frac{C_{TT}}{2 \sqrt{\mu_T^2 + (-\mu_z - \lambda)^2}} \quad (5.5),$$

where,

θ_{0T} = tail rotor collective pitch,

μ_t = normalised airspeed tangential to tail rotor disk,

μ_z = normalised airspeed perpendicular to tail rotor disk,

λ_{0T} = inflow velocity at tail rotor.

The tail rotor collective pitch, θ_{0T} includes pitch flap coupling.

$$\theta_{0T} = \frac{\theta_{00T} + \delta_3 \left(\frac{n_\beta}{\lambda_\beta^2} \right) \frac{4}{3} (-\mu_T - \lambda)}{1 - \delta_3 \left(\frac{n_\beta}{\lambda_\beta^2} \right) (1 + \mu_T^2)} \quad (5.6)$$

where,

θ_{00T} = blade collective pitch at blade root,

δ_3 = pitch flap coupling coefficient,

n_β = blade inertia number,

λ_β = rotor blade flap frequency ratio

The yaw model was to be simulated using TSIM⁷, a FORTRAN based simulation program running on a VAX. A fourth order Runge-Kutta integration algorithm was used for the integration. Initially the model was configured to represent coning and pitch flap coupling (δ_3 was found to equal -45°) with the calibration equation (equation 5.1) controlling the input. The model was trimmed at each experiment by adjusting the initial condition of the tail rotor blade collective pitch as described above.

5.4.2 Pitch-flap coupling

Even although pitch flap coupling is a physical phenomenon, it is important to validate the decision to incorporate pitch flap coupling into the model. Equation 5.4 implies that the addition of pitch flap coupling will make the effective tail rotor collective pitch a dynamic variable with a first order lag rather than a constant as is the case with no pitch flap coupling. This extra pole will effectively slow down the response of the model to an input. It can be seen from figure 17 that in all cases, the model response is faster than the experimental data suggesting that incorporating pitch flap coupling would improve the model. To verify this, event 3 was simulated with

and without pitch flap coupling. From the results (figure 18) can be seen that the fit is improved where pitch flap coupling is incorporated.

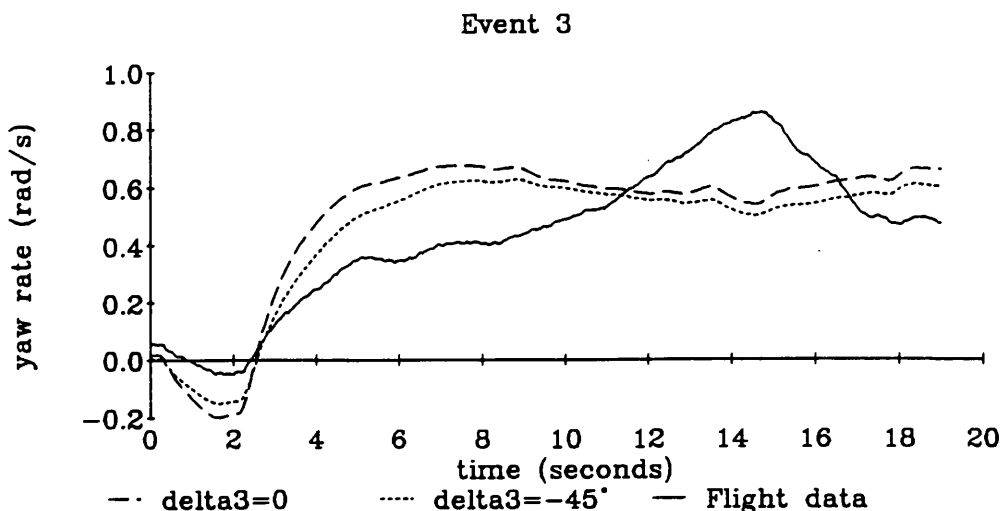


Figure 18 Yaw model time response to step input on tail rotor collective pitch in hover with and without pitch flap coupling

5.4.3 Investigation of the effect of drag on the tail boom

The downwash from the main rotor is blown down over the tail boom. When the helicopter is turning , this generates an angle of attack large enough for the creation of aerodynamic lift on the tail boom in a direction so as to affect the yaw moment of the helicopter. There will also be aerodynamic drag on the side of the tail boom as the helicopter rotates in the yaw axis. A diagram of the tailboom showing the lift effect is given in figure 19.

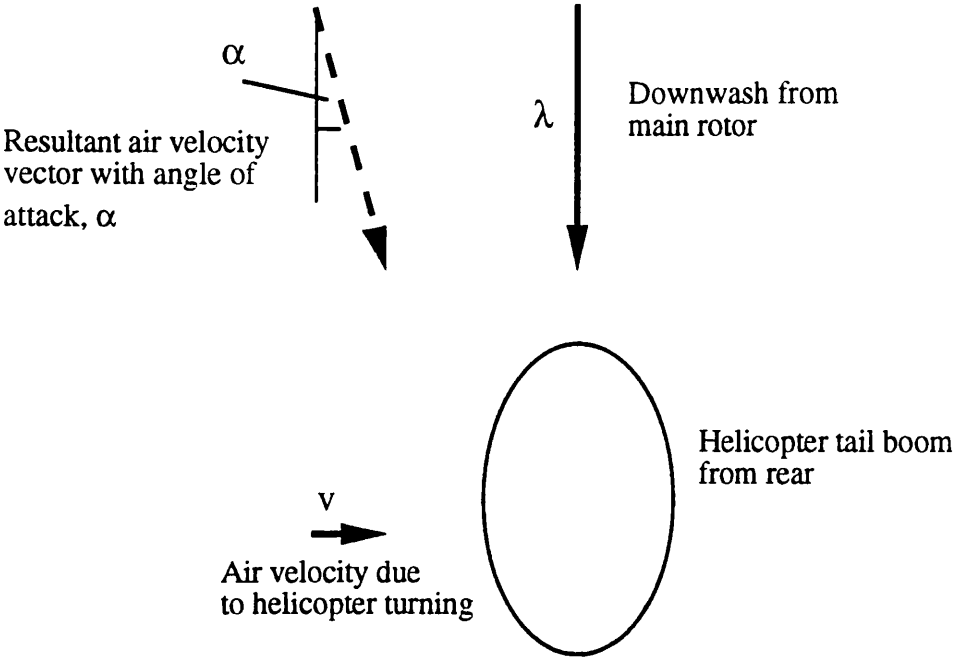
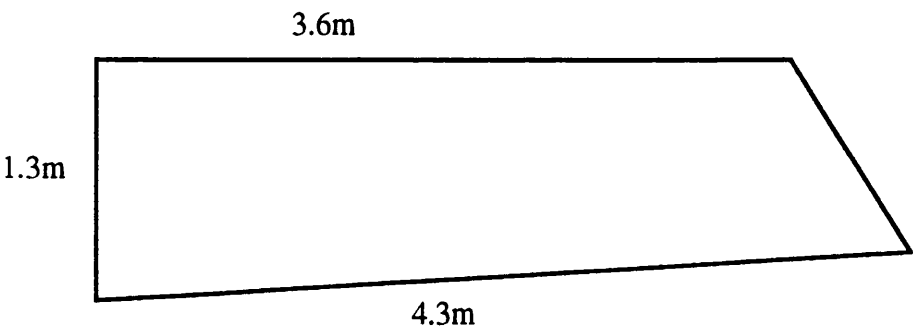


Figure 19 Airflow over helicopter tailboom

This effect is of course dependent on the shape of the tail boom. On the Aerospatiale PUMA, the dimensions of the tail boom are as in figure 20.

Side view



Plan view

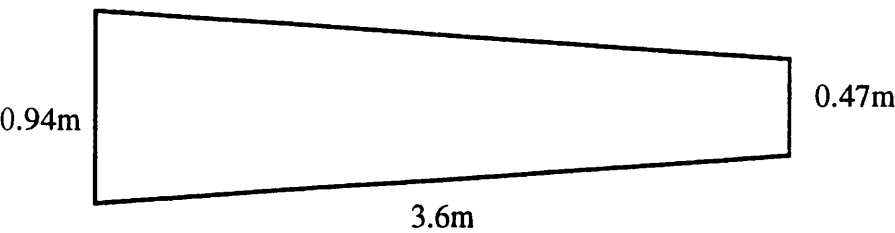


Figure 20 Aerospatiale PUMA tail boom(from Jane's all the World's aircraft⁸)

As can be seen from figure 20, the area of the tail boom looking side on is approximately 4m². The thickness of the tail boom as seen from above is about 60% - too large for any appreciable aerodynamic lift effect. This leaves aerodynamic drag on the side of the tail boom as a possible influence on yaw dynamics. The effect of this drag was calculated roughly for a worst case.

$$D = \frac{1}{2} \rho V^2 C_D s$$

where,

D = aerodynamic drag force,

ρ = air density (1.22),

V = air velocity,

C_D = drag coefficient,

s = area of surface.

Given a turn rate of 0.5 radians per second (from figure 21) at a distance of approximately 6m from the aircraft hub, the air velocity is 3 metres per second. Assuming a worst case, the drag coefficient is 1. The area, from figure 20, is 4m². Therefore $D=21$ N giving a turning moment of 126 Nm at 6m from the hub. This is negligible compared to the 40 000N main rotor torque.

It can therefore be concluded that flow of air around the tailboom has no effect on the aircraft yaw mode.

5.4.4 Investigation of vortex ring and wind effects

The time response of the model with pitch flap coupling incorporated is shown in figure 21 together with the flight data for a set of hover turns to port and starboard.

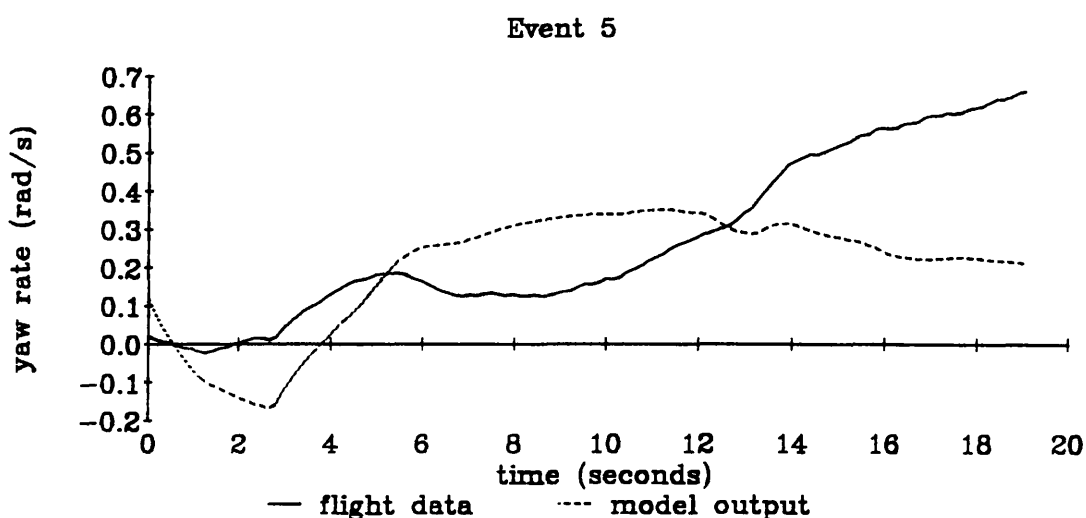
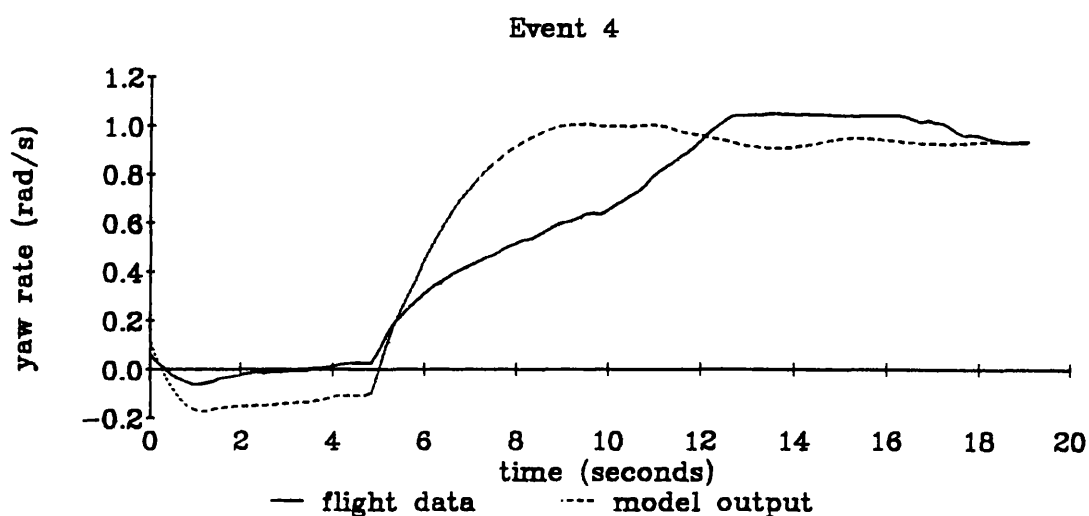
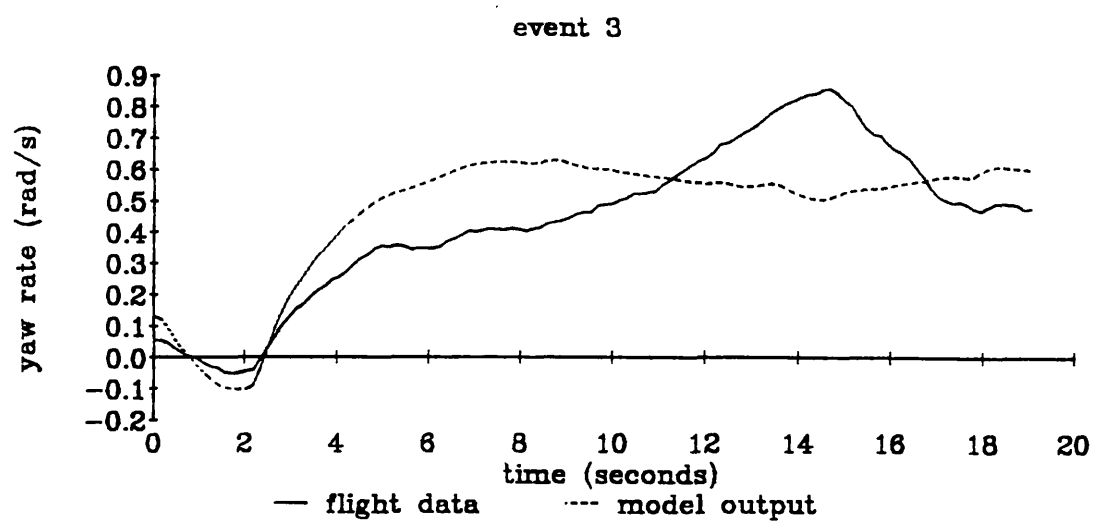


Figure 21a Time response of yaw model with pitch flap coupling to step input in tail rotor collective pitch in hover

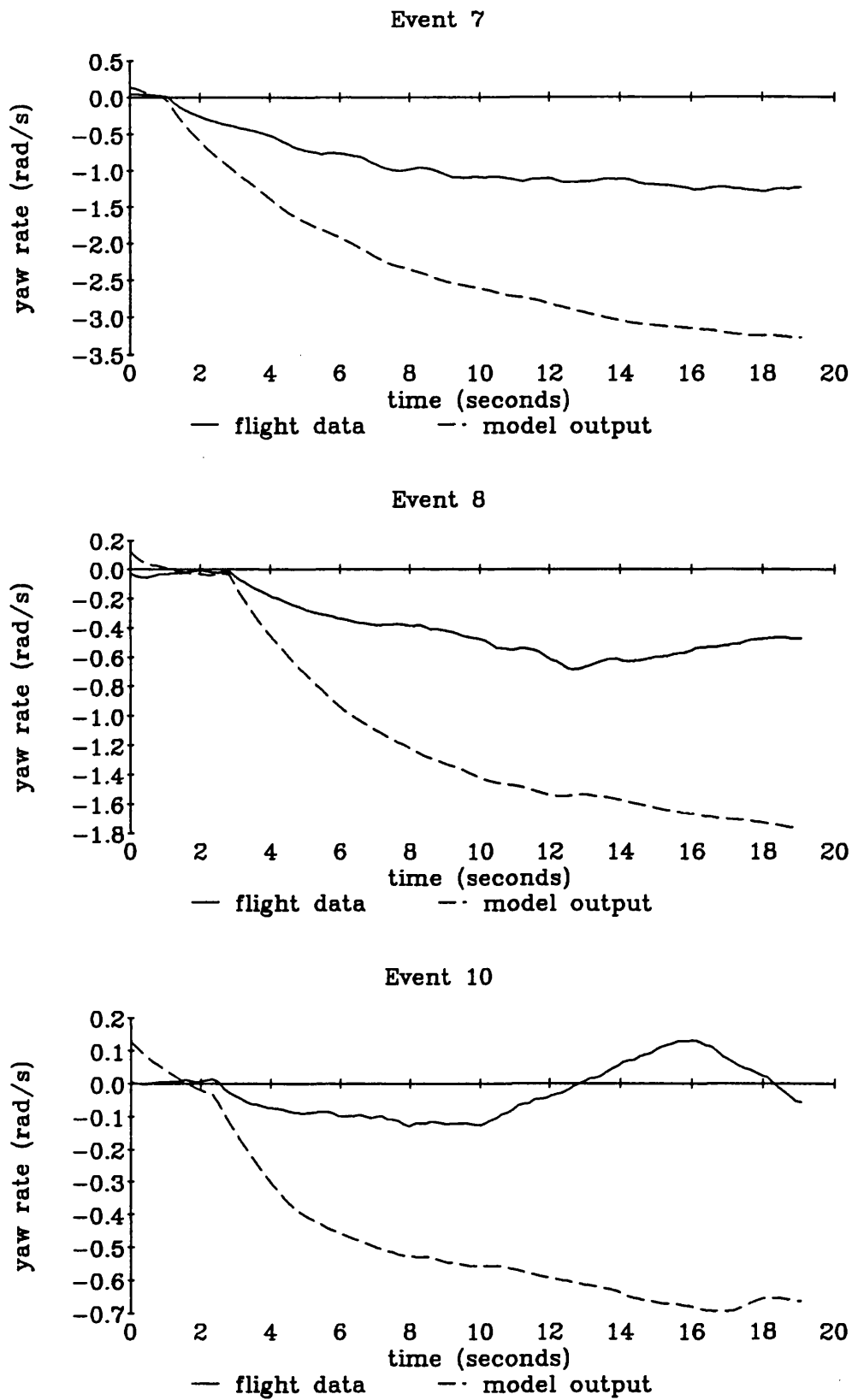


Figure 21b Time response of yaw model with pitch flap coupling to step input in tail rotor collective pitch in hover

It can be seen that the model is relatively good for turns to starboard (positive yaw rate) but to port, (negative yaw rate) gives very poor predictions. This asymmetry hinted at the possibility that the tail rotor was entering the vortex ring condition in turns to port. This was ruled out because the simulation revealed tail rotor inflow at the trim condition to be of the order of 16 metres per second. This corresponds to an aircraft yaw rate of 1.8 radians per second. The vortex ring condition arises when the inflow velocity is close to the velocity of the air around the tail rotor⁹. It can be seen from the flight data in figure 21 that the maximum aircraft yaw rate for these experiments is nowhere near this value.

It was noticed that the flight data exhibited a periodic motion which correlated with the heading angle or yaw attitude. One possibility is a geographical feature on the horizon which would cause the pilot to alter the pitch of the aircraft periodically while trying to maintain hover. The records of pitch attitude from the flight were retrieved and this possibility was ruled out. Another possibility was a low velocity wind. The conditions on the day of the experiment were classified as still air. This means that the wind speed never exceeded 2 knots (1 m/s) during the experiment, but even a small wind speed would have a large effect on aircraft yaw rate if it caused the main rotor downwash to be blown over the tail rotor increasing the tail rotor inflow and therefore increasing the tail rotor thrust.

It was realised that the effect of wind blowing the main rotor downwash over the tail rotor would depend on how much of the tail rotor disk was in the downwash. The downwash was modelled as an eccentric cylinder coming down from the main rotor disk as in figure 22. The fraction of the disk in that cylinder was calculated as follows.

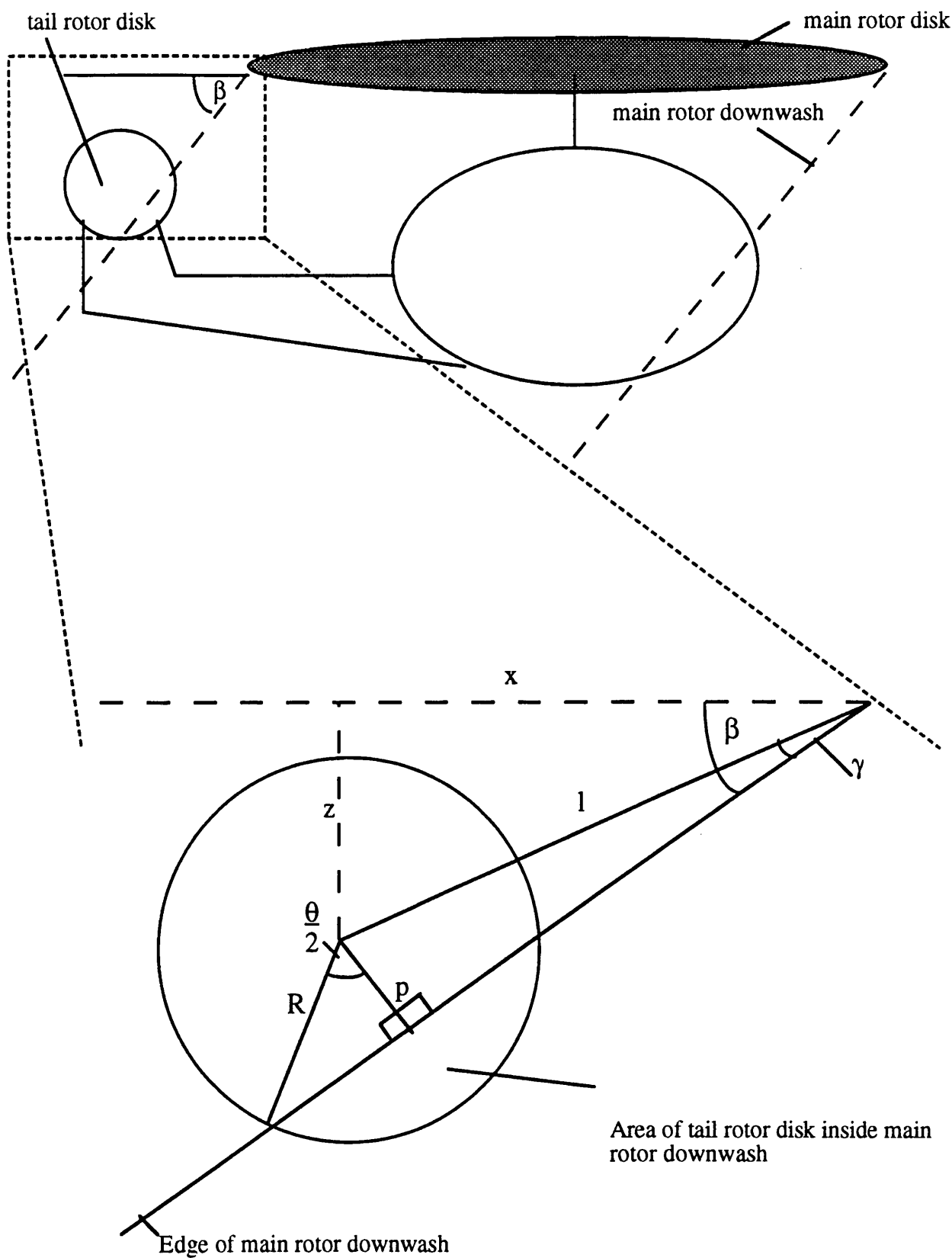


Figure 22 Diagram showing tail rotor disk in main rotor downwash

To calculate the angle of the main rotor downwash, let \mathbf{p} be the vector representing the edge of the downwash.

$$\mathbf{p} = (w, -\lambda, 0),$$

where λ is the downwash velocity and w is the velocity of the wind blowing the aircraft. The wind is coming from the direction indicated by the angle ϕ where $\phi=0$ represents head on. After rotating \mathbf{p} around the vertical axis by ϕ ,

$$\mathbf{p}' = (w \cos \phi, -\lambda, 0).$$

This means that the angle of the downwash is

$$\tan \beta = \frac{-\lambda}{w \cos (\phi - \phi_0)} \quad (5.7)$$

where ϕ_0 is the direction the wind is coming from.

Given,

$$\gamma = \tan^{-1} \beta - \tan^{-1} \left(\frac{z}{x} \right),$$

$$l = \sqrt{x^2 + z^2},$$

$$p = l \sin \gamma,$$

$$\theta = 2 \cos^{-1} \left(\frac{p}{R} \right),$$

the area of the disk that is inside the main rotor downwash is,

$$A = \begin{array}{ll} 0 & p > R \\ \frac{\theta}{2} R^2 - \frac{1}{2} p R \sin \left(\frac{\theta}{2} \right) & R > p > -R \\ \pi R^2 & p < -R \end{array} \quad (5.8).$$

This area was multiplied by an arbitrary constant, an overall scale factor, to give a value for tangential air velocity at the tail rotor.

The shape of the effect of aircraft heading angle on tangential inflow is plotted in figure 23 which represents tangential inflow for event 3, where the helicopter is in a hover turn, for arbitrary parameter values.

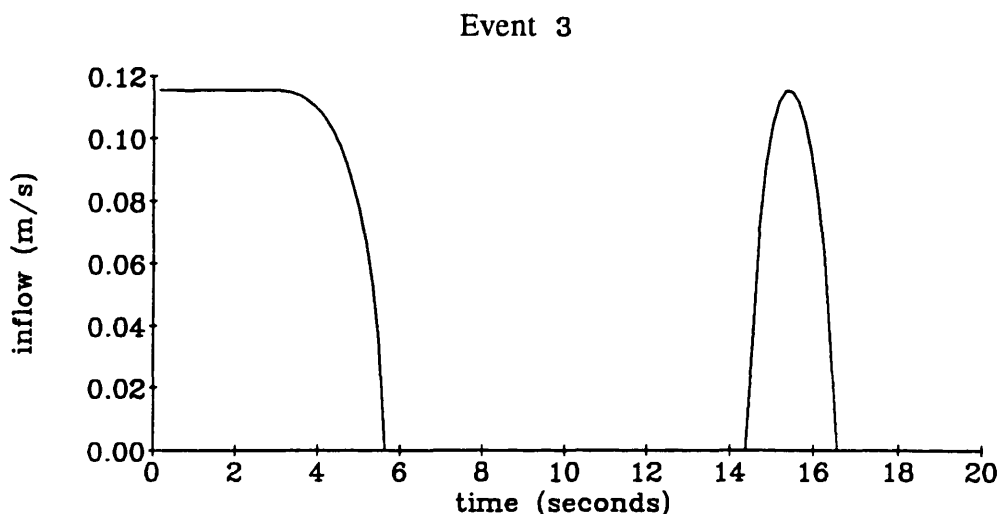


Figure 23 Time response of expected tangential inflow arising from wind blowing main rotor downwash over tail rotor for tail rotor collective step input

It was realised that the main rotor downwash would not form an exact cylinder and the other effects such as tail fin interference and main rotor blade tip vortices would have an important effect, but the model was considered to be a sufficiently accurate representation for an initial investigation into this problem.

The wind model was included in the tail rotor model and had three parameters - the wind direction, the wind speed and an overall scale factor for the thrust coefficient. Because of the iterative loop, it was not possible to derive analytic sensitivity coefficients. Using finite differences also caused problems because of the discontinuities. This made parameter estimation difficult.

In order to investigate the effects of each of the three parameters, a linear search was done where each parameter was varied in turn and the sum of the square of the error between model output and flight data (for yaw rate) over a time history of 19 seconds was plotted. This was done for events 3 and 7, one for port and one for starboard (figure 24).

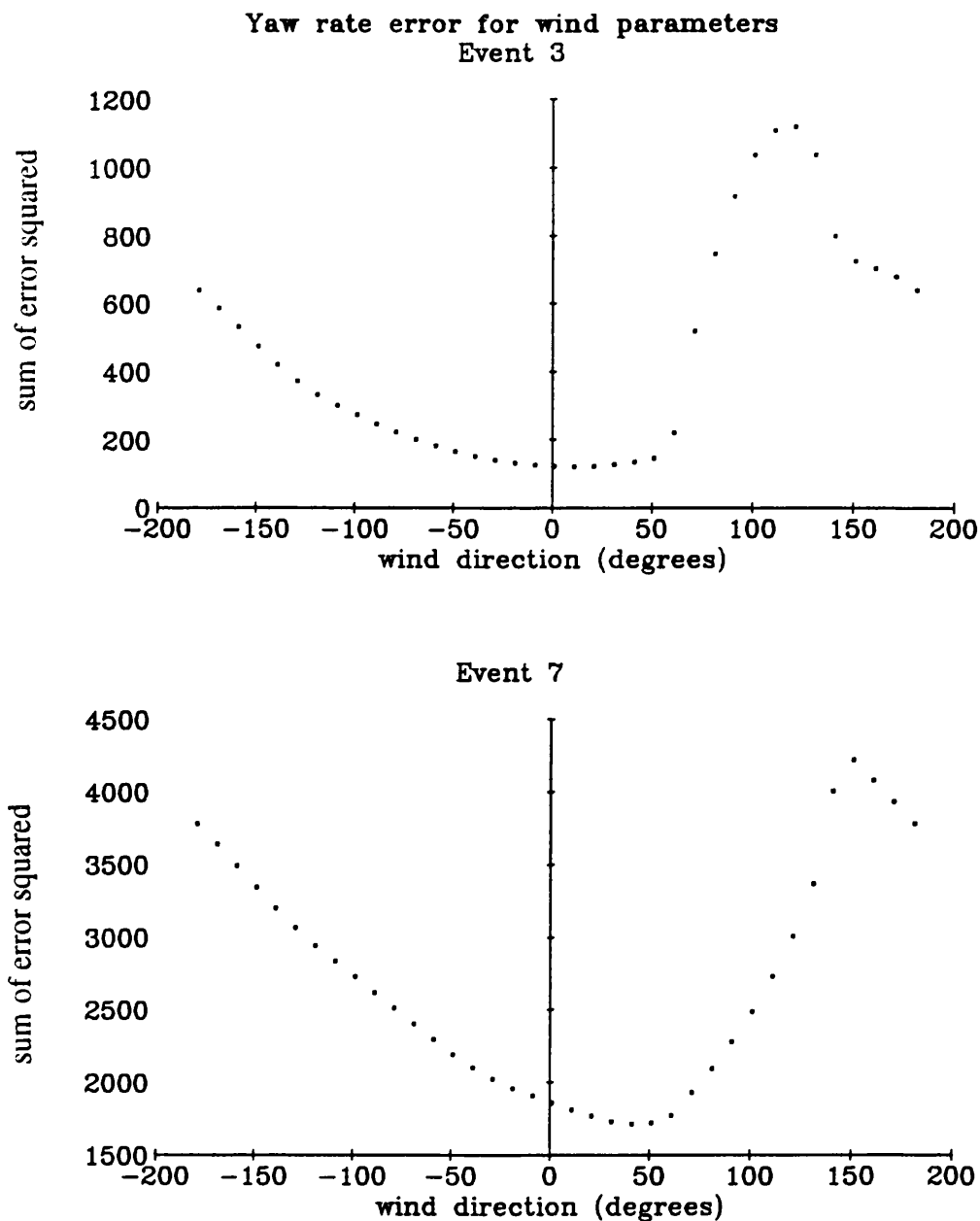


Figure 24 Variation of sum of squared error between flight data and model output for yaw model where downwash is blown over the tail rotor while varying wind direction

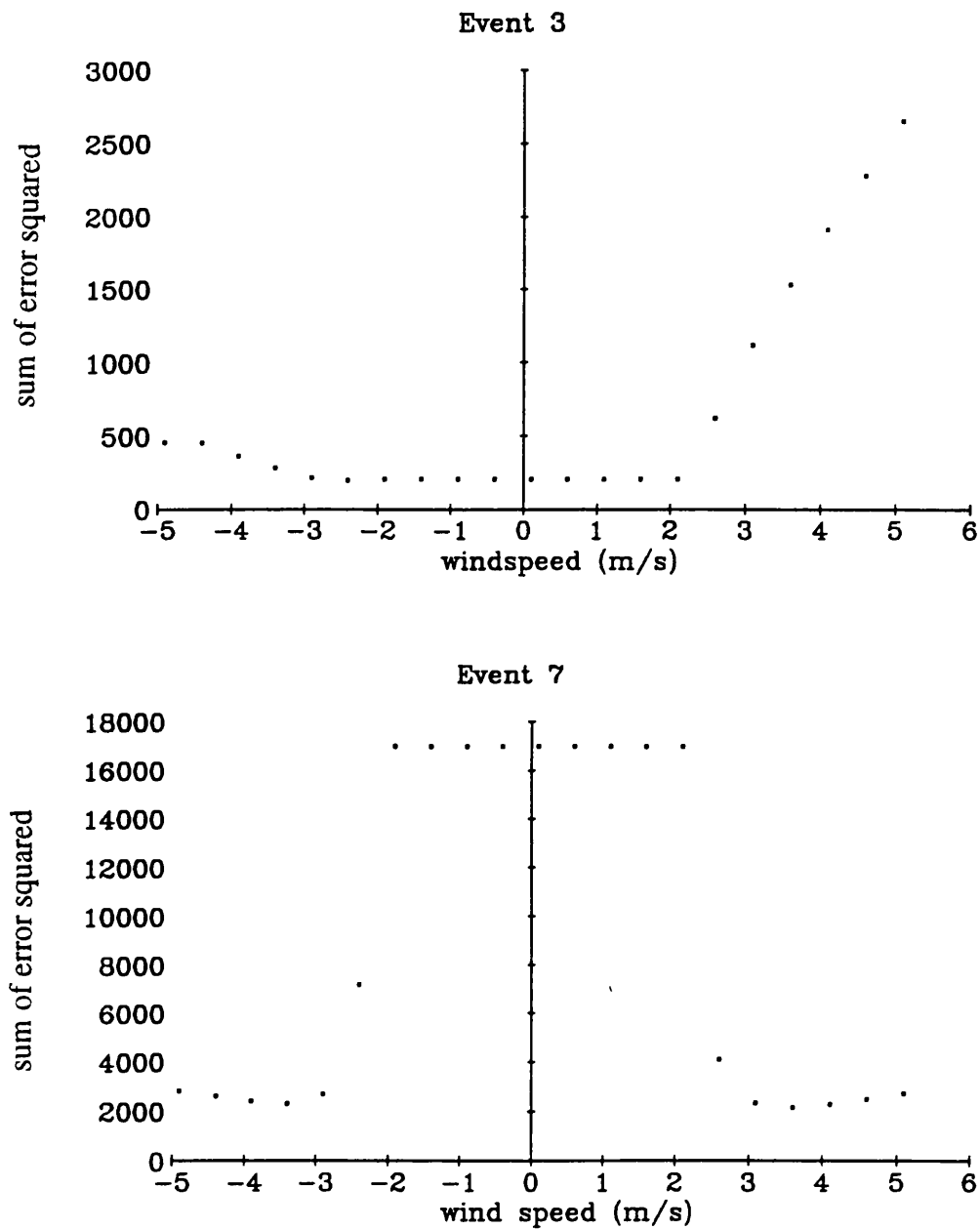


Figure 24 (cont.) Variation of sum of squared error between flight data and model output for yaw model where downwash is blown over the tail rotor while varying wind speed

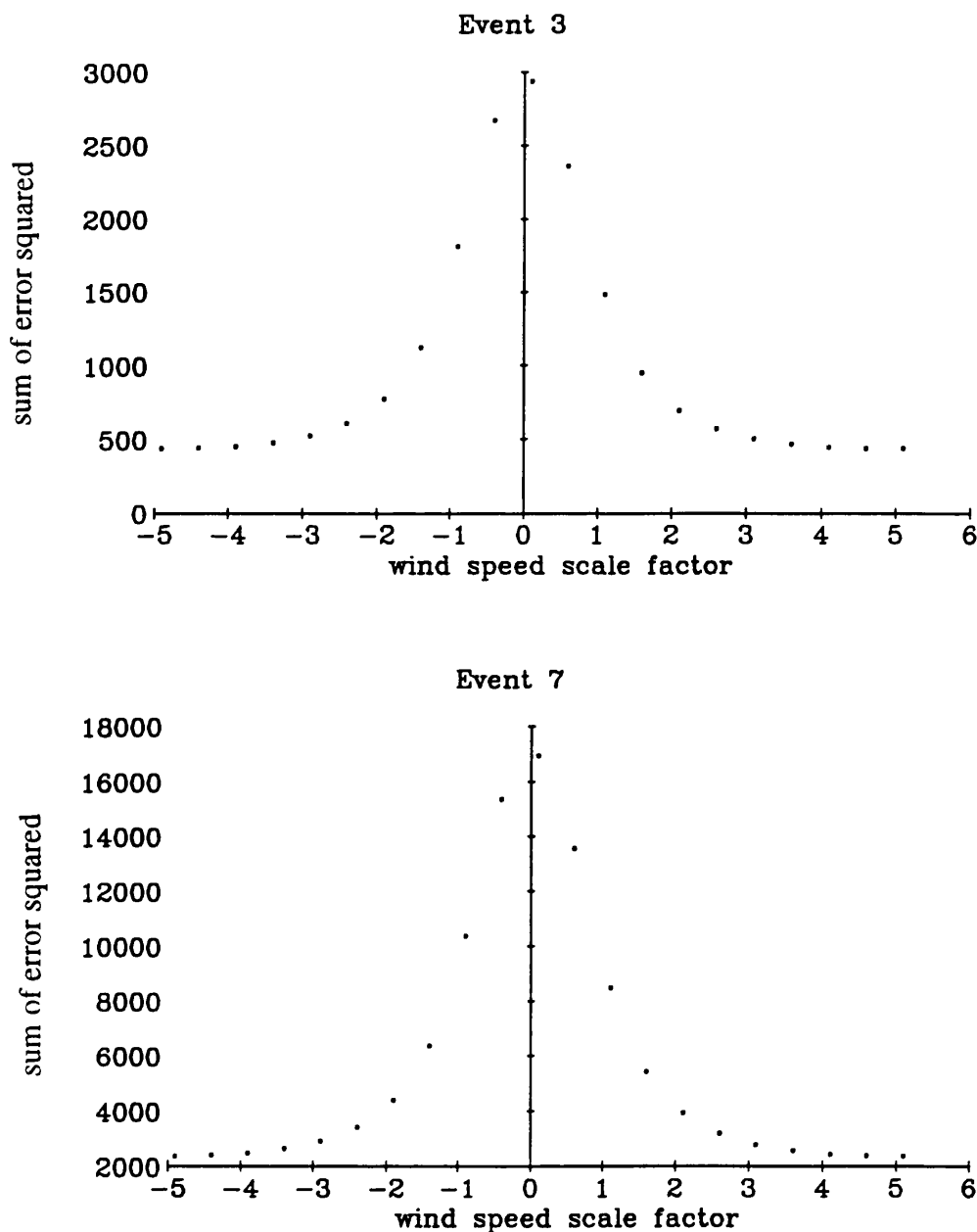


Figure 24 (cont.) Variation of sum of squared error between flight data and model output for yaw model where downwash is blown over the tail rotor while varying effect of additional inflow on tailrotor

The flat part of the graph of wind speed represents the region of operation where the tail rotor is completely outside the downwash of the main rotor. For event 3, this region coincides with the minimum of the cost function implying that adding wind

makes the model worse rather than better. Conversely, the cost function plots for event 7 suggest that the adding the wind to the model improves the accuracy of the model. The tail rotor was simulated with the wind parameters selected as the values giving the smallest value for the cost functions for event 7. The time response (figure 25) shows that although the average magnitude increases to match the flight data, and the error between flight data and model output is reduced compared to the original (figure 21), the wind introduces a periodicity which is not present in the flight data and not realistic regarding the high rates of angular acceleration. It was concluded that although adding the wind decreased the cost function, it did not improve the accuracy of the model. The wind gave the yaw of response an oscillatory nature which reduced the overall magnitude of the model output and the squares of the difference between model output and flight data but it is obviously not a good fit to flight data. This investigation highlighted the importance of visually examining the model output and the experimental data and not relying on a cost function as an indication of model validity.

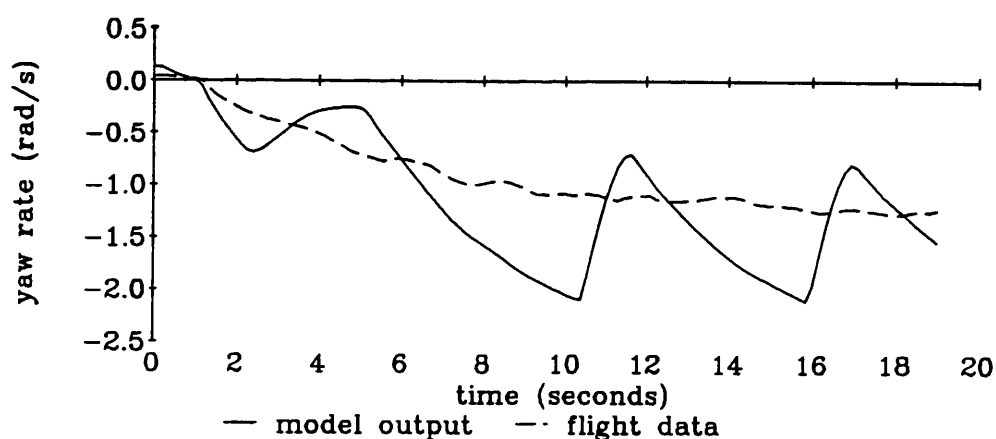


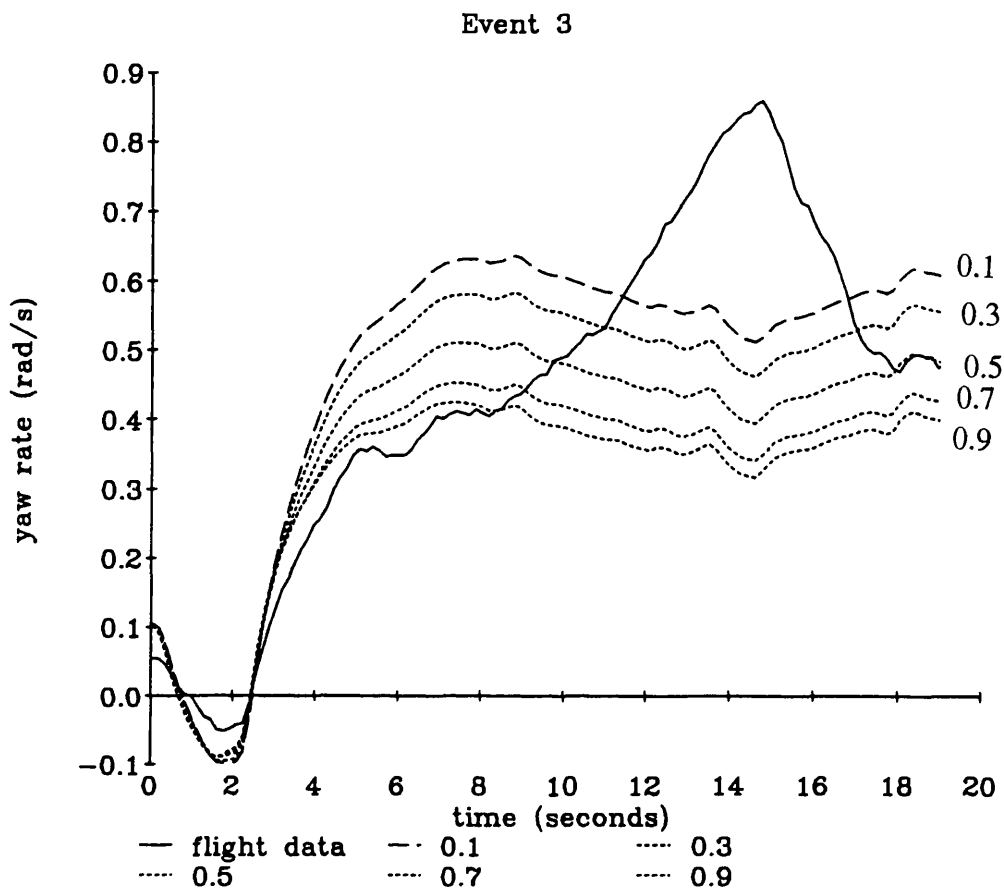
Figure 25 Time response of yaw model with wind blowing main rotor downwash over the tail rotor with optimised parameters

5.4.5 Main rotor downwash

The largest discrepancy between model output and flight data appeared to be the difference in fit for turns to port and starboard. Rotor downwash blowing over the tail rotor would increase the tangential inflow and could as a result increase the thrust generated by the tail rotor quite considerably. Because of the asymmetrical nature of

helicopter yaw dynamics where the tail rotor is counter balancing the main rotor torque, any increase in the thrust produced by the rotor is going to have more of an effect in turns to starboard than to port. The model at this stage did not include any representation of main rotor downwash. The model was altered to include a representation of the main rotor downwash. The additional tangential inflow velocity was defined as a scale factor, INFCON, multiplied by the main rotor inflow. The scale factor was defined as a parameter but parameter estimation was not possible because the model had to be retrimmed with each new value of INFCON. A linear search was carried out on each of the events with INFCON being varied from 0.1 to 0.9. The sum of the error squared was calculated at each time point and the time responses were recorded for turns to port and starboard. They can be seen in figure 26.

Time history for variation of INFCON



Cost function for variation of INFCON

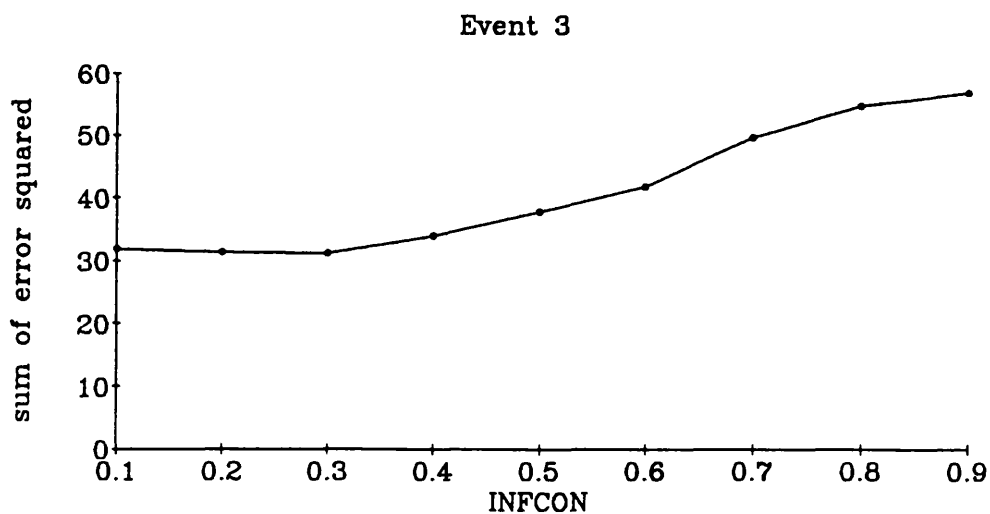
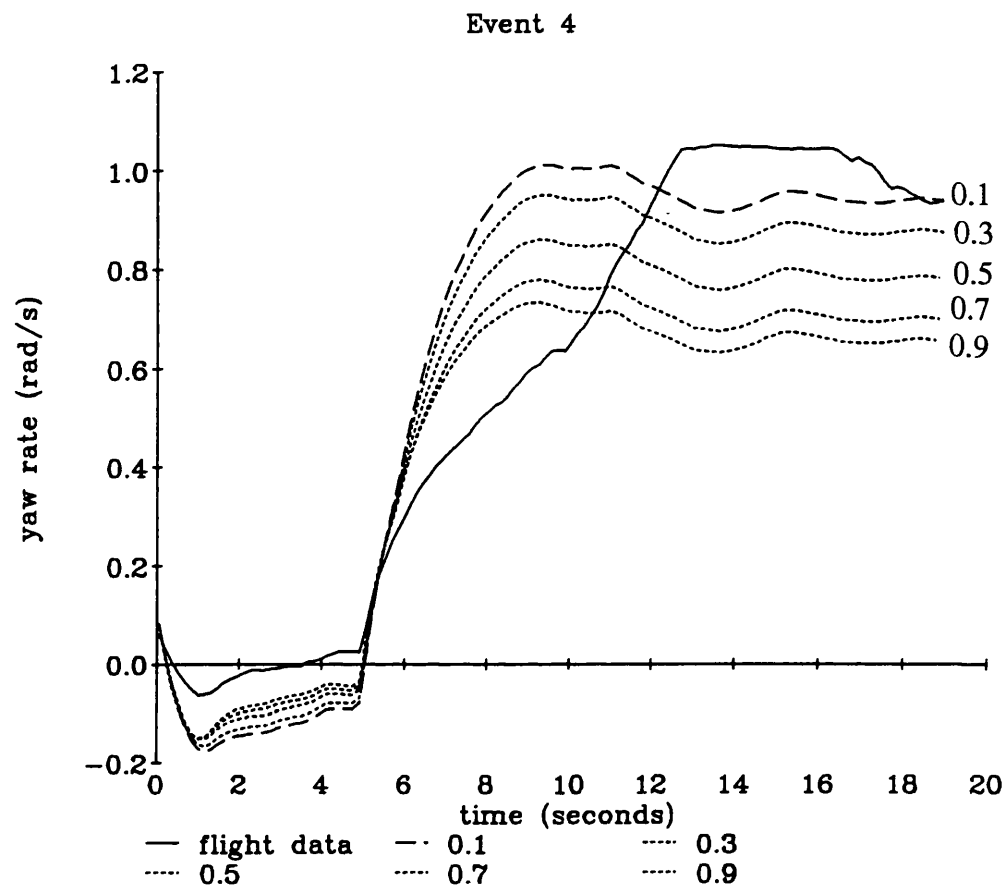


Figure 26 Variation of parameter INFCON which represents the fraction of main rotor downwash contributing to tangential inflow at the tail rotor

Time history for variation of INFCON



Cost function for variation of INFCON

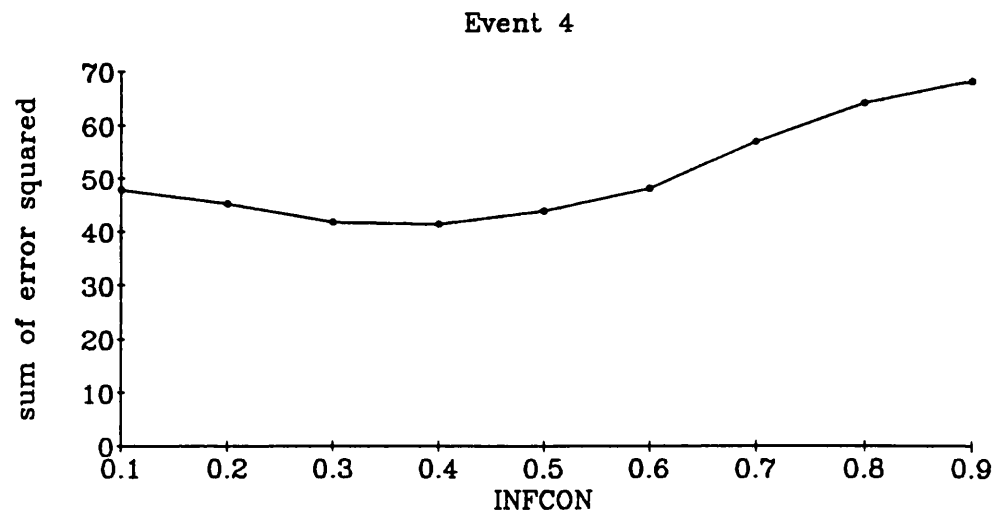
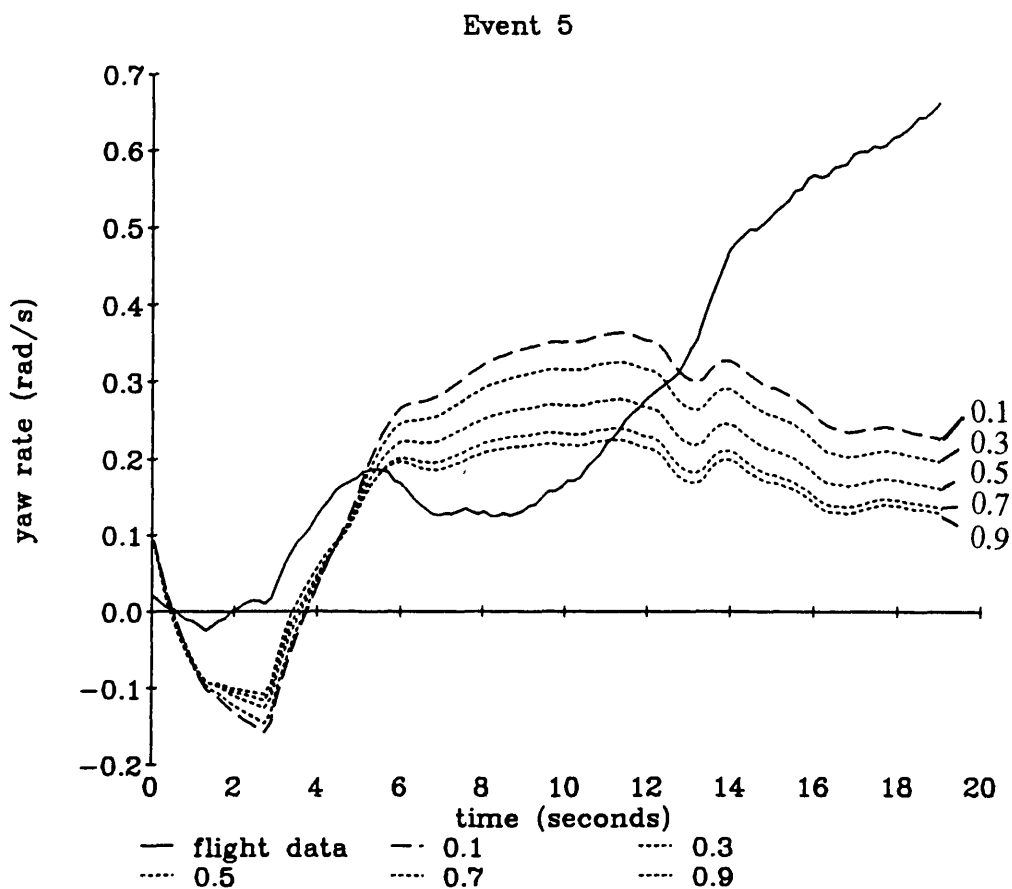


Figure 26 (cont.) Variation of parameter INFCON which represents the fraction of main rotor downwash contributing to tangential inflow at the tail rotor

Time history for variation of INFCON



Cost function for variation of INFCON

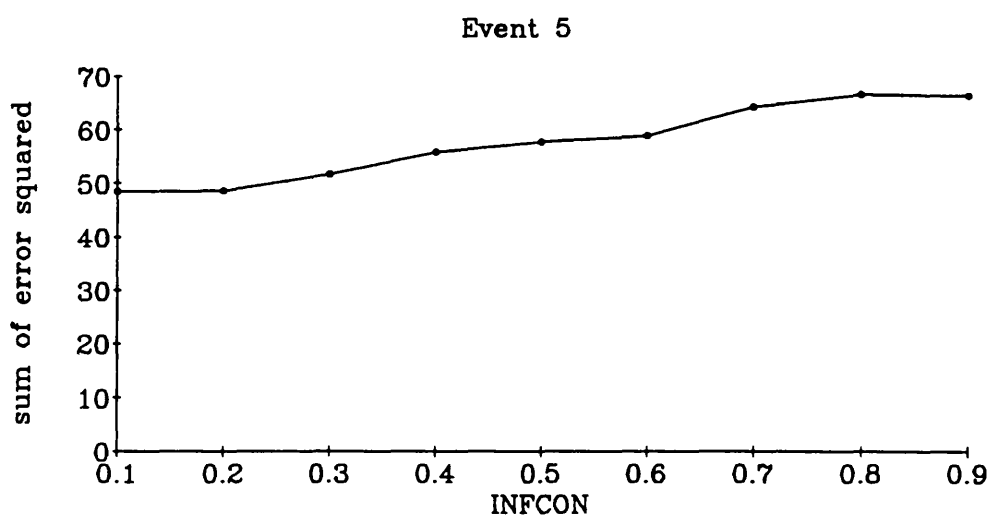


Figure 26 (cont.) Variation of parameter INFCON which represents the fraction of main rotor downwash contributing to tangential inflow at the tail rotor

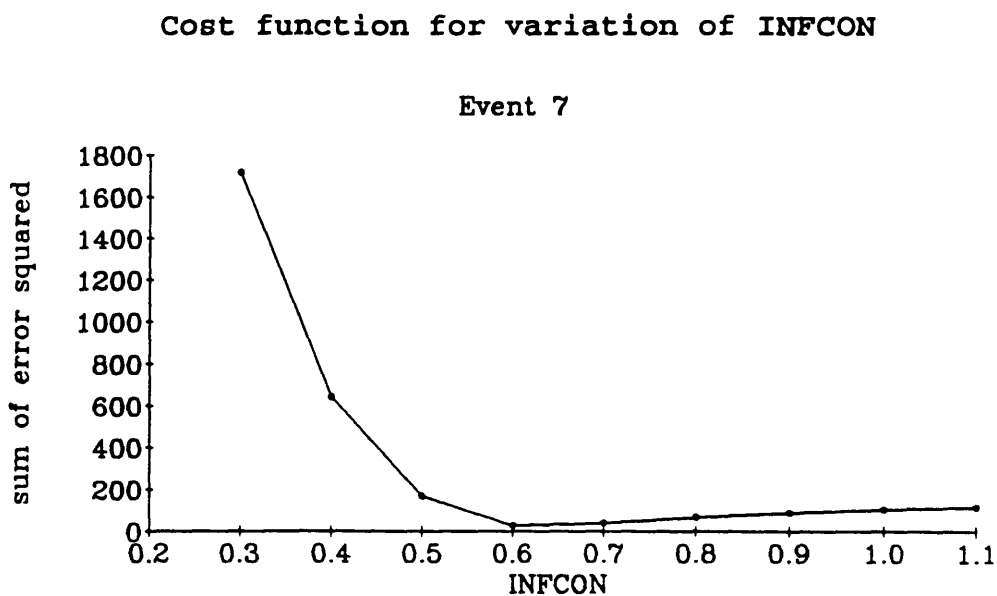
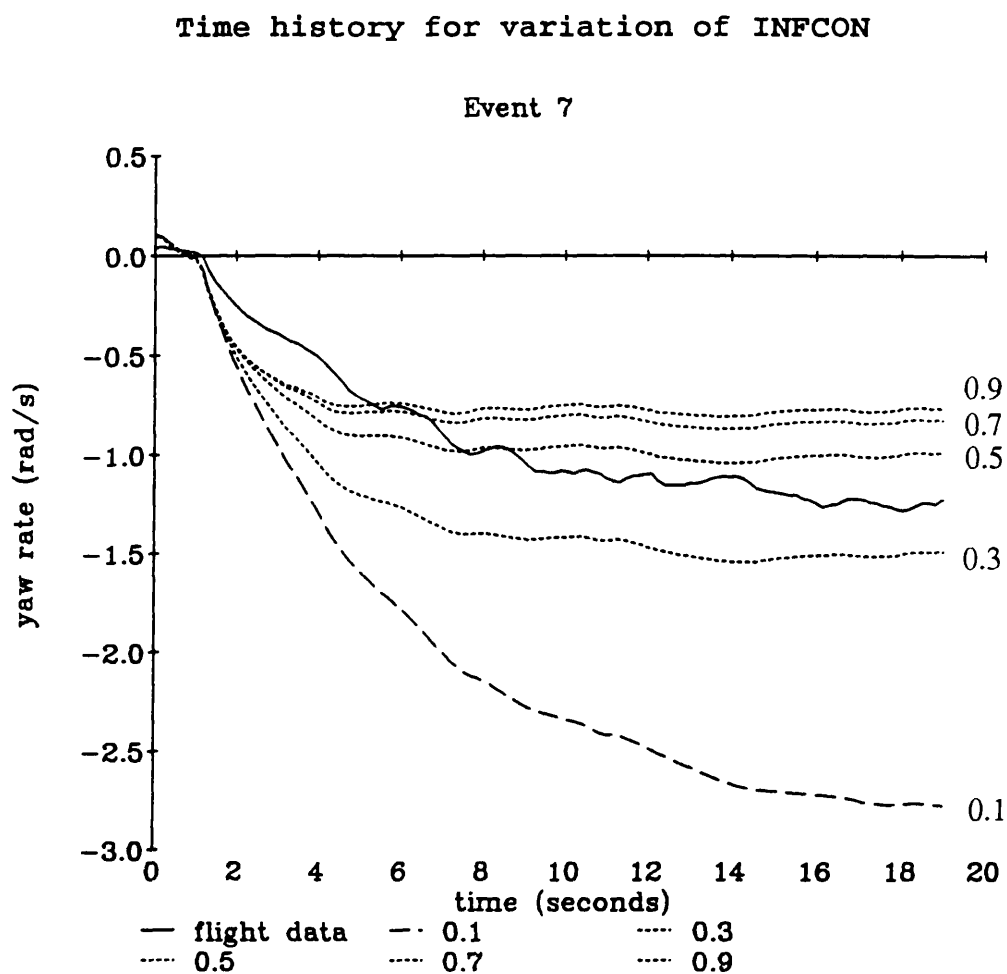
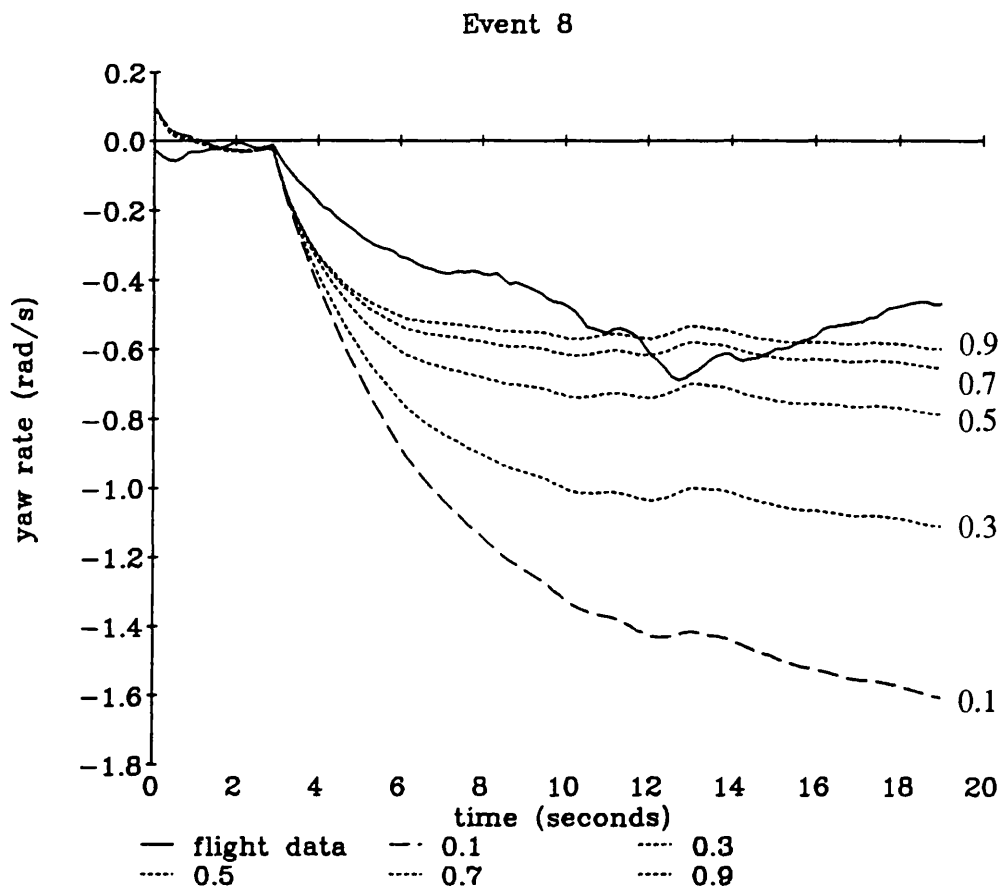


Figure 26 (cont.) Variation of parameter INFCON which represents the fraction of main rotor downwash contributing to tangential inflow at the tail rotor

Time history for variation of INFCON



Cost function for variation of INFCON

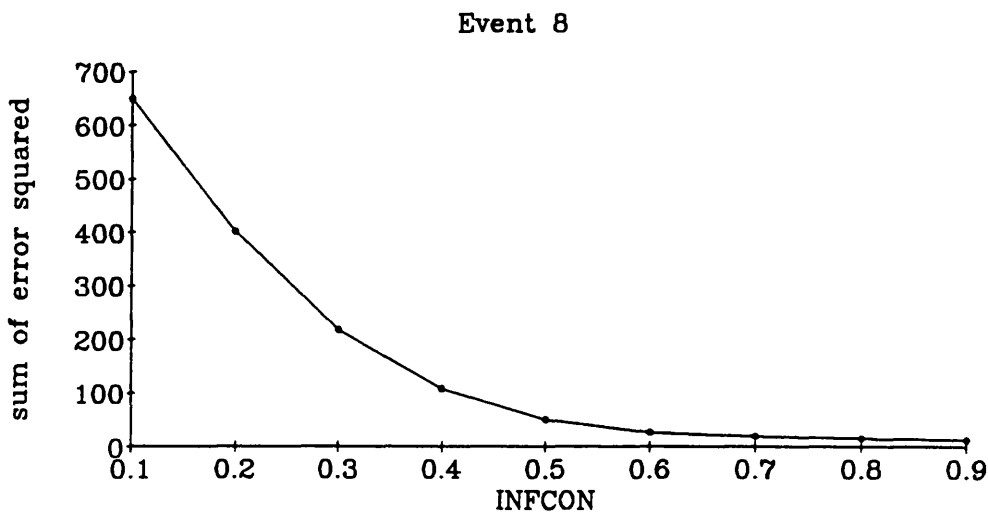
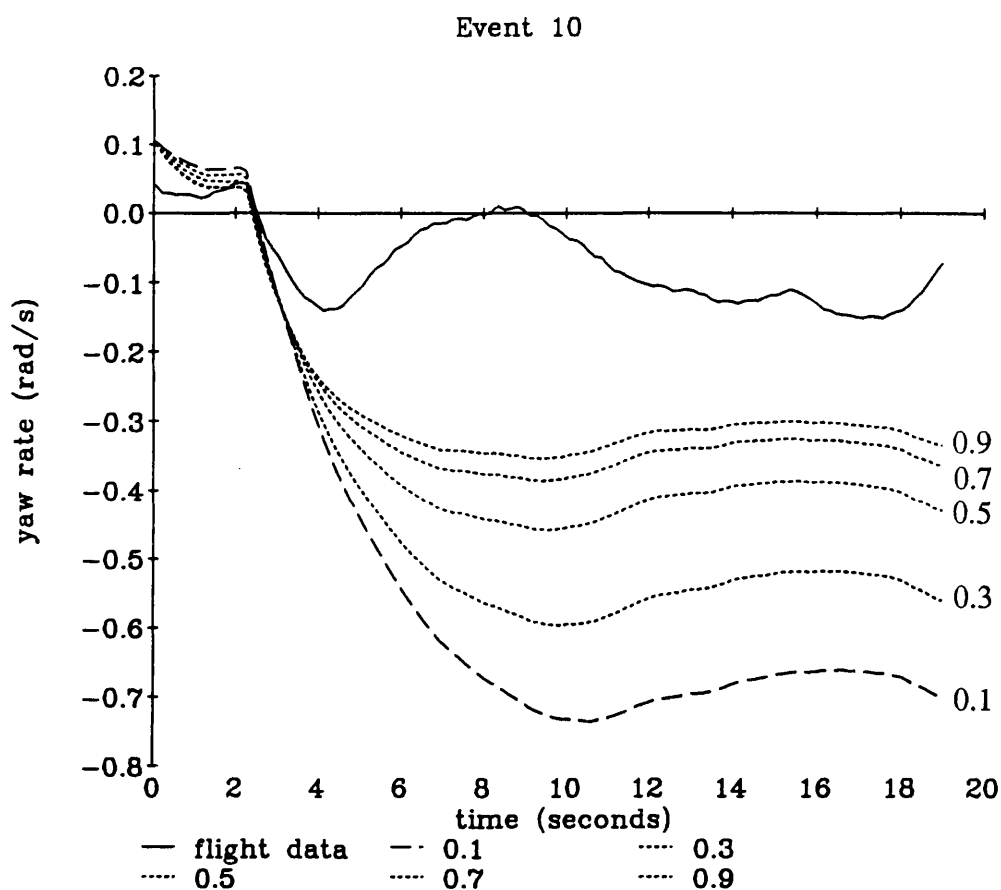


Figure 26 (cont.) Variation of parameter INFCON which represents the fraction of main rotor downwash contributing to tangential inflow at the tail rotor

Time history for variation of INFCON



Cost function for variation of INFCON

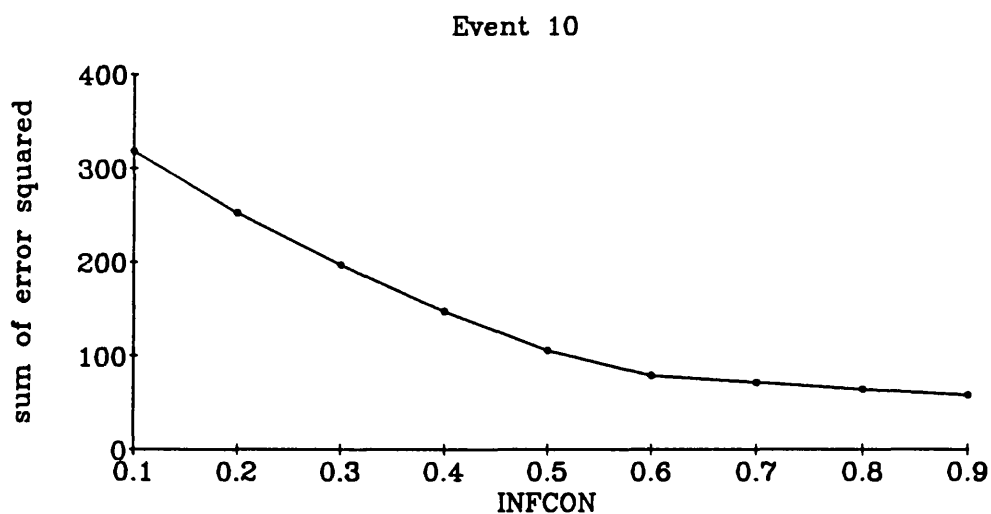


Figure 26 (cont.) Variation of parameter INFCON which represents the fraction of main rotor downwash contributing to tangential inflow at the tail rotor

Visual examination of the time response plots together with the plots of the sum of the squares of the error between flight data and model output implied that main rotor downwash/tail rotor interaction was important and a value of about 0.8 might be realistic. It is not possible to deduce a more accurate or definitive value for the parameter INFCON from the available information. This is because it can be seen that the model in its current state, although an improvement on the original model, could not be said to give an accurate representation of PUMA yaw dynamics.

5.5 Unmodelled cross-coupling dynamics from aircraft motion

Examination of other available flight data for this series of experiments, in particular aircraft roll rate, showed that although the pilot was working to maintain hover, the helicopter was moving and probably exciting other dynamic modes. A potential source of unmodelled yaw dynamics are perturbations in lateral acceleration. This is related to aircraft roll angle since in the hover, the lateral force provided by the tail rotor should be balanced by a lateral force from the main rotor (with the helicopter hovering right wheel down). Other influences include cross coupling between yaw, pitch and roll. Roll coupling due to gyroscopic precession is not negligible, because the tail rotor has a high rate of rotation.

5.6 Conclusions

The inclusion of pitch flap coupling and main rotor downwash in the yaw dynamics model considerably improves the ability of the model to predict yaw rate in hover turns although the extent of the effect of the downwash could not be precisely determined. Tail boom drag proved insignificant in hover but weathervaning could be important in forward flight.

Wind could not be considered a contributory factor from the above results although the yaw sensitivity to wind speed is high and the flight data did appear to exhibit a periodic motion which correlated with aircraft heading. Further flight experiments of the same type, hover turns, are necessary to investigate the effect of wind blowing the main rotor downwash over the tail rotor with wind speed and direction during the experiment recorded as part of the flight data.

The fit of the model to the flight data is still inaccurate for most of the experiments. It is probable that some of these inaccuracies come from unmodelled dynamics due to the helicopter not being in a perfect hover. These inaccuracies could perhaps be eliminated by using the experimental flight data of the aircraft body states to force the model in an open loop simulation¹⁰ thereby eliminating cross coupling effects.

References

- 1 Gaffey, T.M., '*The effect of positive pitch-flap coupling (negative δ_3) on rotor blade motion stability and flapping*', Journal of the Americal Helicopter Society, Vol. 14, No. 2, pp49-67, 1969.
- 2 Lynn, R.R, Robinson, F.D., Batra, N.N. and Duhon, J.M., '*Tail rotor design part 1: aerodynamics*', Journal of the American Helicopter Society, Vol. 15, No. 4, pp2-15, 1970.
- 3 Riley, J., Padfield, G.D. and Smith, J., '*Estimation of rotor blade incidence and blade deformation from the measurement of pressures and strains in flight*', Fourteenth European Rotorcraft Forum, Paper No. 110, 1988.
- 4 Kaletka, J., '*Instrumentation and data processing*', AGARD LS 178, October 1991.
- 5 Padfield, G.D., '*A theoretical model of helicopter flight mechanics for application to piloted simulation*', RAE Technical Report, TR 81048, 1981.
- 6 Padfield, G.D., '*A theoretical model of helicopter flight mechanics for application to piloted simulation*', RAE Technical Report, TR 81048, pp 13, 1981.
- 7 TSIM, Cambridge control Ltd.
- 8 Taylor, J.W.R. (ed), '*Jane's all the worlds aircraft 1976-77*', pp41-42, 1977.
- 9 Prouty, R.W., '*Helicopter performance, stability and control*', PWS publishing, pp102-109, 1986.
- 10 Padfield, G.D. and DuVal, R.W., '*Application areas for rotorcraft system identification : Simulation model validation*', AGARD LS 178, October 1991.

Chapter 6

Validation of a helicopter main rotor model

6.1 Introduction

A helicopter is a complex nonlinear system. It is very useful to be able to focus on specific areas of such a system by splitting it into subsystems. An important system on a helicopter is the main rotor. It provides the lift and control for the helicopter. The main rotor is the source of many of the dynamic modes on a helicopter and is therefore a crucial part of any helicopter model.

In this chapter, a nonlinear physical model of a helicopter rotor will be investigated. Parametric validation techniques will be applied to two areas of the model - the lag damper and the engine and rotor speed models. These techniques of Maximum Likelihood parameter estimation and sensitivity coefficient examination will be shown to be useful model validation methods for this type of problem.

6.2 The rotor model

The main rotor provides the thrust and lift for the helicopter. On the Aerospatiale PUMA, it consists of four blades rotating at 265 rpm, clockwise when viewed from above. Each blade is of aerofoil cross section and is effectively a wing which generates lift and drag as air flows over it.

The pilot controls the pitch of the rotor blades both collectively for changes in overall lift and cyclically where pitch is increased at one side of the rotor relative to the other producing an uneven lift distribution. This generates a turning moment allowing the helicopter to be manoeuvred.

The rate of rotation of the rotor blades is an important design parameter. It wants to be as high as possible while keeping the advancing blade tip speed at the helicopter's maximum forward velocity below the speed of sound. During flight, the forces on the rotor in the plane of rotation vary considerably, and it is necessary to actively control engine power to maintain a constant rotor speed. On the PUMA this is done using an automatic control system as shown in figure 27.

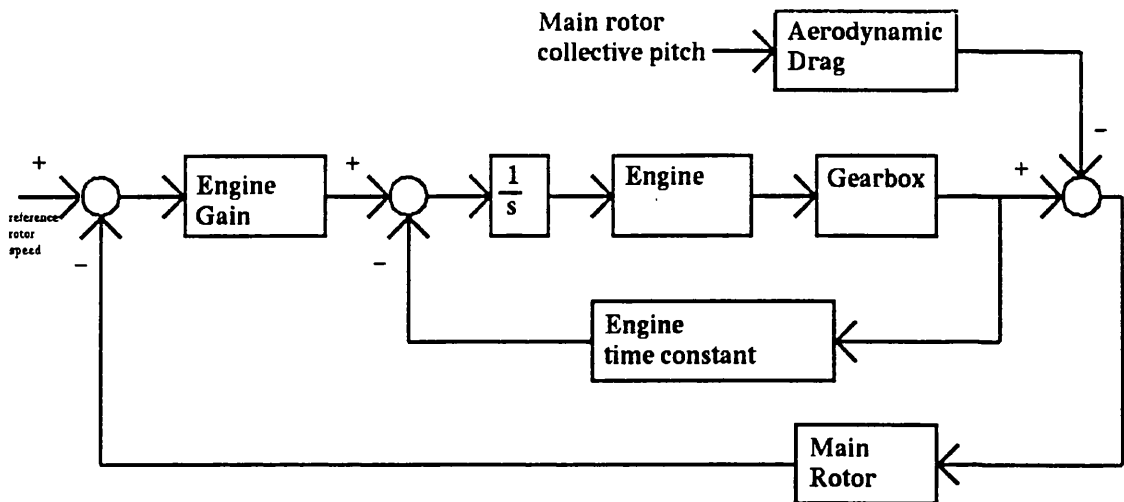
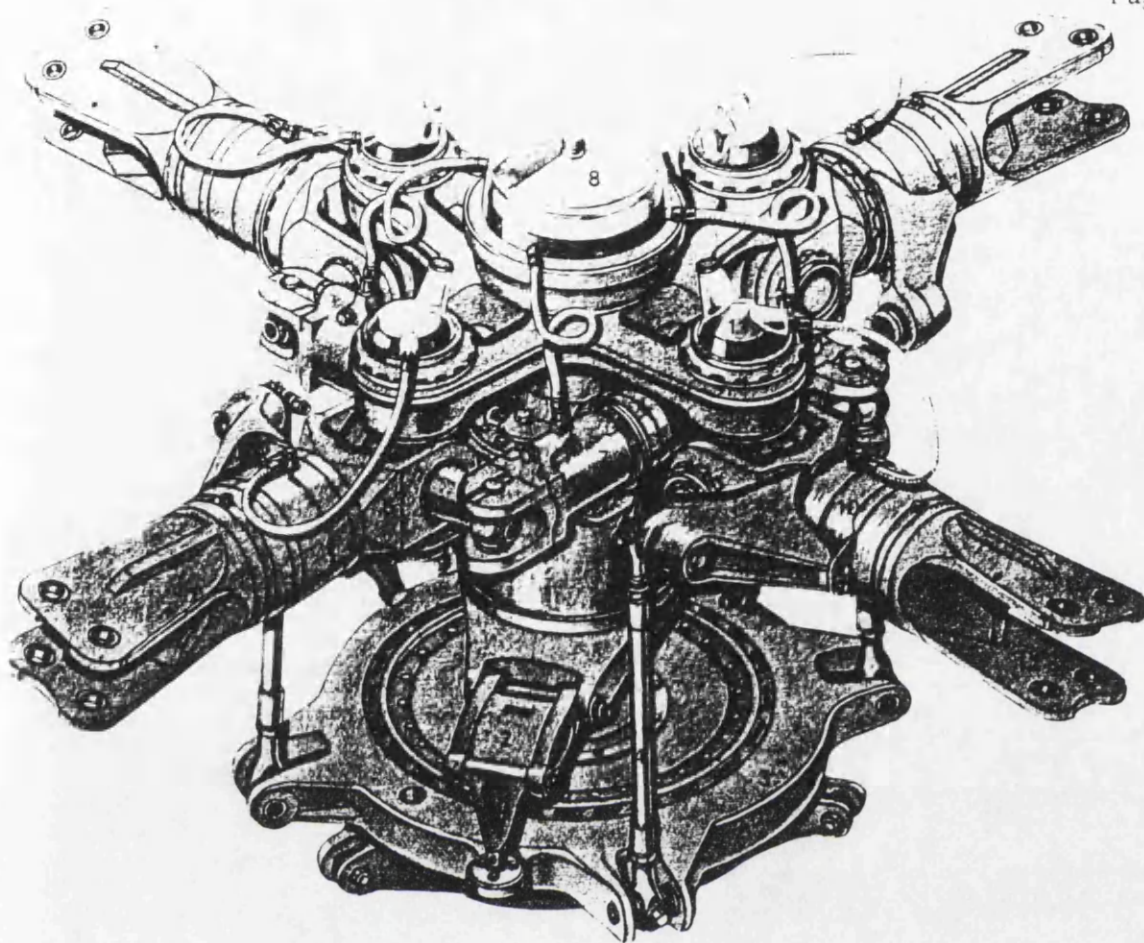


Figure 27 Block diagram of helicopter engine, gearbox and rotor speed controller

The controller parameters are engine time constant and engine gain.

The lift and drag forces generated on each rotor blade vary depending on the airflow over that blade. This means that the blade will tend to flap up and down and backwards and forwards relative to the plane of rotation. To prevent excessive stress levels in the blade root, the rotor blades are designed so as to be allowed to flap and lag. On the PUMA this is done using fully articulated rotor blades. That is blades which are hinged to allow free movement in the flap and lag degrees of freedom. A diagram of the rotor head on an Aerospatiale PUMA is given in figure 28.



Key to diagram

- 1) Lower swashplate
- 2) Scissors Link
- 3) Upper swashplate
- 4) Connecting rod
- 5) Flapping hinge
- 6) Drag hinge
- 7) Hydraulic lag damper
- 8) Drag hinge hydraulic reservoir
- 9) Pitch-change arm
- 10) Pitch-change bearing
- 11) Pitch-change bearing lubrication reservoir
- 12) Blade attachment point

Figure 28 Rotor hub of an Aerospatiale SA330 PUMA helicopter (from Gunston1)

The relevant forces on each blade are the weight of the blade pulling it down, aerodynamic lift pulling it up, aerodynamic drag pulling it back and centripetal acceleration pushing the blade to a straight out position. These forces combine to produce the flap and lag modes. The flap mode has natural damping unlike the lag mode which is underdamped. The lag natural frequency is often near the rotor frequency so resonance is potentially a large problem. A common solution is to put a mechanical damper at the blade root in the lag degree of freedom. In the PUMA, the lag damper is in the form of a dashpot which can be seen in figure 28.

When modelling a helicopter rotor, it is necessary to consider inflow. Inflow is the velocity of the air moving down through the main rotor. It affects the aerodynamic forces generated at the blades and is generated by the downwash from these blades. It can be modelled as a constant, as an instantaneously changing value (assuming no air mass) or as a dynamic variable depending on the complexity of the model. Because of the difficulty of measuring inflow, identifiability is a big problem with the validation of inflow models².

The rotor model used in this chapter accounts for rotor speed dynamics, blade dynamics and inflow and consists of six nonlinear differential equations of which two are second order representing five degrees of freedom - dynamic inflow, blade flap, blade lag, heave velocity and rotor speed. The equations are

$$\dot{v}_i = \left[\frac{3}{16} a c N \frac{\Omega}{\pi R} \left(\frac{2}{3} \Omega R \theta_0 - v_i + w - \frac{2}{3} R \dot{\beta} \right) - \frac{3}{2R} \left(\frac{1}{1 + \frac{a\sigma}{16v_i}} \right) \frac{v_i}{k} \left(\frac{v_i}{k} - w + \frac{2}{3} R \dot{\beta} \right) \right] \frac{1}{0.637 C_0} \quad (1).$$

$$\begin{aligned} \ddot{\beta} = & \dot{w} \frac{M_\beta}{I_\beta} + 2\Omega \dot{\zeta} \beta + \beta \dot{\zeta} \dot{\Omega} \\ & - \dot{\beta} (\Omega - \dot{\zeta}) \frac{1}{8} + \frac{w}{6R} (\Omega - \dot{\zeta}) - \frac{v_i}{6R} (\Omega - \dot{\zeta}) + \theta_0 (\Omega - \dot{\zeta})^2 \frac{1}{8} \\ & - \Omega^2 \left(1 + hR \frac{M_\beta}{I_\beta} \right) \beta \end{aligned} \quad (2).$$

$$\begin{aligned} \ddot{\zeta} = & 2\beta \dot{\zeta} \dot{\beta} - 2\Omega \beta \dot{\beta} + \dot{\Omega} \left(1 + hR \frac{M_\beta}{I_\beta} \right) - \Omega^2 \zeta hR \frac{M_\beta}{I_\beta} + \frac{\gamma}{6R} (\Omega - \dot{\zeta}) (v_i - w) \theta_0 + \frac{\gamma}{8} (\Omega - \dot{\zeta}) \dot{\beta} \theta_0 \\ & - \frac{\gamma}{4R^2} (v_i - w)^2 - \frac{\gamma}{3R} (v_i - w) \dot{\beta} - \frac{\gamma}{8} \dot{\beta}^2 + \frac{\gamma}{8a} \delta_0 (\Omega - \dot{\zeta})^2 - A_\zeta \dot{\zeta} \end{aligned} \quad (3).$$

$$\begin{aligned} \dot{\Omega} = & \frac{N}{I_R} \left[(I_\beta + hRM_\beta) (\ddot{\zeta} + 2\dot{\zeta} \zeta (\Omega - \dot{\zeta})) + hRM_\beta \zeta (2\Omega \dot{\zeta} - \beta \dot{\beta} - \dot{\beta}^2 - \dot{\zeta}^2) \right] \\ & - \left[\frac{4}{3} \rho \pi R^3 \dot{v}_i + \frac{2\rho \pi R^2}{\left(1 + \frac{a\sigma \Omega R}{16v_i} \right)} \frac{v_i}{k} \left(\frac{v_i}{k} - w + \frac{2}{3} R \dot{\beta} \right) \right] \frac{(v_i - w)}{\Omega I_R} - \rho \pi R^4 \Omega^3 \frac{\left(\delta_0 \frac{\sigma}{8} \right)}{\Omega I_R} \\ & + \frac{Q}{I_R} - temp4 \end{aligned} \quad (4).$$

$$\dot{Q} = G (\Omega - \Omega_0) + T_e Q \quad (5).$$

$$\begin{aligned} \dot{w} = & \Omega \dot{\beta} N I_\beta \frac{\gamma}{6mR} + \frac{\ddot{\beta} N M_\beta K_{M_\beta}}{m} + \Omega v_i N I_\beta \frac{\gamma}{4mR^2} - \Omega w N I_\beta \frac{\gamma}{4mR^2} \\ & - \Omega^2 \theta_0 N I_\beta \frac{\gamma}{6mR} + \frac{1}{2} \rho (v_i - w)^2 \frac{S_{fp} C_{df}}{m} + 9.81 \end{aligned} \quad (6).$$

where,

a	6	aerodynamic lift curve slope
$A\zeta$	25	Lag damper damping coefficient
c	5.401	rotor blade chord
C_0	1	dynamic inflow apparent air mass
c_{df}	1.5	main rotor downforce drag coefficient
G	-44480	engine controller gain
I_β	1189	moment of inertia of rotor
k	1.4	inflow momentum correction factor
$K_{M\beta}$	0.7	moment correction factor
m	5400	aircraft mass
M_β	249	flap moment of inertia of one blade
N	4	number of blades
Q		engine torque
R	7.498	blade radius
s_{fp}	1.767	fuselage area exposed to main rotor downforce
T_e	-1.0	engine controller time constant
$temp4$		trim variable in rotor speed equation to allow for unmodelled engine torque losses in the tail rotor and gearbox
v_i		inflow velocity
w		aircraft heave velocity
β		coning angle
δ_0	0.012	profile drag coefficient
γ	10.18	Lock number
θ_0		blade collective pitch
ρ		air density determined from flight trim condition
σ	0.0917	main rotor solidity
Ω		rotor speed
Ω_0	27.76	trim rotor speed
ζ		blade lag angle

The model was derived by A. T. McCallum and S. S. Houston of the Defence Research Agency at Bedford. It is derived from expressions in Padfield³, Houston⁴ and Blackwell, Feik and Perrin⁵.

The rotor speed equation (equation 6.4) is based on momentum theory while the inflow, blade dynamics and heave equation are based on blade element theory.

Identified areas of uncertainty in the model are the lag damper and the engine model. An explicit lag damper representation had been a recent addition to the model and had not yet been validated. The engine and controller model is a first order approximation and as such should be validated to verify that it is an adequate representation.

This model is very nonlinear. That is, any linearised representation would change significantly different with even a small change in operating point. This means that more conventional validation techniques using linearised models would be of limited use. It is necessary therefore to validate this model as a full nonlinear representation using nonlinear model validation techniques. The methods chosen were parameter estimation and sensitivity coefficient examination. An important criterion for choosing such methods is the availability of relevant flight data. Parameters selected for parameter estimation must be identifiable with measurable output variables. For example, lag damper parameters could be estimated because of the availability of blade lag angle data and the high sensitivity of blade lag angle to the lag damper parameters.

This chapter can also be considered an investigation of the validation methods used.

6.3 Parameter Estimation based Model Validation Methods

The rotor model is constructed from physical nonlinear differential equations. Areas of uncertainty being investigated are the lag damper and the engine model using blade lag angle and rotor speed data respectively with a maximum likelihood based technique of parameter estimation.

The parameter estimation process requires sensitivity coefficients to be generated. Sensitivity coefficients are necessary for the minimisation of the parameter estimation cost function. They can also be a very useful aid to model validation if examined visually as part of the parameter estimation process (see chapter 3).

Given that an output error parameter estimation technique is to be used, there are still many possible variations. The experimental data to be used and the parameters to be estimated must be selected. Selection of experimental data includes choice of output

variables as well as the specific experiment. Concerning the experiment, the test input signal should excite the part of the model being validated while having as little effect as possible on the rest of the system. The complete data record may contain a trimmed steady state at the start of the record before the input is applied and recovery after the system response to the aircraft. It is therefore necessary to select the portion of the data to be used in the estimation process.

It may be advantageous to pre-process or filter the data in some way before the parameter estimation. This should however be approached with caution as it is better not to alter the data if possible. Any filtering would have to be accounted for in the model validation process.

The results of the parameter estimation include Cramer-Rao bounds which are estimates of the relative variance of the different parameters. These parameter estimates can be compared with physical values or with results from different experimental data sets. Good agreement is an indication of model validity.

6.3.1 Investigation of model structure

Parameter estimation can also be used to investigate changes in model structure. If changing the structure of a part of the model has an effect on the estimated parameters and Cramer-Rao bounds on a related parameter set, parametric techniques can be used to validate that model structure. For example, if changing the structure significantly increases the size of the Cramer-Rao bounds, the likelihood of that structure being a more accurate representation of the system is decreased.

If a parameter estimate varies considerably with different data sets and these data sets are from similar experiments, it is probable that the parameter does not represent a physical quantity and the structure of that part of the model should be re-examined. Agreement for different data sets on the other hand, suggests that the structure is correct and the results of the parameter estimation can provide a value for the parameter if it is not easily discernible from theory.

6.4 Sensitivity methods

The generation of sensitivity coefficients is necessary for the parameter estimation process. The sensitivity coefficients are derived analytically in this research for

accuracy and because of difficulties with a finite difference technique and nonlinearities (see chapter 3, section 3.2). Initially, the sensitivity coefficients can be used to select the output variables to be used and to detect any linear dependence between the parameters. This is where the effect of one parameter on an output is the same as the effect of another parameter and it causes problems in the parameter estimation process.

Visual examination of the sensitivity coefficient time histories can give valuable insight to the model and parameters and how the parameters affect the model.

6.5 Maximum Likelihood Parameter Estimation - Theory

Parameter estimation is a statistical technique where a cost function involving the model and experimental data is postulated and then minimised with respect to the parameters. The maximum likelihood method is based on a cost function where the object is to maximise the probability of fitting the model output to flight data given the parameter set. The method can eliminate bias errors due to state and measurement noise for linear or linearised systems and is described in Iliff⁶, from which the following equations are taken.

The first step is to define the system.

$$\dot{x}(t) = f[x(t), u(t), \xi] \quad (6.9),$$

$$z(t_i) = g[x(t_i), u(t_i), \xi] + G(\xi)\eta_i \quad (6.10),$$

where,

x = dynamic system variable,

\dot{x} = time derivative of x ,

t = time,

t_i = time at interval number i ,

f = arbitrary nonlinear function,

g = arbitrary nonlinear function,

ξ = parameter vector (to be estimated),

z = model output,

G = weighting matrix,

η_i = measurement noise.

The maximum likelihood method used here is for a complete nonlinear model so it is not possible to implement the Kalman Filter necessary to account for state noise.

The maximum likelihood cost function for estimation with no bias errors due to measurement noise is,

$$J(\xi) = \frac{1}{2} \sum_{i=1}^N [z(t_i) - \tilde{z}_\xi(t_i)]^T (GG^T)^{-1} [z(t_i) - \tilde{z}_\xi(t_i)] + \frac{1}{2} N \log \|(GG^T)\| \quad (6.11).$$

where,

z = model output,

N = number of data points in simulation run.

To minimise the cost function J , a Gauss-Newton algorithm is used. The parameters are adjusted according to equation 6.12.

$$\hat{\xi}_{L+1} = \hat{\xi}_L - [\nabla^2_\xi J(\hat{\xi}_L)]^{-1} [\nabla_\xi^T J(\hat{\xi}_L)] \quad (6.12),$$

where,

L = iteration number.

The first and second derivatives are,

$$\nabla_\xi J(\xi) = - \sum_{i=1}^N [z(t_i) - \tilde{z}_\xi(t_i)]^T (GG^T)^{-1} [\nabla_\xi \tilde{z}_\xi(t_i)] \quad (6.13),$$

$$\nabla^2_\xi J(\xi) \equiv \sum_{i=1}^N [\nabla_\xi \tilde{z}_\xi(t_i)]^T (GG^T)^{-1} [\nabla_\xi \tilde{z}_\xi(t_i)] \quad (6.14).$$

The Cramer-Rao bound represents the uncertainty of the parameter value and is calculated by taking the information matrix, inverting it, and taking the square root of the corresponding diagonal element. That is,

$$H^{-1} = 2J_{\min} \left\{ \sum_{i=1}^N [\nabla_\xi \tilde{z}_\xi(t_i)]^T (GG^T)^{-1} [\nabla_\xi \tilde{z}_\xi(t_i)] \right\}^{-1} / (N-1) \quad (6.15),$$

$$C-R_i = \sqrt{H^{-1}(i,i)} \quad (6.16),$$

where C-R is the Cramer-Rao bound and i is the parameter number.

The application of maximum likelihood estimation in this thesis is for single output systems only so the weighting matrix $(GG^T)^{-1}$ is unimportant and can be set to one. This simplifies the estimator equations.

6.6 Experimental

The rotor model was to be validated as a subsystem. That is, without any representation of non-rotor states with the exception of heave which is vertical velocity. For this type of approach to be successful, it is important that unmodelled dynamic modes are not excited during the experiment. This was achieved in the helicopter flight experiments by designing the experiment to be a step response to main rotor collective pitch from a hover condition resulting in aircraft motion mainly in the vertical axis.

Two sets of flight data were available. A 2 degree step upwards and a 1 degree step downwards, both measured at the blade root. The experiments were flown using the Defence Research Agency at Bedford research PUMA helicopter. Recorded rotor data included individual blade flap and lag angles, blade pitch, engine output torque, rotor speed and aircraft vertical acceleration. The data were recorded using the aircraft's on board data acquisition system (Section 5.2 in this thesis). The individual blade data were averaged to give overall values since in the hover there should be little or no cyclic pitch or flap variation. The blade angle data was sampled at a rate of 1024 Hz. Rotor speed was sampled at once per rotor revolution or about 4.4 Hz. This was considered a possible source of problems in the engine model validation and is considered in section 6.8.3.

The step response of the rotor model using the blade collective pitch data shown in the figures, as the model input is shown in figures 29 and 30 respectively for flight experiments 1 and 2 together with the relevant flight data.

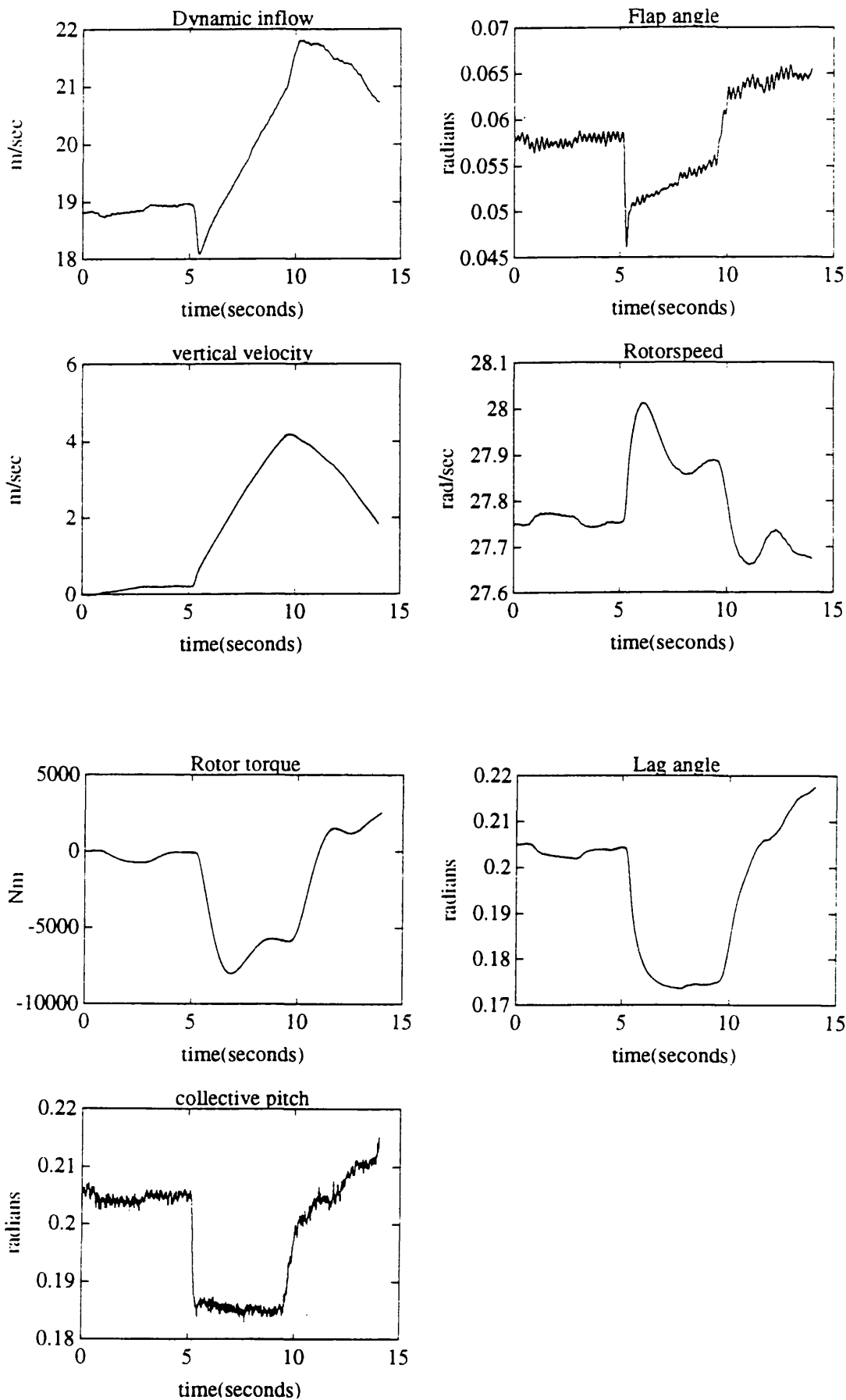


Figure 29 Time response of nonlinear rotor model to the main rotor collective step inputs (shown at bottom of figure) used in flight experiment 1

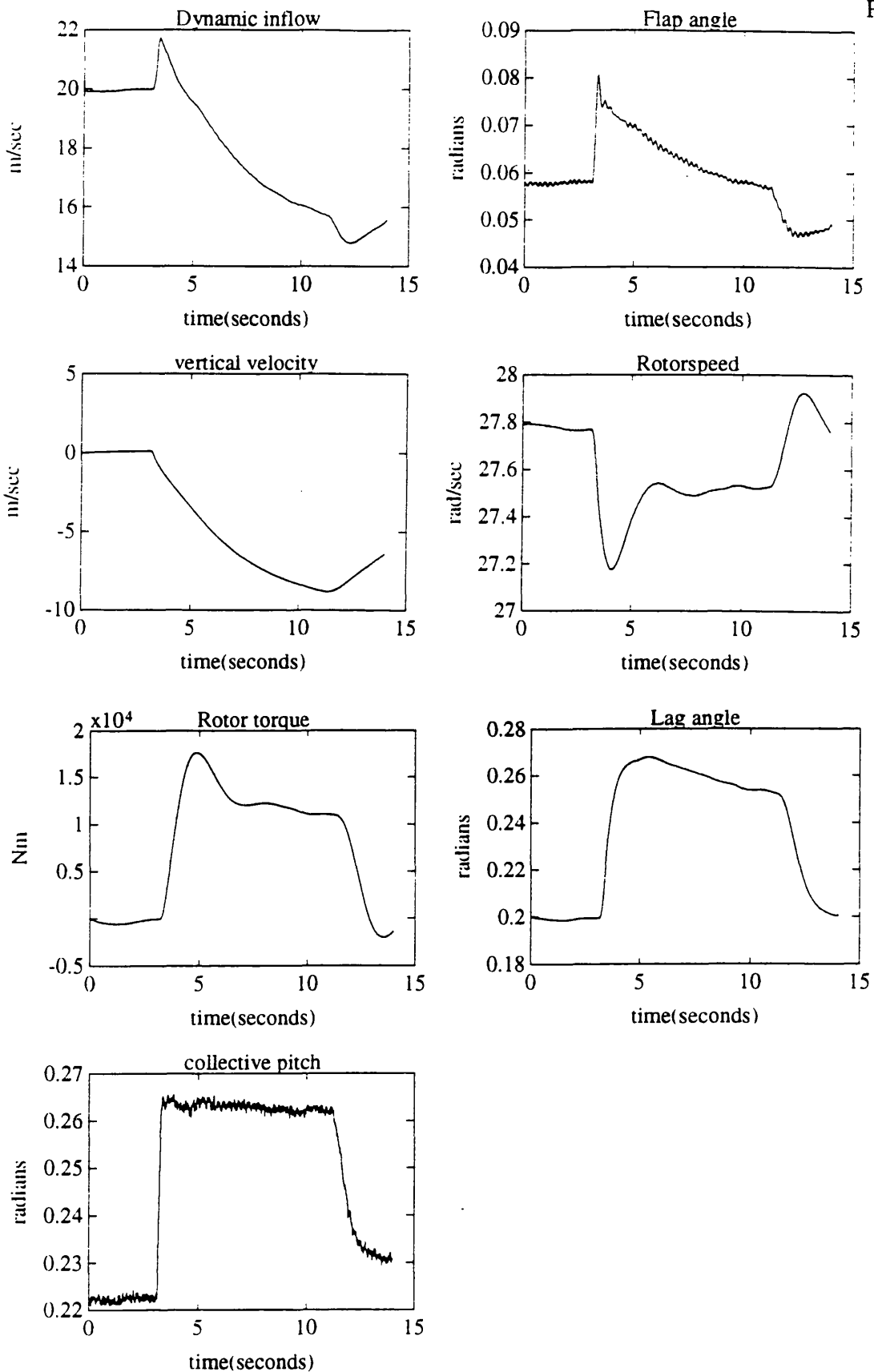


Figure 30 Time response of nonlinear rotor model to the main rotor collective step inputs (shown at bottom of figure) used in flight experiment 2

6.6.1 Computer implementation of parameter estimation

The parameter estimation process requires the model to be simulated to produce the output variable z and the sensitivity coefficients. The system equations were differentiated manually to derive the sensitivity equations for each parameter (see chapter 3 section 3.2). This cosystem was simulated alongside the main model.

The system was simulated using TSIM⁷, a FORTRAN based continuous time simulation program. The equations are integrated using a fourth order Runge-Kutta algorithm with a step size of 0.0078125 seconds which is equivalent to a 128 Hz sampling rate. This is much higher than any of the dynamic modes in the model. The simulation package TSIM allows integration of FORTRAN routines into the model. This is how the parameter estimation routines were integrated with the simulation. TSIM also has graphical output capabilities facilitating analogue matching and examination of sensitivity coefficient time histories. The software was used on a Digital MicroVax II.

6.7 Lag damper model validation

The original model from DRA Bedford (section 6.2) had one term to represent the lag damper. This was the damping coefficient A_{ζ} . It had been set at 25 as a result of some rig tests (section 6.7.1).

When a mathematical model is changed by for example, adding an extra term, it is important that the new model is validated. In this case, the model was to be validated using parameter estimation and sensitivity coefficients. This was because of the parametric nature of the problem. As well as the damping term, two other terms were added to the equation (equation 6.6) to represent a spring term and friction within the damper. This gave an overall expression for the damper as.

$$\ddot{\zeta} = \dots - A_{\zeta} \dot{\zeta} - C_{\zeta} \zeta - D * \text{sign}(\dot{\zeta})$$

... represents the blade lag dynamics not including the lag damper (centrifugal, coriolis, inertia and gravitational forces).

- ζ represents rotor blade lag angle
- A_ζ represents the viscous damping coefficient. A_ζ had the highest sensitivity coefficient.
- C_ζ represents a spring constant. The most likely source of the springiness is the fluid in the damper being compressed. The low sensitivity of this parameter suggests that it is not of great importance.
- D represents friction present in the damper - hence the nonlinear signum function. Friction works against you whichever direction you are moving in.

The sensitivity coefficients were evaluated as single value sensitivity coefficients (chapter 3, section 3.3) showed that the effect of each parameter on the blade lag angle was significantly larger than that on any other model output (see table 2 and figure 31), so blade angle data was used for the parameter estimation.

output variable	sensitivity (%age change in output for 1% change in $A\zeta$)
flap angle	0.0025%
vertical acceleration	0.0025%
rotor speed	0.000075%
engine torque	0.001%
lag angle	0.05%

Table 2 Single value sensitivity coefficients for lag damper parameter $A\zeta$

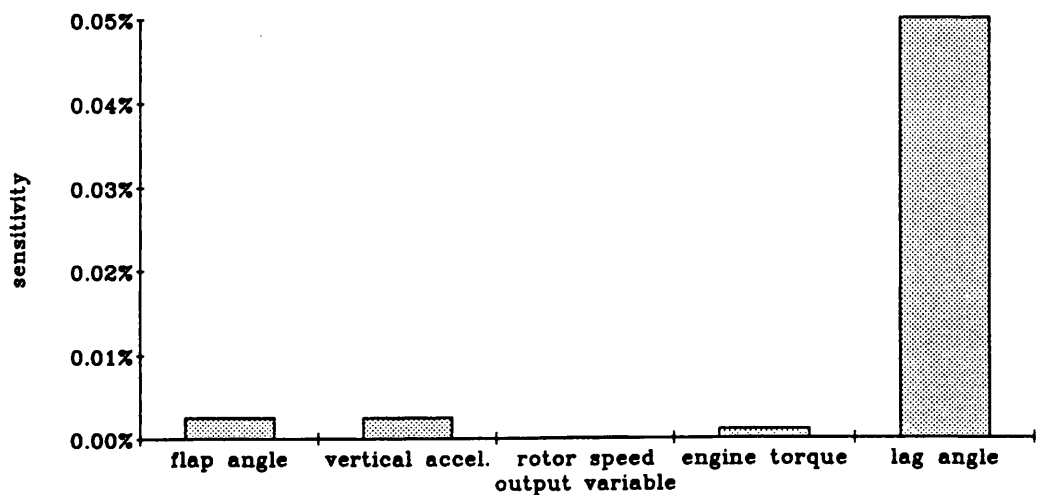


Figure 31 Bar chart showing relative sensitivities of output variables to lag damper damping coefficient

Before the parameter estimation, it was necessary to trim the model. That is, to ensure that the initial conditions of the output variable matched the first few seconds of the flight data record representing trimmed flight in the hover before application of the experimental input. The simulation program used, TSIM⁷, had an inbuilt trim routine which solved the trim equations (state derivatives equal to zero) for selected input parameters. The main parameters used were temp4 and air density. Temp4 was part of the rotor speed equation and represented unmodelled aerodynamic forces on the blades as well as losses in the transmission system.

Air density was not recorded during the flight experiments and it was used as a trim variable. This was done because it was noticed that, of the two experiments, both started from the hover but with different values for collective pitch. The experiments took place one after the other on the same flight. After ruling out a change in mass due to fuel burn as the cause of the collective pitch angle discrepancy (the change in mass was not nearly sufficient), it was concluded that a change in altitude and therefore air density was the only reasonable explanation. It is not unreasonable that the helicopter would change altitude between experiments because it is important that such experiments are done in clean air (no wind or turbulence) and it is sometimes necessary to move in order to find some. Another possibility is that the pilot was complying with an air traffic control instruction to change altitude because of traffic. The trimmed values of air density were checked and the equivalent altitude checked in all cases to verify that the altitude was reasonable for this aircraft.

6.7.1 Parameter estimation results

The three parameters were estimated using the flight data for lag angle up to the end of the step input in each case. A fourth order Runge Kutta algorithm was used to integrate the nonlinear equations. Sensitivity coefficients were calculated from exact sensitivity equations derived analytically.

	<u>experiment 1</u>	<u>experiment 2</u>
A_{ζ}	44.6 (0.46)	36.3 (0.55)
C_{ζ}	-0.34 (0.0017)	-1.71 (0.031)
D	-0.13 (0.0045)	0.059 (0.0087)

Cramer-Rao bounds are given in brackets.

As can be seen, the values for A are relatively similar but C_{ζ} and D vary considerably. This suggests that although the parameter estimation algorithm does improve the fit of model output to flight data by estimating values for C_{ζ} and D, these values for C_{ζ} and D do not represent real lag damper parameters. C_{ζ} and D are numerically unidentifiable.

It was decided that the most acceptable representation of the lag damper would be the single parameter A_{ζ} . A_{ζ} was estimated on its own for both experiments.

The results were,

	<u>experiment 1</u>	<u>experiment 2</u>
A_{ζ}	34.8 (0.54)	30.9 (0.91)

It can be seen that the results are reasonably close. This increases confidence in the lag damper model. The theoretical value of this parameter was derived by mechanical testing of a damper on a test rig (section 6.7.2) and gave a value of 25 for A_{ζ} .

Graphical analysis gives a good indication of model validity. Figure 32 shows a time response of lag angle for both flight experiments with the original model, the estimated model and the flight data superimposed on each other. Figure 33 shows the effect of the estimated parameter from one experiment on the output of the model for the second experiment.

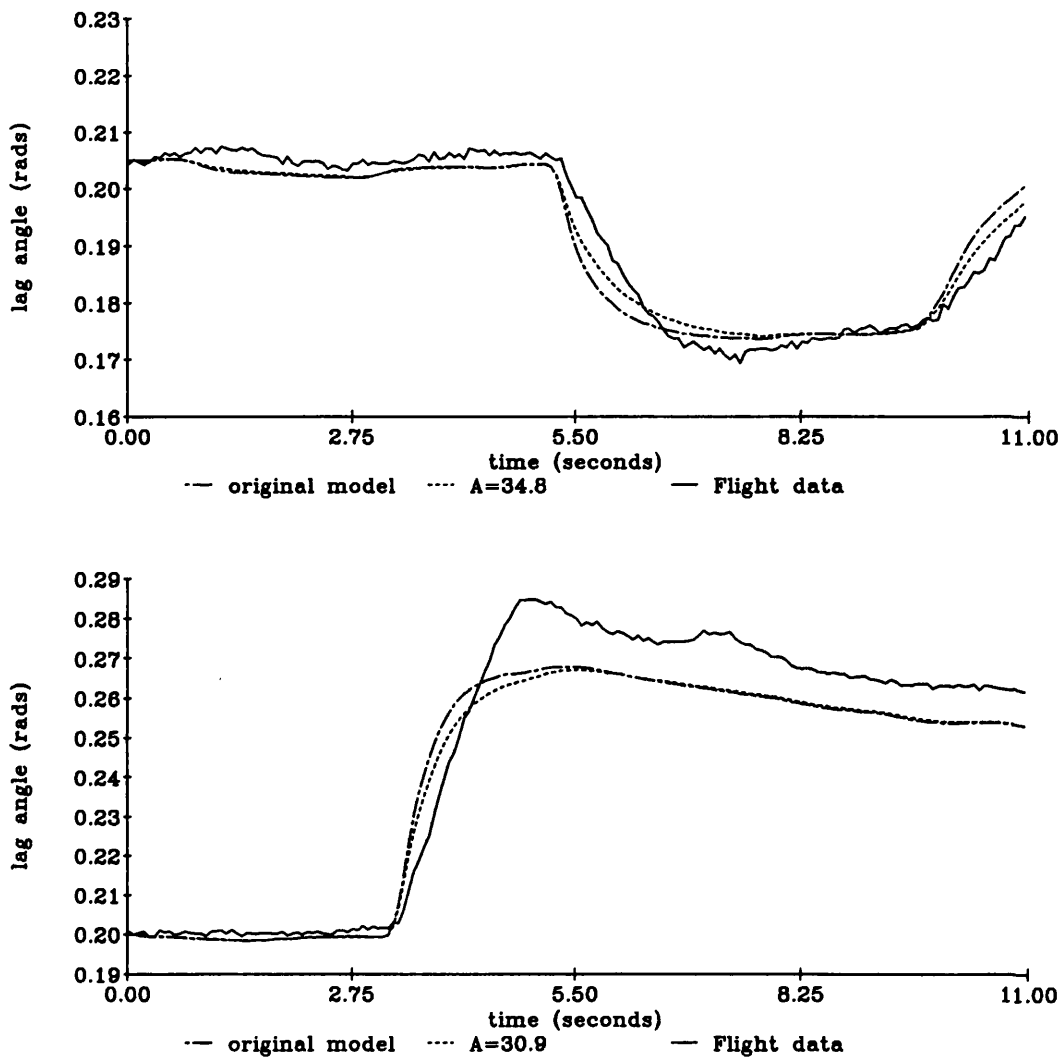


Figure 32 Time response of blade lag angle for flight experiments one and two with original model and estimated model

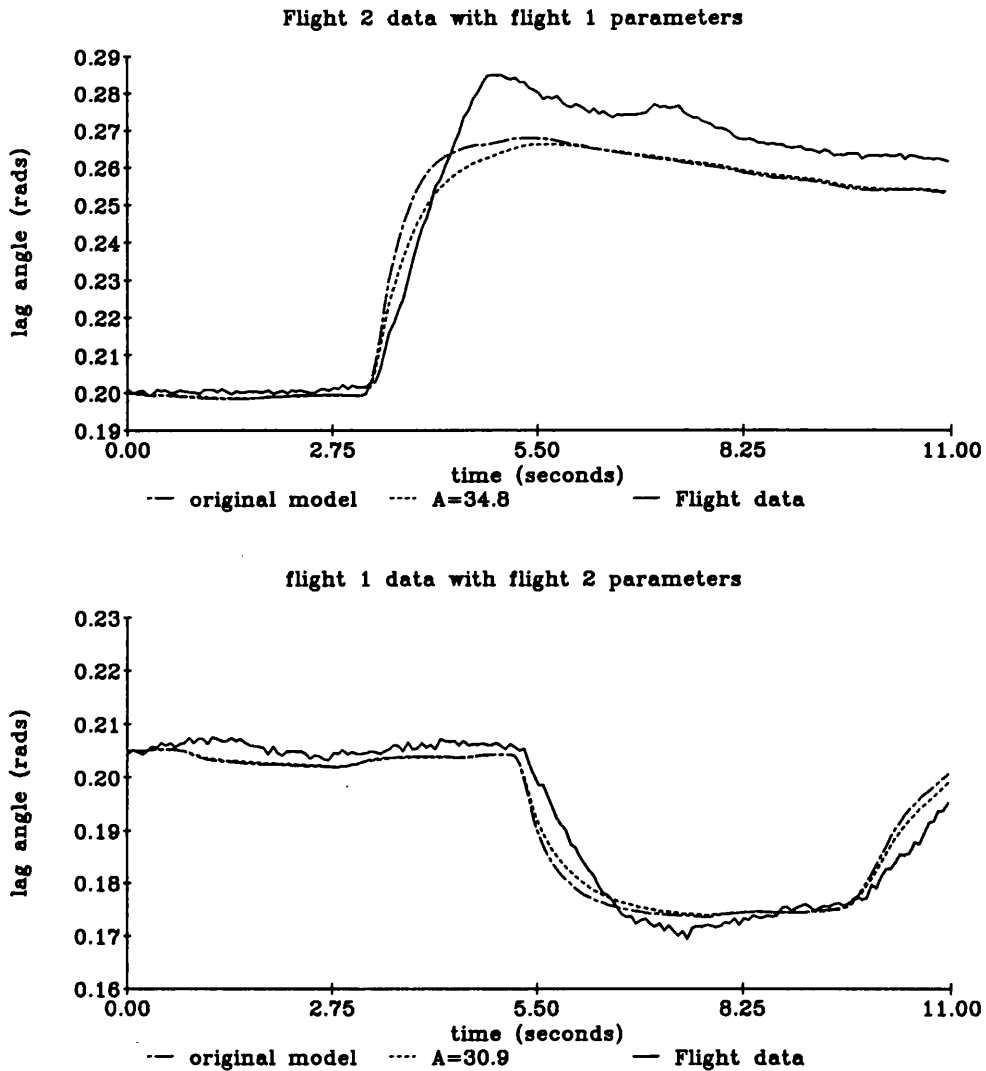


Figure 33 Time response of blade lag angle for one flight experiment with parameters estimated from other flight experiment and original model

It can be seen that the estimated value of A_z slightly improves the fit in each case and also that using the value estimated from one experiment with the data from the other experiment gives a better fit. This final stage of analogue matching with different data is important because it indicates that the fit is not simply the result of a statistical best fit.

6.7.2 Derivation of original value for lag damper damping coefficient

The lag damper of the Aerospatiale PUMA helicopter has been tested by the aircraft operators and manufacturers. The damper was removed from the helicopter and placed

in a mechanical test rig. It was then subjected to an input amplitude typical of its operational range at three different frequencies. Its response to this input was measured and the damping coefficient was calculated at 25 kg/s.

6.7.3 Lag damper validation - conclusions

From this validation exercise, it can be concluded that the best representation for the lag damper is one term in the damping coefficient A_{ζ} . The parameter estimation results suggest a value slightly higher than the physical value of 25 (section 6.7.2), but it would be necessary to do more flight experiments to verify this. A different experimental input might help to indicate why the estimate for A_{ζ} was high.

Maximum likelihood parameter estimation proved a useful technique for this validation exercise. The sensitivity coefficients were helpful in selecting the output variable for the identification.

6.8 Engine model validation

The main rotor model for the Aerospatiale Puma helicopter as described in section 6.2 includes a model of the engine and rotor speed dynamics of the helicopter. Examination of the fit between flight data and model output for rotor speed showed that this part of the rotor model did not give a good fit to flight data. Different model structures and parameter sets for the engine and rotor speed model were examined using both low sampling rate rotor speed flight data and digitally filtered rotor azimuth data at a much higher sampling frequency. Improvements to the model were suggested and validated with a second set of flight data.

6.8.1 Engine/Rotor speed model

Helicopter rotor speed is often modelled as a constant, and the engine speed is usually maintained at a constant value by an automatic control system. However, in the Puma helicopter which has an automatic control system controlling fuel flow, the rotor speed does vary transiently and it is likely that this variation is important when modelling large excursion manoeuvres⁵.

The rotor speed dynamics are represented in the model by two first order nonlinear differential equations,

$$\begin{aligned}
\dot{\Omega} = & \frac{N}{I_R} \left[(I_\beta + hRM_\beta) (\ddot{\zeta} + 2\dot{\zeta}(\Omega - \dot{\zeta})) + hRM_\beta \zeta (2\Omega\dot{\zeta} - \beta\dot{\beta} - \dot{\beta}^2 - \dot{\zeta}^2) \right] \\
& - \left[\frac{4}{3} \rho \pi R^3 \dot{v}_i + \frac{2\rho\pi R^2}{\left(1 + \frac{a\sigma\Omega R}{16v_i}\right)} \frac{v_i}{k} \left(\frac{v_i}{k} - w + \frac{2}{3} R\dot{\beta} \right) \right] \frac{(v_i - w)}{\Omega I_R} - \rho \pi R^4 \Omega^3 \frac{\left(\delta_0 \frac{\sigma}{8}\right)}{\Omega I_R} \\
& + \frac{Q}{I_R} - temp4
\end{aligned} \tag{6.4}$$

$$\dot{Q} = G(\Omega - \Omega_0) + T_e Q \tag{6.5}$$

The variables are as defined in section 6.2.

In equation 6.4 the dynamic variable is rotor speed. Equation 6.5, representing the engine and speed controller in which the dynamic variable is rotor torque. Flight data are available for both rotor speed and engine torque although the engine torque data includes tail rotor torque and transmission system losses not represented in the main rotor model. It was decided that rotor speed data alone should be used for this exercise since the difference between rotor torque and engine torque can be up to 15%.

The rotor speed equation uses momentum theory to calculate the variation of rate of rotation of the rotor as the aerodynamic load on the rotor varies. The rotor torque equation consists of a gain factor multiplied by rotor speed and a first order lag in the engine dynamics.

The theoretical parameter values for the engine and controller equation, equation 6.5, are gain, $G = -44480$, and time constant, $T_e = -1.00^3$.

There were two sets of flight data available for the engine model validation as with the lag damper. A useful rule of model validation and development is that half the data should be used for development and half the data used for checking the model⁸. This was done with the engine/rotor speed model validation. Data from flight experiment 1 were used for the model development and data from flight experiment 2 were used for the validation (analogue matching).

6.8.2 Parameter Estimation

The available flight data from experiment 1 consisted of a step response to main rotor collective in the hover. The recovery was also included in the experimental record but it was noted that during the recovery the helicopter had encountered some vortex ring flow which was not modelled. It was decided therefore to base the parameter estimation on that portion of the time history corresponding to the step and omit the recovery. A plot of the main rotor collective can be seen in figure 34.

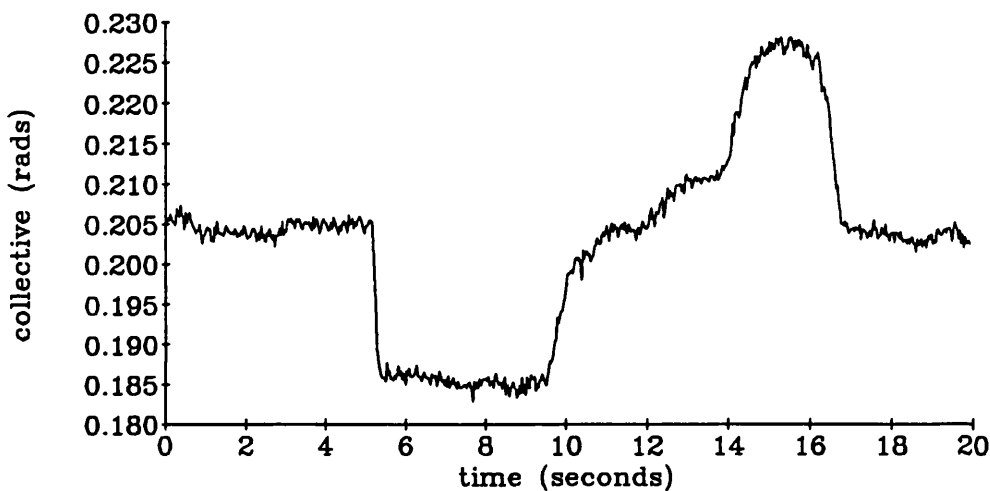


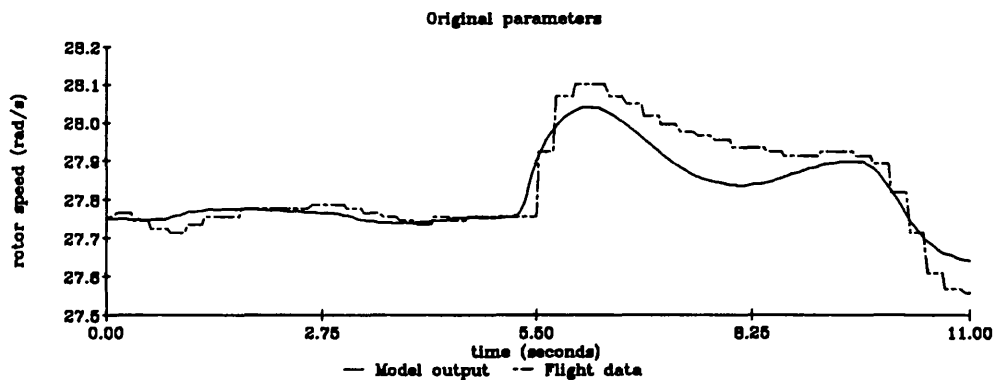
Figure 34 Time response of main rotor collective pitch for flight experiment 1 (system input)

The model was trimmed to the values in the first five seconds of the flight data by varying trim inflow and air density (see section 6.7). The model was then driven by the experimental record of the main rotor collective pitch. The maximum likelihood parameter estimation method defined in section 6.5 was used to estimate values for the parameters G and T_e in equation 6.5. Cramer-Rao bounds were also calculated to give an indication of the accuracy of the parameter estimate. These should not be taken as an absolute indication of the error magnitude but can be used for comparison between different estimations. The Cramer-Rao bound is given in brackets after the parameter estimate.

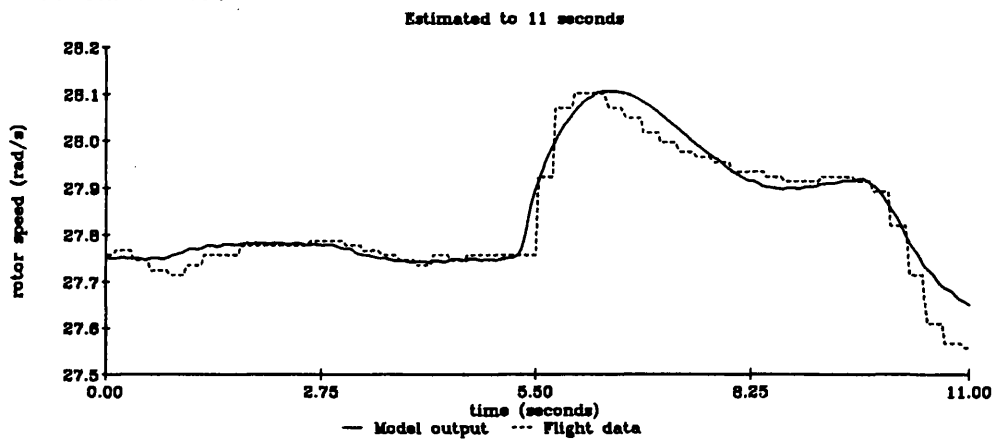
The sensitivity coefficients were initially calculated using a finite difference approach. This was because different model structures for rotor speed dynamics were to be tested and analytic sensitivity equations would take a considerable amount of time to

develop for each model. However, once a model structure had been decided upon, analytic sensitivity equations were used. This was because analytically derived sensitivity coefficients give much better convergence to the cost function minimum (chapter 3, section 3.2).

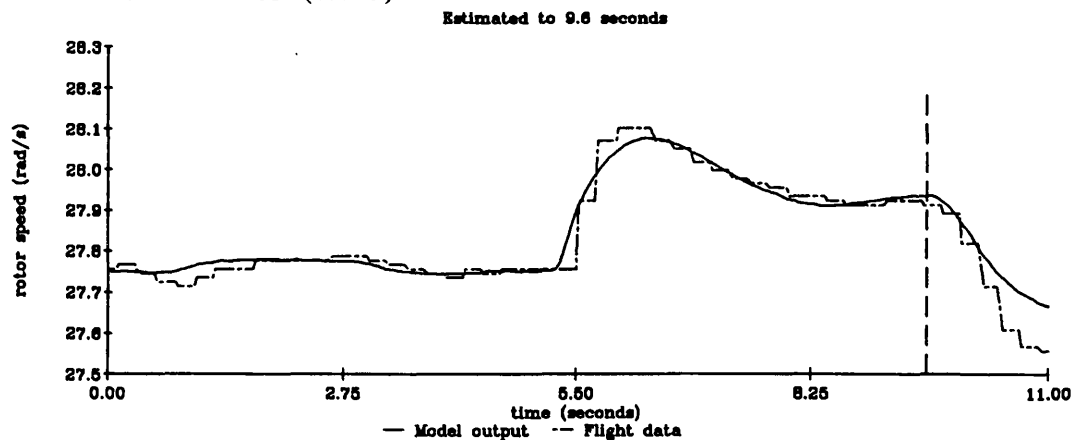
The parameters G and T_e , the gain and the time constant were estimated over a 9.6 second time history consisting of solely the step response and an 11 second time history including some of the recovery but not the vortex ring mode. The time responses are shown in figure 35 for the original parameters and for the estimated time responses over the two record lengths. The time response of the sensitivity coefficients at the transient for each of the two parameters is also shown. The sensitivity coefficient (figure 36) shows the effect each parameter has on the system transient response. It can be seen that the initial response is affected more by T_e and the 'double peak' of the response is affected by the engine gain, G . This technique of sensitivity coefficient examination can provide useful clues to the origins of discrepancies found when matching experimental data to model output.



Gain=-44500
Time constant=-1.00



Gain=-32200 (420)
Time constant=-0.983 (0.015)



Gain=-35700 (400)
Time constant=-1.160 (0.015)

Figure 35 Time response of model output and flight data of rotor speed for different engine parameter values with sensitivity coefficients

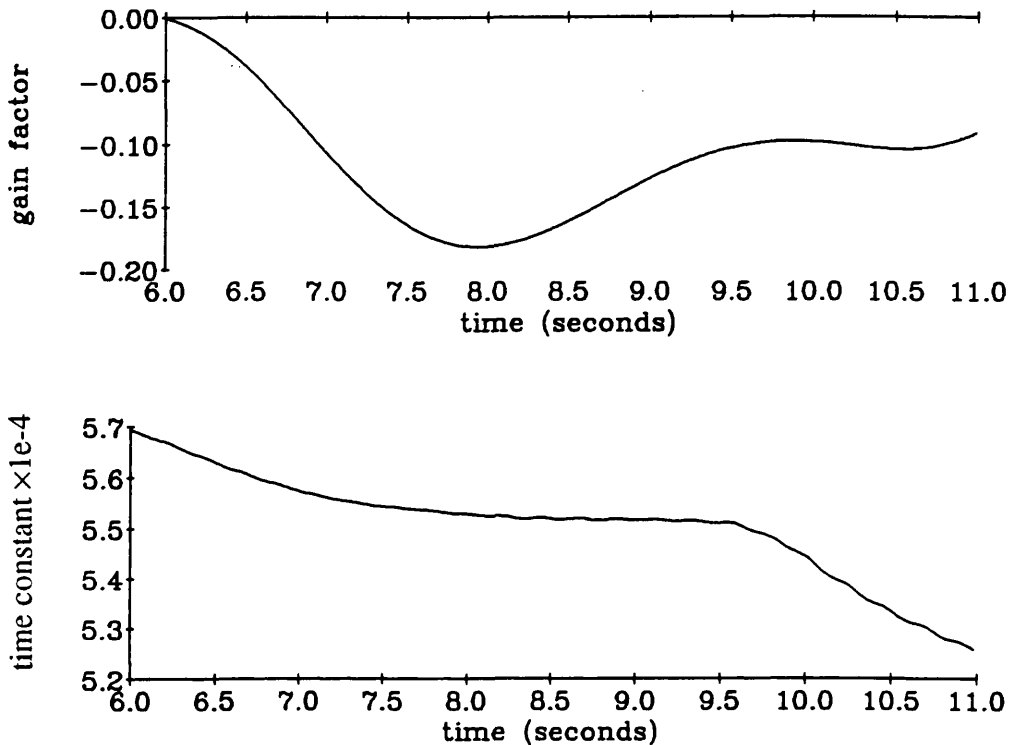


Figure 36 Sensitivity coefficients of engine parameters T_e and G over time history of system transient

Figure 35 illustrates the importance of the selection of the part of the time response over which to base the estimation. In this case it was decided to use a 9.6 second record length and to look only at the step response because this seemed to be a better representation of the system time response.

6.8.3 Effect of flight data sampling rate

It can be seen from figure 35 that the effective sampling frequency is very low, about 4.4 Hz. It is possible that such a low sampling frequency could affect the parameter estimation process, so an alternative source of rotor speed data was sought. This is because of the method of measurement of rotor speed on the Puma. A disk with a line painted along a radius is attached to the rotor shaft and the data acquisition system (chapter 5, section 5.2) records the time at which the line passes an optical sensor i.e. once for each revolution of the disk. This means that a value of rotor speed is calculated once per revolution of the rotor, about 4.4 times per second. A zero order hold is used by the data acquisition system to give a data sampling rate of 64 Hz.

The azimuth angle is also recorded. This is the actual angular displacement of the rotor disc from a reference. This is recorded at a much higher sampling rate, 1024 Hz, and theoretically, it should be possible to differentiate the signal to get a record of rotor speed. If the azimuth signal is differentiated directly, the resulting time history is too noisy to be of any use for parameter estimation. It is therefore necessary to filter the data.

This was done by subtracting the part of the data corresponding to a constant trim rotor speed using a least squares algorithm giving a perturbation data record. This perturbation data was then filtered using a low pass filter before numerically differentiating and adding to the DC trim value calculated in the least squares fit. It was realised that a filter with a well defined cutoff would be required. The cutoff frequency was selected as 1 Hz. This was above the frequency of the modes of interest but would remove any rotor frequency noise. A fifth order Butterworth filter was used with the frequency response shown in figure 37.

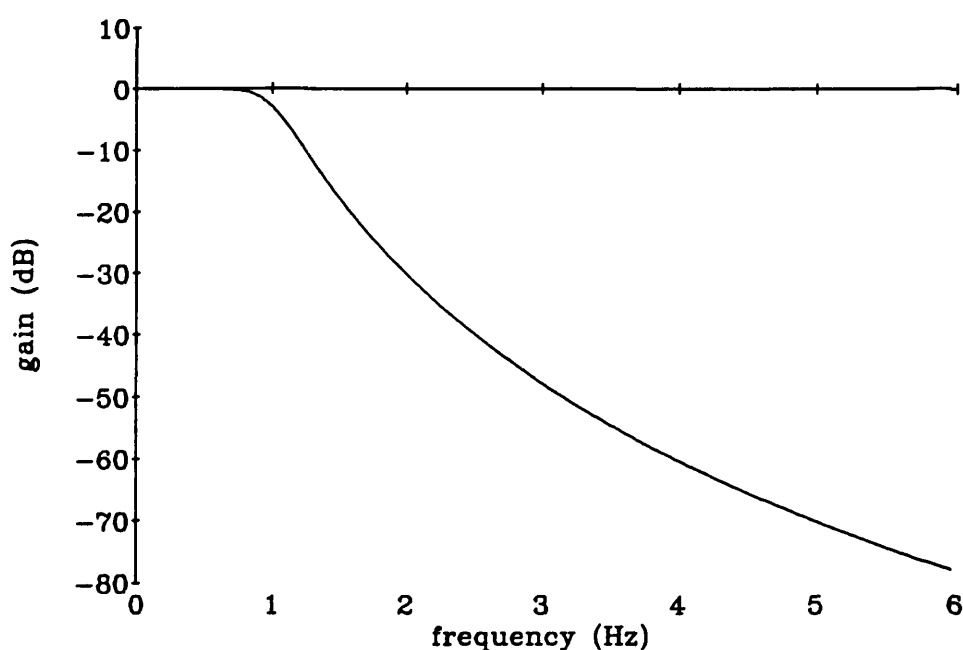


Figure 37 Frequency response of low pass filter used on differentiated azimuth data to represent rotor speed

The data were filtered using the `Filtfilt` command in PRO-MATLAB¹ which includes forward-reverse filtering for zero phase shift.

The time history of rotor speed resulting from processing the azimuth data is shown compared with the rotor speed record in figure 38. The main differences are the magnitude of the peak at the beginning of the transient and an effective time delay in the low sampling rate data.

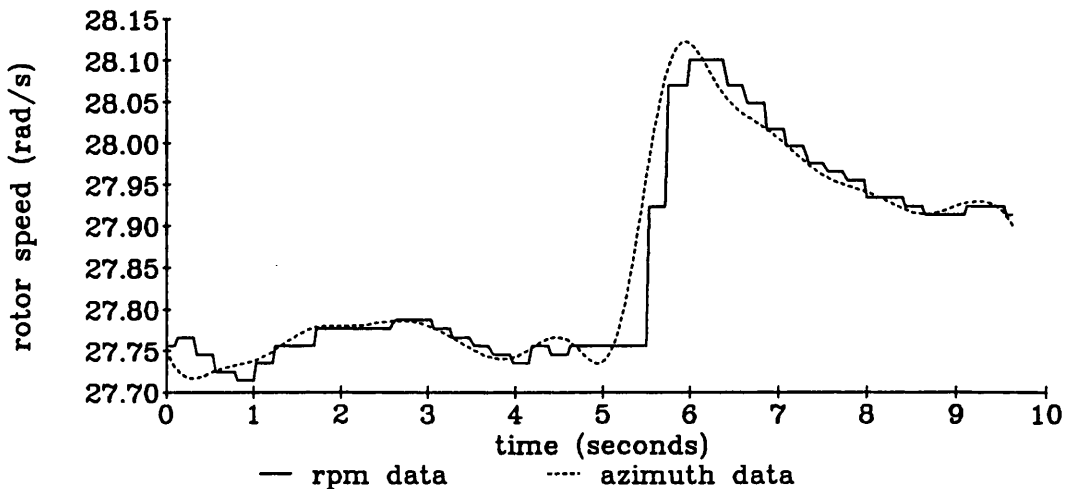
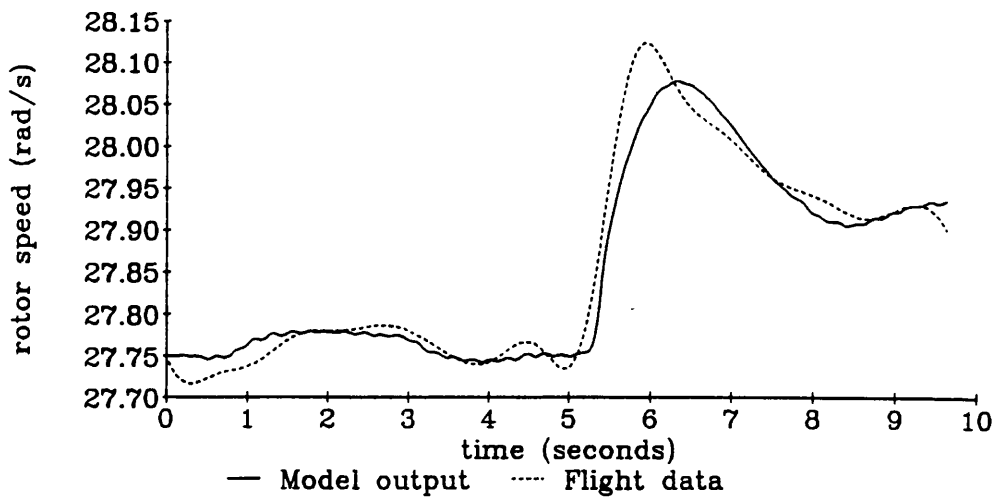


Figure 38 Comparison of rotor speed data with differentiated and filtered azimuth data

Figure 38 shows good correlation between the rotor speed and azimuth flight data records.

The increased magnitude of the peak for the azimuth data made a difference in the model to flight data fit. It is probable that the azimuth record gives a more accurate representation of this peak because it can be seen how the low frequency sampling would reduce its magnitude. Another effect of low frequency sampling with a zero order hold is to introduce a $T/2$ time delay where T is the sampling period of the data.

The two engine parameters were estimated over a time period of 9.6 seconds using the filtered and differentiated azimuth data to represent rotor speed. The results are shown in figure 39.



Gain=-38690 (570)

Time constant=-1.235 (0.021)

Figure 39 Time response of rotor model where engine parameters have been estimated using differentiated azimuth data

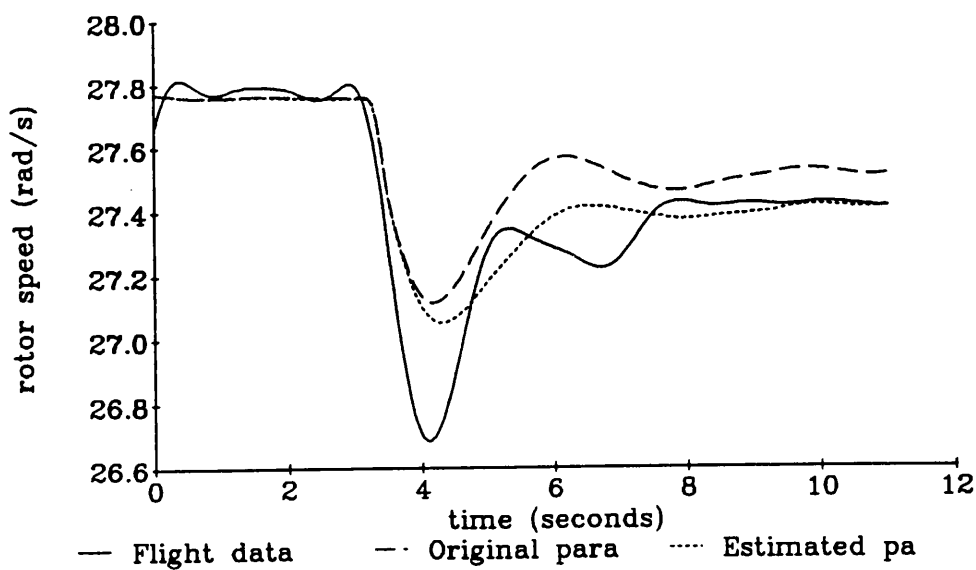
The Cramer-Rao bounds show these results to be worse than those achieved for the low sampling rate rotor speed (figure 35) although it can be seen that the azimuth record is effectively noise free and therefore likely to be more accurate because of its higher sampling rate and method of measurement. The increased Cramer-Rao bounds could be the result of an increased error squared term caused by the size of the initial transient peak at time = 5.5 seconds.

6.8.4 Time delays in the system

It has been suggested that time delays could be more prevalent in flight data than previously thought¹⁰. To test whether this time delay would have any effect on the parameter estimation process, the time delay was estimated as a parameter. An analytic expression for the sensitivity coefficient of a time delay was calculated and found to be the derivative of the input to the time delay (chapter 3, section 3.2). This was implemented and a value for the time delay was estimated. This value was small and varied considerably for different record lengths and different sets of flight data giving positive and negative results. The conclusion was that it was not possible to identify a time delay from the flight data.

6.8.5 Engine model validation - conclusions

The low sampling rate of the flight data does not make much difference to the fit although its limitations are revealed in the size of the initial transient peak. The parameter estimates for both sources of flight data are reasonably close to each other if slightly different from the original theoretically derived values of $G=-44480$ and $T_e=-1.0$ calculated from simplified engine equations³. Taking the azimuth derived data as the more accurate, the parameter values of $G=-38690$ and $T_e=-1.235$ were used for a simulation with a second set of flight data as the second part of the model development and validation process. The results of this analogue matching comparison (figure 40) show that the fit is better than that achieved with the original parameters.



Gain=-38690

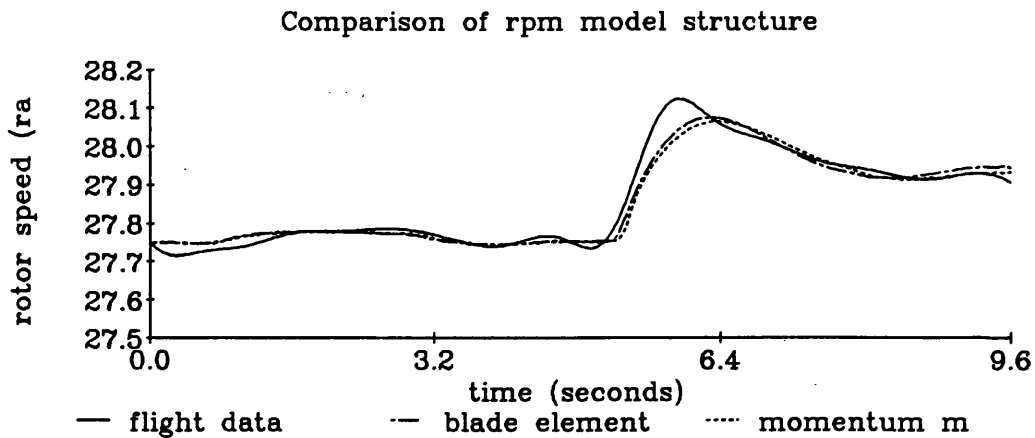
Time constant=-1.235

Figure 40 Flight data and experimental input from flight experiment 2 with parameter values estimated with data from experiment 1

6.9 Investigation of an alternative model structure for rotor speed dynamics

The aerodynamic load on the rotor is calculated using inflow and momentum dynamics in the rotor speed dynamics equation (equation 6.4). The rest of the rotor model (the flap and lag dynamics) use blade element theory. It was decided to represent rotor speed dynamics using blade element dynamics to calculate the aerodynamic load for the rotor speed equation. The linearised expressions for lift and drag at a rotor blade element were analytically integrated along the rotor blade to give an expression for the total aerodynamic torque on the main rotor. This expression was summed with engine torque to give a differential equation for rotor speed.

The blade element model was implemented and tested by estimating the engine parameters and comparing the time response of the model with flight data. Figure 41 shows the time response of rotor speed calculated using a blade element model superimposed on the time response of rotor speed using the momentum method to calculate rotor speed. The time responses are plotted for the theoretical parameter values and parameter values estimated with the blade element model. It can be seen from the Cramer-Rao bounds as well as the fit, that the momentum/inflow model and the blade element model give a very similar response.



Momentum model

Gain=-38690 (570)

Time constant=-1.235 (0.021)

Blade element model

Gain=-41690 (540)

Time constant=-1.244 (0.019)

Figure 41 Time response of rotor model for two model structures for rotor speed dynamics

One accepted technique for model structure validation is to plot the residuals against time. The residuals are defined as the difference between model output and experimental data. If the model structure is correct, the residual plot should show zero mean. Any peaks indicate faults with the model structure. The residual plots for the inflow momentum model and the blade element model are shown in figure 42.

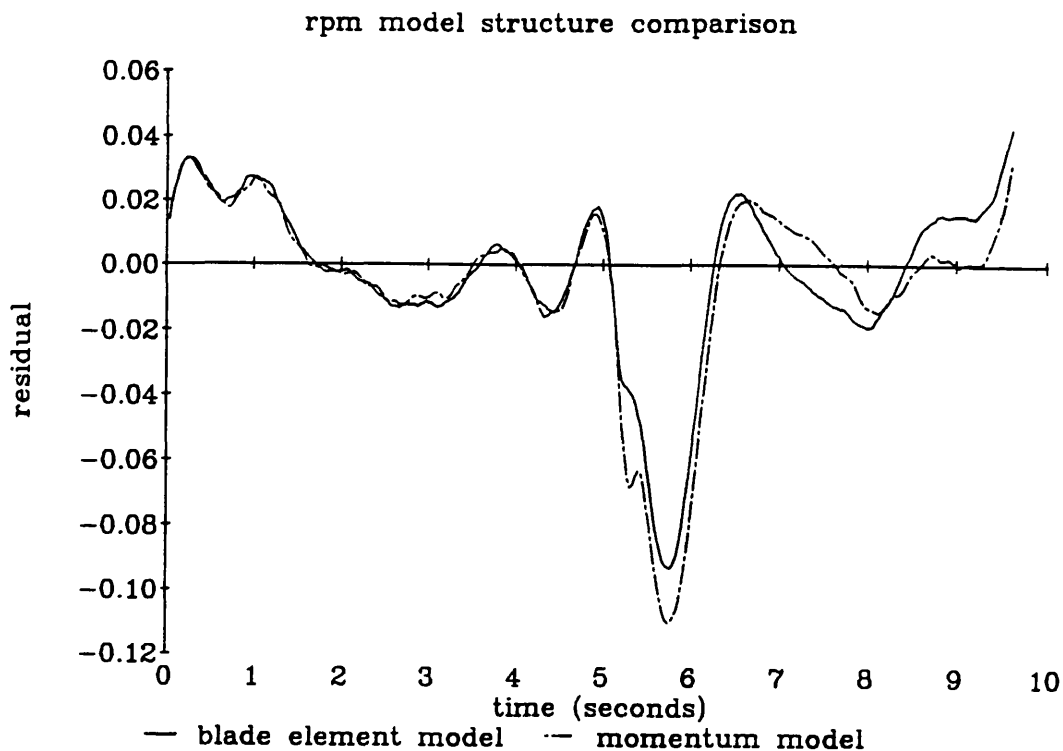


Figure 42 Comparison of residuals for the two different model structures for rotor speed dynamics

Figure 42 also shows the similarity of the models. It reveals a possible structure fault in the model at the start of the transient indicating a starting point for and future development work on the model.

The momentum/inflow rotor speed dynamics equation is based on the dynamic inflow equation which uses blade element theory. This could explain why the blade element results and the momentum results are so close.

6.10 Investigation of the effect of blade twist

On the Aerospatiale PUMA helicopter, the rotor blades are twisted so that the angle of attack at the tip of the blade is 8.5 degrees less than at the root. This is necessary because the airspeed at the blade tip is much higher and therefore more lift is generated at a given angle of attack. If the angle of attack was constant along the blade, the blades would tend to bend up at the tips causing structural problems.

The blade twist was incorporated into the rotor speed blade element equation. However, when an attempt was made to simulate the model, it was impossible to trim the model in the hover with the experimental flight data. It was deduced that the reason for this was that the rest of the rotor model although derived from blade element theory, did not include expressions for blade twist and if twist was to be included in the model at all, it would need to be in all the rotor equations.

This is a large task but worth considering as a next stage for the development of this rotor model.

6.11 Conclusions

Parameter estimation is a valuable technique for model validation. It has been applied to two problems in this chapter, the lag damper and the engine and controller model. It was concluded that one term, the damping coefficient, was necessary to represent the lag damper within the rotor model. Its value, and the values of the engine parameters were estimated to be slightly different from the original theoretically derived parameter values, but more experimental data would be required to justify changing the parameter values from the theoretical value.

Sensitivity coefficients were also used as part of the model validation process. By comparing the estimates with other estimates and with theoretical values, the model was validated. Analytically derived sensitivity coefficients were used to improve convergence in the parameter estimation process, a particular problem with nonlinear models.

It is also very useful to examine sensitivity coefficient time responses alongside time responses of the model and experimental data because this can give valuable physical insight into the problem and the effects of individual parameters on the model. This

technique was used with the engine parameters. Single value sensitivity coefficients proved useful in the initial stages of the parameter estimation to select the output variables and flight data to be used in the cost function. They can also be useful for parameter selection prior to estimation (see chapter 3, section 3.3).

Analogue matching of time responses to experimental data from a different experiment proved a useful validation method and was successfully applied to the rotor model.

The combination of analogue matching, sensitivity coefficient examination and parameter estimation is a powerful simulation based validation tool. But physical knowledge of the system is also required if full advantage of these methods is to be taken.

References

- 1 Gunston, B., 'Aircraft', pp199, Sundial publications Ltd, 1978.
- 2 Leith, D.J., Bradley, R., and Murray-Smith, D.J., 'The identification of coupled flapping/inflow models for hovering flight', Seventeenth European Rotorcraft Forum, Paper No. 91-45, 1991.
- 3 Padfield, G.D., 'A theoretical model of helicopter flight mechanics for application to piloted simulation', RAE Technical report 81048, 1981.
- 4 Houston, S.S. and Black, C.G., 'On the identifiability of helicopter models incorporating higher order dynamics', Fifteenth European Rotorcraft Forum, Paper 67, 1989.
- 5 Blackwell, J., Feik, R.A. and Perrin, R.H., 'Identification of rotor dynamic effects in flight data', Fifteenth European Rotorcraft Forum, paper no. 68, 1989.
- 6 Iliff, K.W., 'Parameter estimation for flight vehicles', Journal of Guidance and control, Vol. 12, No. 5, pp 609-622, 1989.
- 7 TSIM, Cambridge Control, Ltd.
- 8 Murray-Smith, D.J., 'Problems and prospects in the validation of dynamic models', to be presented at the Fourth International Symposium on Systems Analysis and Simulation, Berlin, August 1992.

- 9 FiltFilt command, in Little, J. and Shure, L., '*Signal processing toolbox user's guide*', The Mathworks, pp2_58-2_60, 1988.
- 10 Blackwell, J., and Perrin, R.A., '*Identification of Time Delays in Flight Measurements*', Journal Guidance, Control and Dynamics, Vol 14, No 1, pp 132-139, Jan 1991.

Chapter 7

An Interactive Model Validation Computer Program

7.1 Introduction

Many model validation methods have been investigated and evaluated in this research. These methods have all been suitable for, and applied to nonlinear physical models of a helicopter but are equally valid for nonlinear physical models of any dynamic system.

All the approaches considered are similar in that they have been based on continuous time simulation runs of the models using numerical integration. Although all the methods are useful techniques in their own right, they are at their most valuable when applied to the problem in a coordinated fashion as part of an integrated strategy. An important part of that strategy is physical knowledge of the system being modelled. Physical knowledge improves interpretation of the validation results and together with the validation tools, provides the basis for a comprehensive model development and validation methodology.

It is desirable to have an integrated approach to validation and development of nonlinear physical models. A computer package with a graphical output and a selection of validation tools would be an ideal addition to the development and validation process. This chapter will specify such a computer package using the model validation tools which have been developed and investigated in this thesis. It will also suggest how such a package could be implemented and applied.

7.2 Specification for a model validation and development package

7.2.1 User interface

The object of a computer program like this is to facilitate the use of model validation techniques in an integrated environment. A good user interface is therefore very important. A windows based system would be easy to use and familiar to most computer users. Such a system is one in which different programs are run in windows on the screen display. Information can be copied from one window to another and

when a window is not in use it can be shrunk to an icon. Such a system can be easily driven by mouse with pull down menus presenting program options. A high quality graphical output would be required for such a program and graphics hard copy output would also be necessary.

7.2.2 System simulation

The basic model would be represented by differential equations or transfer functions which would be integrated to simulate the system and generate time response data. It should also be possible to enter the model in graphical form such as a block diagram or signal-flow graph. The equations which make up the model should be easily accessible. It should be possible to view and edit these equations in a 'model' window. It should be possible to linearise the model to state space or transfer function form (where appropriate) and to generate eigenvalues of the system. It would be necessary to input the operating point or a range of operating points and it should be possible to plot variations of linearised parameters or eigenvalues against operating point.

The fundamental function of the program will be simulation in the time domain. Continuous time simulation is the basis of all the validation methods investigated in this thesis. It should also be possible for the user to select simulation parameters such as integration algorithm, step size, system input (from recorded data or otherwise), trim conditions or parameters, and the length of the simulation. It should be possible to plot time responses of selected variables in the model separately or superimposed on other time responses or experimental data. It should be possible for these time responses to be stored on disk or processed by filtering or converting to the frequency domain. It should also be possible to vary individual system parameters and to plot system time responses for a variety of parameter values as with the yaw model validation in chapter 5.

7.2.3 Parameter Estimation

Parameter estimation is an important validation method and should be an integral part of this validation program. The parameter estimation routine would allow a chosen parameter set to be estimated using selected experimental data for specified output variables. User options would be varied and would include the parameter estimation method - the cost function definition and the search algorithm. A good range of available search algorithms is important to ensure rapid convergence for nonlinear

models. It should also be possible to estimate parameters over selected segments of the system time response like for example, a particular part of a transient response and to weight individual segments of the time response. Parameter estimates should be presented with Cramer-Rao bounds where appropriate and it should be possible to manually weight selected outputs and parameters for the parameter estimation process.

7.2.4 Sensitivity Coefficients

Sensitivity coefficients are necessary for parameter estimation as well as being very useful for model validation in their own right. Exact sensitivity coefficients can be generated from cosystems of sensitivity equations which run in parallel with the system equations. Such sensitivity equations are best derived by differentiating the system equations manually (chapter 3, section 3.2), but it is probable that computer algebra packages will eventually develop to the stage that this process could be automated. It is also possible to calculate sensitivity coefficients using finite difference methods and this should be an option.

The sensitivity coefficient routines need to be fully integrated with the simulation and parameter estimation routines because the cosystem runs in parallel with the real system simulation and sensitivity coefficients are an integral part of parameter estimation search algorithms.

7.2.5 Animated time responses with respect to parameter variations

Sensitivity coefficients can be a very quick way of analysing the effects of small parameter changes on the model output. By summing the sensitivity coefficient with the model output, an approximation to the output of the modified model can be calculated almost instantaneously. It is in effect a first order linearisation of the model where,

$$f(x+\delta) \approx f(x) + \delta \frac{\partial f}{\partial x}.$$

This facility could be used to give an animated plot of the model output over experimental data while parameters were varied continuously by, for example, moving a slide on a bar with the mouse. As the parameters get further from the operating point at which the sensitivity coefficients and the time response were calculated, the

modified response is going to be less accurate. This depends of course on the degree of nonlinearity of the model. It should however, be possible to simulate the nonlinear system and generate sensitivity coefficients repeatedly while the animated parametric investigation is taking place. This means that the operating point and therefore the time response and the sensitivity coefficients are periodically being updated maintaining accuracy. The deviation of the current parameter values from those at which the sensitivity coefficients and the time response were calculated could be displayed alongside the modified time responses. This repeated updating of the system would mean that the user would only have to leave the parameters constant for as long as it took to run one nonlinear simulation to get an accurate time response for the nonlinear model at that parameter set. The length of this time period would of course depend on the computer hardware as well as the parameters for the simulation procedure. This facility would be extremely useful in the model development stages and would give a 'feel' for the model to supplement knowledge about the system being modelled. A possible screen display of this routine is given figure 43.

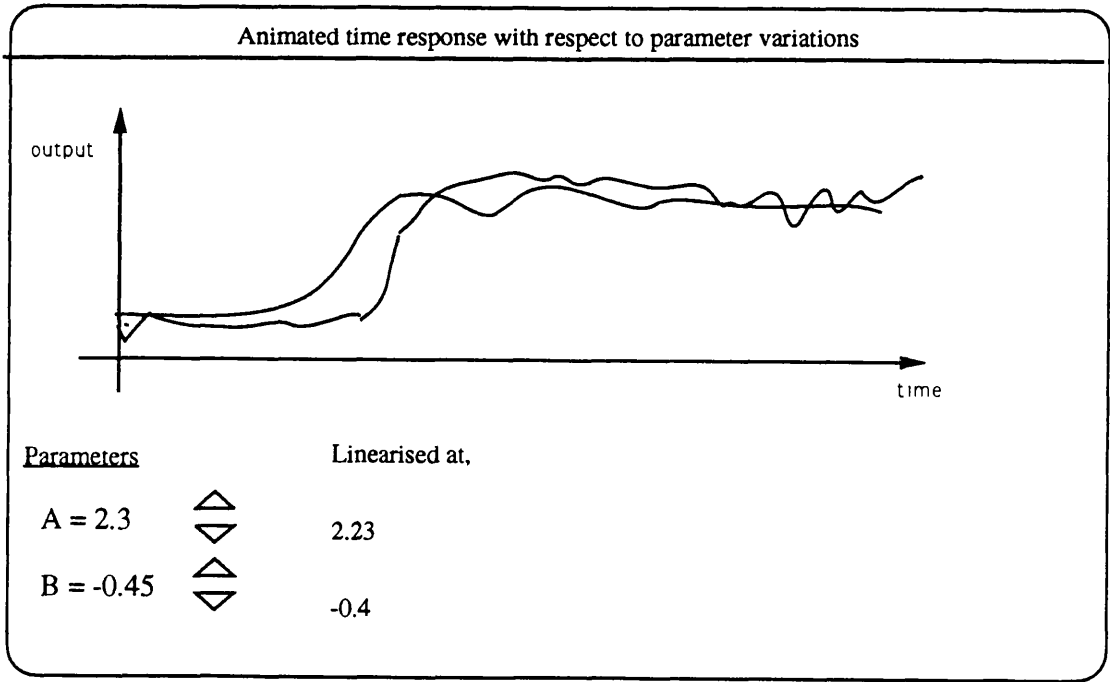


Figure 43 Example of window showing animated time response algorithm output with system time response over experimental data and parameter values

7.2.6 Parameter Selection

Parameter selection is an important part of parameter estimation and it should be possible to display single value sensitivity coefficients (chapter 3, section 3.3 and chapter 6, section 6.7) to indicate the effects of the parameters on system outputs. Equation error estimation could also be incorporated into the parameter estimation routines as another parameter selection technique.

7.2.7 Model Structure Investigation

It should be possible to have more than one model structure in the system at any one time so that the different model structures can be simulated together and the results compared. The full range of model analysis methods should be available for application to all the model structures so that these structures can be compared. It should also be possible to produce residual plots for model structure investigation.

7.2.8 Input and Output

All the results should be clearly displayed on the screen and it should be possible to save them to disk, send them to a printer or copy them to other applications such as mathematical analysis packages or control system design programs. An example of what the output from this program would look like is given in figure 44.

7.2.9 Other validation methods

This program should be as comprehensive as possible in that it should contain a wide range of validation methods. One useful method for large excursion manoeuvres of complicated nonlinear systems such as helicopters is inverse simulation (chapter 2, section 2.6). Inverse simulation means taking a recorded flight path from a flight experiment as the input to an inverse model. The output of the simulation is the control time histories which can then be compared with the recorded control time histories. This method could run in its own window allowing comparison of the results of one validation method with another as well as applying different methods together.

Other possible validation techniques include open loop simulation (chapter 2, section 2.7) which could be integrated with other techniques. Open loop simulation allows a subsystem of the main system to be modelled as an independent system by using

experimental records of system state variables as inputs to the model rather than calculating them in a model of the complete system. Open loop simulation means that a subsystem model can be simulated and validated independently using other techniques. It is a method which works together with other methods and is therefore ideal for an integrated package such as the one specified in this chapter.

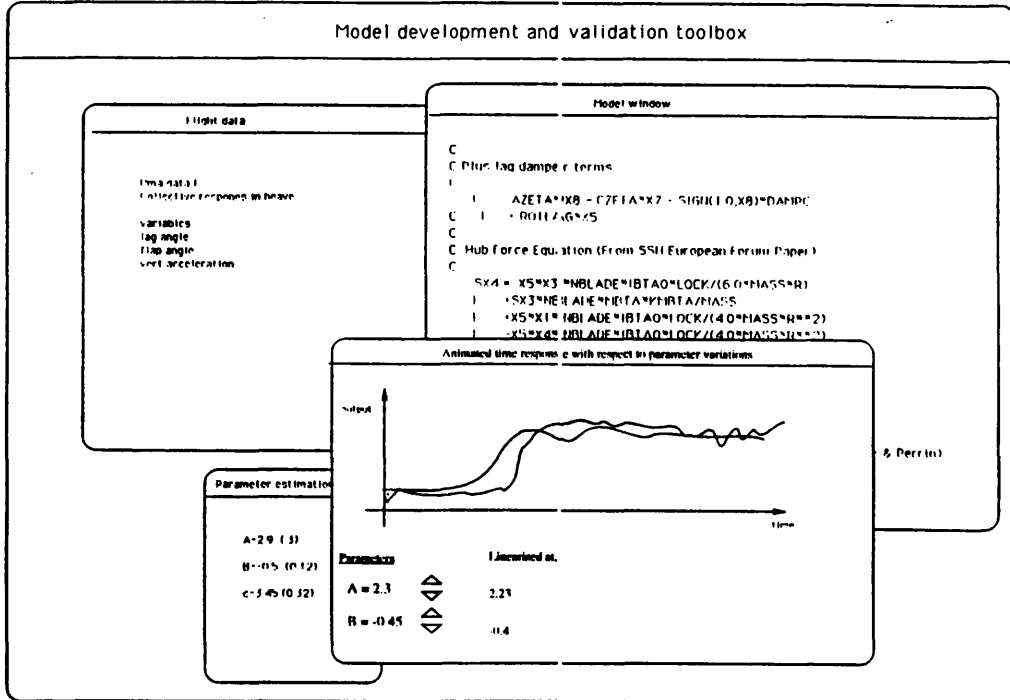


Figure 44 Possible appearance of validation program output showing windows based output with various applications running

7.3 Implementation of Validation Program

One result of this research is that validation methods involving simulation of nonlinear models is slow. Simulation times will have to be considerably reduced for a program this to be a viable proposition. One way of speeding up program execution is to use a parallel computer. This is where the computer consists of many microprocessors running in parallel and communicating with each other and a master controller processor.

The INMOS Transputer¹ is a suitable processor for this type of application. It is designed to be used in parallel and is very fast. The transputers are supplied as

individual units with on board memory and can be connected together in a network structure with a control hierarchy which depends on the application. They can be interfaced via a master controller to a variety of desk top computers to provide input and output facilities like a keyboard, mouse and high resolution colour display.

One problem with implementing software on a transputer network is splitting the program up into sections that can run efficiently in parallel. Communication between transputers is comparatively slow and should be kept to a minimum meaning that processes running on individual transputers should be as independent as possible.

Despite these problems, this model development and validation program is one which leads itself to parallel implementation very well. The most computationally intensive part of the program is simulation of the model itself and sensitivity cosystems. Each sensitivity cosystem requires the results from the system simulation at each integration time step but they are completely independent of each other. An effective parallel implementation of this would be to have the main system simulation and one cosystem running on each transputer as cosystems. This would mean that simulation times would not be dependent on the number of cosystems.

Another common task for this program will be simulation of the same basic system with various sets of parameter values generating a linear search or with alternative model structures. This would be a simple task to implement of a transputer network because the simulations are independent of each other and each simulation could run on its own transputer communicating the results back to the host computer.

The upgrading of the time response and sensitivity coefficients for the animated parameter adjustment facility described in section 7.2.5 is an excellent candidate for parallel processing. The animated parameter adjustment using sensitivity coefficients and the system simulation to update the values are completely independent and can therefore run completely in parallel with communication of the time responses and the sensitivity coefficients from the simulation process to the parameter animation process at the end of each simulation.

A complicated screen display can take a lot of computer power to drive it but it is possible to drive the screen display from a transputer graphics board. This means that graphics processing does not take up valuable processing time - it is done in parallel to the operation of the main validation program.

There are several programming languages available for transputer networks. Some are transputer specific such as OCCAM² and some are parallel versions of conventional languages such as C, PASCAL or FORTRAN. The language used would depend on the availability of ready-written routines and the experience and programming skills of the programmer. Proprietary software packages or other high level languages can only be used on transputers if there has been a version written specifically for transputer implementation or if the source code is available and can be transferred to the transputer. This could cause problems and increase programming time. One possible solutions is to run the package on the host computer while interfacing with the transputers.

An example of a transputer network is shown in figure 45 for the case where different versions of the same system are being simulated in parallel.

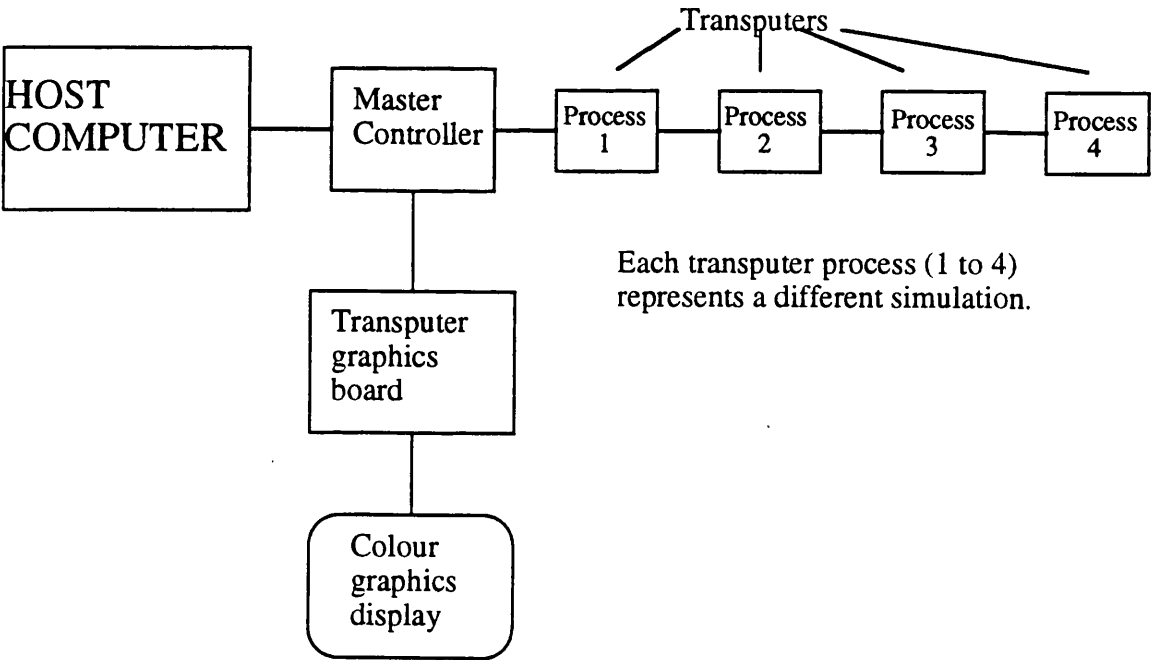


Figure 45 Transputer network for simulating four systems in parallel

7.4 Applications

A package such as this would be most useful in the model development stage. Its ease of use and flexibility would allow the user to test a wide range of possible models improvements both structural and parametric and to develop criteria for validation of these improvements.

With good physical knowledge of the system and familiarity with the software, the engineer could interact with the model, experimenting with a variety of alterations to the model using the tools to develop a better, fully validated model. Such a package and the inherent development and validation techniques could of course be used with a model of any continuous time dynamic system and is by no means restricted to helicopter applications.

References

- 1 *'The transputer databook'*, INMOS, 2nd edition, 1989.
- 2 Jones, G. and Goldsmith, M., *'Programming in Occam 2'*, Prentice Hall International (UK) Ltd., 1988.

Chapter 8

Conclusions and Suggestions for Further Work

8.1 Conclusions

The general conclusion of this research is that model development and validation should be regarded as an interactive process where the user has available a range of validation techniques and selects the most appropriate for the task. The selection criterion should be based on available experimental data, prior knowledge of uncertainties in the model, the type of nonlinearities in the model and importantly, physical knowledge of the system being modelled.

Various validation techniques have been investigated in this thesis.

8.1.1 Validation methods

The model distortion method described in chapter 4 was designed for the nuclear power industry and published applications of the technique have been from that industry. Experiments with simple systems and simulated data showed how the model distortion method works and how it might be applied. The method does not however work in cases where the experimental data is noisy and is therefore of little use for helicopter models where the flight data contains a lot of high frequency noise due to the high vibration environment. It would be a worthwhile exercise to evaluate the technique with a simpler nonlinear system if relatively noise-free measurement data was available.

Analogue matching of model time responses to experimental data is a common validation technique. In this thesis, it has been extended to include several output plots for different parameter values superimposed on each other and on the flight data as a parameter estimation technique with a linear search. Analogue matching was found to be particularly useful in the latter stages of model validation especially where the initial stages of the development process were based on experimental data. The model output was compared with experimental data using previously untouched data from a different experiment.

Sensitivity coefficients proved very important for model validation. Single value sensitivity coefficients give an indication of the effect of particular model parameters on specific outputs and are useful for selecting the output variables and parameter sets for parametric validation techniques. Single value sensitivity coefficients can give considerable insight to the model and represent a powerful validation tool.

Time histories of sensitivity coefficients can be produced by solving a sensitivity cosystem. This is a dynamic system made up of differential equations which are the partial derivatives of the main system equations with respect to some model parameter. The cosystem is simulated alongside the main system to produce time histories of the sensitivity coefficients. These sensitivity coefficients are an essential part of the parameter estimation process providing the gradient information for the cost function minimisation. Examination of the sensitivity coefficient time responses proved a very useful validation tool in this research. It provides information as to the effect of the parameter on a particular model output over the length of an experiment, ie. the effect on trimmed condition, different parts of the transient response and the steady state response. This can provide clues to model development and is especially useful when used in conjunction with other parametric techniques such as parameter estimation.

Maximum Likelihood parameter estimation produces estimates for model parameters free from bias errors due to measurement noise. Parameter estimation was successfully applied to the nonlinear rotor model (chapter 6). The estimates were compared with theoretical values for physical models and with estimates of the same parameters from other models or the same model and a different experimental data set. In all cases, consistency of parameter values implies model validity. Analytically derived sensitivity cosystems (chapter 3) were used to produce the sensitivity coefficients for the parameter estimation search algorithm. This allowed estimation of discontinuous nonlinear elements to be estimated accurately.

All of the above validation techniques were applied to helicopter mechanics and application of all of the validation techniques was greatly enhanced by physical knowledge of the system being modelled. Physical knowledge is necessary for interpretation of the results of all the validation methods and should be considered essential when validating physical models.

8.1.2 Application of model validation methods

The combination of physical knowledge and analogue matching in chapter 5 led to the development of an improved yaw model for the real time helicopter model. It was concluded that in the hover, pitch-flap coupling should be included in the representation of tail rotor dynamics. the main rotor downwash affects the yaw model by increasing the thrust generated by the tail rotor.

A main rotor model was investigated in chapter 6 with respect to the lag damper and the engine and controller representation. Single value sensitivity coefficients were used for output variable selection and a combination of parameter estimation, analogue matching, and sensitivity coefficient time response examination was used for the development and validation stages. Model structures were verified for the lag damper and engine controller but more flight data are necessary to produce validated parameter values.

Single value sensitivity coefficients were also used to select the parameter set for the model distortion technique (chapter 3, section 3.3.1).

8.1.3 Summary

This thesis has shown that nonlinear mathematical models can be validated using nonlinear validation techniques. Various nonlinear validation techniques have been investigated with respect to physical models of helicopter mechanics. These techniques have been evaluated and an integrated approach to model validation and development has been established. This integrated approach is presented in chapter 7 as a specification for a hypothetical computer program for model development and validation.

Such a program would allow the user to apply selected validation methods to a mathematical model and experimental data and essentially to interact with the model and the validation process to produce a better and more accurate representation of the system and improved model validity.

8.2 Suggestions for further work

The mathematical helicopter models which have been used in this thesis are models of subsystems of the helicopter rather than the complete system. Specifically, they represent the yaw mode and the main rotor in isolation as independent systems rather than as part of a model of the complete aircraft. This greatly facilitates the modelling process because the model is much simpler, but in a complex system such as a helicopter with a lot of cross-coupling between different modes, great care must be taken to ensure that unmodelled dynamics from the rest of the aircraft are not affecting the subsystem. In the case of the yaw model and the main rotor model, this was done by experiment design whereby the pilot maintained all aircraft states in a trimmed state except that being excited in the experiment. This is a useful technique but it assumes that the pilot is a perfect controller in that he can remove all the cross-coupling effects.

One possible solution is to measure all the aircraft states and use that data as input to the subsystem. This is known as open loop simulation and is defined in chapter 2, section 2.7. Open loop simulation effectively isolates the subsystem as a separate dynamic system. Further research into this technique is needed to provide a theoretical basis for open loop simulation, to investigate the effects of noise and bias error in the flight data on the subsystem output and to determine the effect of the degree of cross-coupling present in the model. Open loop simulation has the potential to be a powerful validation tool. The ability to isolate individual sections of the model means that these sections could be developed and validated separately as individual systems. This modular approach would substantially improve the validation process. A possible example of an application area for open loop simulation is the helicopter yaw model developed in chapter 5 of this thesis. The excitement of other modes in the aircraft was identified as a source of difficulty in validating the model. Open loop simulation would remove the effects of these other modes and allow the yaw model to be developed and validated as a subsystem.

The structure of the lag damper and the engine controller representation in the main rotor model was validated using parameter estimation (chapter 6), but the values for the parameters were not conclusive because of insufficient flight data. Further experiments would allow values for the parameters to be derived and validated. As stated in chapter 6 section 6.10, a worthwhile next stage in the development of the rotor model is inclusion of rotor blade twist. This would be more representative of the

real system and should improve the model with respect to rotor dynamics without causing any significant increase in computation time during simulation.

The model validation computer package described in chapter 7 is realisable. If implemented on a transputer network as suggested it would be fast enough and it would be a powerful aid to model development. Such a program would allow many validation tools to be used interactively and would substantially improve the model validation process.

Appendix 1

Model Distortion Method

Program Listings

The following routines were used to evaluate the model distortion technique (chapter 4). The example given here is for a linear helicopter model and two parameters. The programs were run on a Digital VAXstation 2000 using MATLAB from batch jobs. MATLAB is a mathematical analysis package based on matrices. The programs took several hours to run.

Program makedataheli

This program is written in MATLAB. It generates the simulated data used for the model distortion technique in chapter 4. A linear eight state representation of a helicopter is simulated and the output differentiated with respect to time. There is an option to add white noise to the signal.

Program listing

```
% Generates time history data for an eighth order system used as an
example for
% the butterworth technique.
load th0;                %get system input vector
load pumahover           %get system equations
h=7.8125e-3;            %set up integration
t=(0:h:10)';
zer=zeros(t);
in=zer;
in(1:100)=ones(1,100);
in(101:200)=-1*ones(1,100);
u=[th0(1:1281) zer zer zer]; %system input
%load hover;
[y,x]=lsim(A,B,C,D,u,t); %simulate system
w=x(:,2);                %select vert velocity (w) from output
wdot=diffate(w,h);       %differentiate vertical velocity
rand('normal');
%r=rand(w)*.1;           % generate additive white noise
t=0;
w=w+r;
wopt=w;
input=u;
```

Variables

th0	main rotor collective (system input)
h	integration interval
t	time vector for simulation
zer	column of zeros

in	input
u	input vector
A,B,C,D	system matrices
y	simulation output
x	simulation state time histories
w	vertical velocity
wdot	vertical acceleration
r	white noise vector
wopt	simulated data with added noise
input	system input used to generate simulated data

The differentiation routine is done by a small MATLAB routine using a second order algorithm.

Function diffate

This function is written in MATLAB and it differentiates the input x , using a numerical second order algorithm where x is a time history with step size h .

Program listing

```
function y=diffate(x,h);
%differentiates function x using second order method. Error is  $\sim h^2$ 
[a b]=size(x);
switch=0;
if a<b,
    switch=1;
    x=x';
end
y=x;
[a b]=size(x);
n=a;
y(1,:)=(-3*x(1,:)+4*x(2,:)-x(3,:))/(2*h);
for k=2:n-1,
    y(k,:)=(x(k+1,:)-x(k-1,:))/(2*h);
end
y(n,:)=(x(n-2,:)-4*x(n-1,:)+3*x(n,:))/(2*h);
if switch==1,
    y=y';
end
```

Variables

y	function output (derivative)
x	function input
h	step size
a	number of rows in input vector
b	number of columns in input vector
switch	flag to make input have more rows than columns so that input is b time histories with a points in each
n	number of time points
k	loop variable

Program optimise

This program, written in MATLAB, executes the model distortion algorithm on a linear helicopter model using simulated data generated in `makedataheli` above. The technique of Lagrange multipliers is used (see chapter 5). It is implemented using the MATLAB `fsolve` command¹. The parameter values at each time point are stored in an array which can be plotted or processed to produce a mean and variance.

Program listing

```
% Does Lagrange optimisation for model distortion technique
lamda=1;
zw0=-0.28;           %initial parameter values
zt0=-0.35;
h=7.8125e-3;         %integration interval
paras=[];            %initialise variables
details=zeros(16,1);
details(1)=2;
p2=[zw0;zt0;lamda];  %initialise parameter values
data=ones(1,42);
for t=h*2:h:19.9,    %main loop
    u=input(t/h+1,:); %assign variables to pass to system
    equations
    data(1)=w(t/h+1);
    data(2)=wdot(t/h+1);
    data(3:10)=x(t/h+1,:);
    data(11:14)=u;
    data(15)=zw0;
    data(16)=zt0;
    data(17:24)=A(2,:);
    data(25:28)=B(2,:);
    data(29:36)=A(5,:);
    data(37:40)=B(5,:);
    %data(41)=w(t/h+1);
    %data(42)=wdot(t/h+1);
    [p2,tcode]=fsolve('systemheli',p2,details,data); %solve lagrange
    equations
    zw=p2(1);
```

```

zt=p2(2);
pp=[zw;zt;w(t/h+1)];           %save results in array
paras=[paras pp];
term=[term; tcode];
if fix(t*10)==t*10,             %backup results to disk
    save updatehovsimnoise paras t term
end
end
save updatehovsimnoise paras t term

```

Variables

```

lamda    Lagrange parameter
zw0,zt0  initial parameter values
h        integration interval
paras    cumulative save vector for parameter values
details  control vector for fsolve algorithm
p2       parameter vector
data     variable to be passed to system model
t        time vector
u        system input
A,B      system matrices
w        vertical velocity simulated data
wdot     vertical acceleration
tcode    solve algorithm termination code
term     cumulative save vector for tcode

```


Function systemheli

This is a MATLAB function, called by optimise, which evaluates the differential equations making up the model. The model in this case is an eighth order linear helicopter model. It is possible to weight individual parameters in this function.

Program Listing

```
% Second system
function dG=systemheli(p,data);
w=data(1);
wdot=data(2);
x=data(3:10);
u=data(11:14);
zw=p(1);
zt=p(2);
lamda=p(3);
zw0=data(15);
zt0=data(16);
A2=data(17:24);
B2=data(25:28);
A5=data(29:36);
B5=data(37:40);
%v=data(41);
%vdot=data(42);
x(2)=w;
%x(5)=v;
A2(4)=zt;
A2(2)=zw;
%A5(1)=yu;
zwweight=1/1;
ztweight=1/1;
dG=[2*zwweight*(zw-zw0)+lamda*x(2);
2*ztweight*(zt-zt0)+lamda*x(4);
wdot-(sum(A2.*x)+sum(B2.*u))];
```

Variables

dG function output

p parameter vector
data data used by function
w vertical velocity simulated data
wdot vertical acceleration
x state values
u system input
A,B system matrices
w vertical velocity simulated data
t time vector
zw,zt,lamda parameters
zw0,zt0 initial values of parameters
A2,A5 relevant parts of system A matrix
B2,B5 relevant parts of system B matrix
zwweight weight for parameter zw
ztweight weight for parameter xw

Appendix 2

Yaw Model Development

Program Listings

The following programs simulate the yaw motion of an Aerospatiale PUMA helicopter. The programs are written in a FORTRAN based simulation language, TSIM² and were used to development and validate the model as described in chapter 5. The TSIM code is compiled and run in a TSIM operating environment.

The programs were run on a Digital MicroVAX II.

Programs YAW.SIM and TAILROTOR.SIM

The following TSIM routines define the yaw model. The system equations are in a separate include file, TAILROTOR.SIM.

Program listing YAW.SIM

```

C      This program simulates the yaw motion of a helicopter with tail
C      rotor dynamics.
C
C      Written September 1991
C
C
C      INCLUDE 'EXTERNALSVARS2.INC'
C
C
C      Define states of system
C
C      RA - aircraft yaw rate
C      YAW - aircraft heading angle
C      TH0T - tail rotor collective after delta3 adjustment
C      TH00T - tail rotor collective before delta3 adjustment
C      X0T - tail rotor jack
C      YT - side force generated by tail rotor

```

```

C      VELZT - wind velocity normal to plane of tail rotor
C
C      STATE RA,YAW,HEAD
C      NODE VELZT,YT,TH00T,X0T,JACK,BIAS,VELT
C
C
C      Define system parameters
C
C      -- helicopter configuration
C
C      VARIABLE FINBLK,FINARE,RT,UA,WA,LENGTH,INFCON,INFLOW
C      VARIABLE Q,XCG,OMEGAT,VA,HEIGHT,P,DELTA3,OLDCTT,TH0T
C      VARIABLE NLT,A0T,SOLT,INRTIA,RHO,FAILFG,CTT,INFLOT
C      VARIABLE VLIMHI,VLIMLO,FAILVR,OLDINF,A0,A1,A2,A3
C      VARIABLE NORM,HEADIN,ROLL,RPM,RRATE,TORQUE,YRATE
C      VARIABLE RTML,DLAMTR,UMTSQ,FYRT,WT,INFLO2,THRUST
C      VARIABLE FNFLOT,FTH0T,FFILFG,FCTT,FRTML,FLAMTR,FUMTSQ
C      VARIABLE WINDAR,WINDB,WINDSC,R,WMIN,WMAX,WTOT,BETA,CHIO
C      VARIABLE DWASH,WIND,GAMMA,LL,PP,THETA
C      INTEGER COUNT
C
C      Initialisation
C
C      IF (INIT.NE.0) THEN
C          FINBLK=1.0-3.0/4.0*FINARE/(3.1415*RT**2)
C
C          VELT=SQRT(UA**2+(WA-INFCON*INFLOW+Q*(LENGTH+XCG))**2)
C          1      / (OMEGAT*RT)
C      ENDIF
C
C
C      INCLUDE 'TAILROTOR.SIM'
C      SIMSTOP
C
C      END

```

Program listing for TAILROTOR.SIM

```

C      FIRST ORDER TAIL ROTOR MODEL
C
C      FROM G. PADFIELD HELISTAB REPORT
C
C      WRITTEN AUGUST 1991
C
C
C      INPUT ACTUATOR
C
C      STH00T=-0.30*JACK+BIAS
C
C
C      EFFECT OF WIND COMING FROM ONE DIRECTION AND BLOWING MAIN ROTOR
C      DOWNWASH OVER TAIL ROTOR
C
C      BETA=-INFLOW/WIND/COSD(HEAD-CHI0)
C
C      GAMMA=ATAN(-BETA)-ATAN(1.013/1.646)
C      IF (GAMMA.LT.0.0) THEN
C          GAMMA=GAMMA+3.1415
C      ENDIF
C      LL=SQRT(1.646**2+1.013**2)
C      PP=LL*SIN(GAMMA)
C      IF (PP.GE.RT) THEN
C          WINDAR=0.0
C          GOTO 123
C      ENDIF
C      IF (PP.LE.-RT) THEN
C          WINDAR=3.1415*RT**2
C          GOTO 123
C      ENDIF
C      THETA=2*ACOS(PP/RT)
C      WINDAR=THETA/2*RT**2-0.5*PP*RT*SIN(THETA/2.0)
CC
C      IF (WINDAR.LT.0.0) THEN

```

```

        WINDAR=0.0
    ENDIF
123  CONTINUE
CC
    INFCON=WINDAR*WINDSC
C
    SVELZT= (-VA+ (LENGTH+XCG) *RA-HEIGHT*P)
1      / (OMEGAT*RT)
C
    SVELT=SQRT (UA**2+ (WA-INFCON*INFLOW+Q* (LENGTH+XCG) ) **2)
1      / (OMEGAT*RT)
C
C    ITERATE FOLLOWING TWO EQUATIONS FOR TAIL ROTOR INFLOW AND
THRUST
C    COEFFICIENT USING NEWTON METHOD
C
    DO 100 COUNT=1,10
C
    RTML=-VELZT-INFLOT
    UMTSQ=VELT**2+RTML**2
C
C    pitch-flap coupling
    TH0T=(TH00T+DELTA3*NLT*4.0/3.0*RTML)
1      / (1.0-DELTA3*NLT*(1+VELT**2))
C
    CTT=(A0T*SOLT/2.0) * (TH0T/3.0*(1.0+3.0/2.0*VELT**2)
1      +RTML/2.0)
C
    TEMPTR=2.0*(UMTSQ**1.5)-RTML*CTT+0.25*A0T*SOLT*UMTSQ
C
    DLAMTR=-(2.0*INFLOT*SQRT(UMTSQ)-CTT)*UMTSQ/TEMPTR
C
    INFLOT=INFLOT+DLAMTR
100  CONTINUE
C
    IF (INFLOT.NE.0.0) THEN
        FAILFG=DLAMTR/INFLOT

```

```

ELSE
    FAILFG=1
ENDIF

C
    THRUST=2.0*RHO*3.1415*RT**2*(VELZT+INFLOT)*INFLOT
C
C    FORCE EQUATION
C
    SYT=RHO*(OMEGAT*RT)**2*SOLT*A0T*(3.1415*RT)**2
1    *(CTT/A0T/SOLT)*FINBLK
C
C    DYNAMIC EQUATION
C
    SHEAD=YRATE*180/3.1415
    SYAW=-RA*180/3.1415
C
    SRA=(LENGTH*YT-TORQUE)/INRTIA

```

Variables

TH00T Tail rotor collective pitch
 JACK tail rotor jack position
 BIAS offset for tail rotor collective (calculated in model trim algorithm)
 BETA Angle of main rotor downwash
 INFLOW Main rotor inflow
 CHI0 wind direction
 HEAD aircraft heading
 WIND windspeed
 GAMMA angle of downwash in xz plane
 LL distance from centre of tail rotor disc to edge of main rotor disc
 RT radius of tail rotor
 PP, THETA see figure 22
 WINDAR area of tail rotor disc inside main rotor downwash
 INFCON multiplicative factor representing effect of main rotor downwash on tail rotor thrust
 VELZT perpendicular inflow to tail rotor
 LENGTH distance from tail rotor to aircraft reference point

XCG	distance from aircraft reference point to centre of gravity
VA	aircraft sideward velocity
RA	aircraft yaw rate
HEIGHT	height of tailrotor above centre of gravity
P	aircraft roll rate
OMEGAT	rate of rotation of tail rotor
VELT	tangential inflow to tail rotor
UA	aircraft forward velocity
WA	aircraft vertical velocity
INFLOW	tail rotor inflow
Q	aircraft pitch rate
COUNT	loop variable
RTML	perpendicular air velocity at tail rotor
INFLOT	tail rotor inflow
UMTSQ	square of total airflow at tail rotor
THOT	tail rotor collective pitch
DELTA3	pitch/flpa coupling
NLT	blade flapping ratio
CTT	thrust coefficient at tail rotor
A0T	tail rotor lift curve slope
SOLT	tail rotor solidity ratio
TEMPTR	intermediate variable
DLAMTR	derivative of inflow used in Newton optimisation
FAILFG	flag to indicate if convergence in inflow calculation was successful
THRUST	tail rotor thrust
RHO	air density
YT	force generated by tail rotor
FINBLK	blockage factor caused by tail fin
TORQUE	main rotor torque in hover
INRTIA	moment of inertial of helicopter about z-axis
FINARE	area of tail fin

Notes

States are defined as dynamic variables in the system which are integrated. Nodes are defined as input/output variables or summing junction variables. Equations are stated $S<variable>=<expression>$.

If the variable is a state this is a differential equation and the S represents the differential operator. If the variable is a node, it is evaluated as an algebraic expression.

Appendix 3

Main Rotor Model Development

Program Listings

The following programs simulate the main rotor of a helicopter. The model is in the form of eight nonlinear differential equations. The program is written in TSIM², a FORTRAN based simulation language. It is used in chapter 6 of this thesis.

The main program is ROTOR.SIM. This defines the variables and routines used in the simulation. There are several include files within ROTOR.SIM. These include HOVMAINOLD.SIM, the rotor equations, HOVCOSAZ.SIM, the cosystem equations for parameter A_{ξ} , ESTIMATE.SIM, a program used for parameter estimation and EXTERNALSMULTI.INC which is a collection of routines for parameter estimation. These routines are written in FORTRAN and are called from the TSIM operating environment from which simulations are run and results analysed.

Program listing for ROTOR.SIM

```

C      This program simulates the main rotor of a helicopter with flap
C      and lag dynamics. There are many simulations, the main one and
C      cosystems for sensitivity coefficient calculation.
C      A least squares method is then used to estimate this
C      parameter using flight data. There are also routines to
C      calculate the
C      sensitivity coefficients by a finite difference method for
C      checking
C      the cosystem during its development and for comparisons of the
C      two
C      methods.
C      This version is modified to include the system equations in the
C      correct
C      TSIM format.
C
C      Written May 1990
C      Updated July 1990

```

C Updated September 1990

C

INCLUDE 'EXTERNALSVARS2.INC'

C

C

C Define states of system

C X1 - Induced velocity/ Inflow (normlised)

C X2 - Beta0, Multiblade Flapping angle

C X3 - Beta0dot. Multiblade flapping velocity

C X4 - Vertical Velocity (of hub and aircraft)

C X5 - Rotorspeed

C X6 - Engine Torque

C X7 - Zeta0, Multiblade Lag Angle

C X8 - Zeta0dot, Multiblade lag velocity

C

STATE X3,X4,X7,X8

STATE X1,X2,X5,X6

STATE CCX1,CCX2,CCX3,CCX4,CCX5,CCX6,CCX7,CCX8

STATE EEX1,EEX2,EEX3,EEX4,EEX5,EEX6,EEX7,EEX8

STATE DDX1,DDX2,DDX3,DDX4,DDX5,DDX6,DDX7,DDX8

NODE TH0

VARIABLE CD,UPUT,DUPUT,DCDDP,LIFTDG,LIFTER,KENG

VARIABLE ANS1,ANS2,ANS3,ANS4,ANS5

VARIABLE M11,PITTL,CT,FAILFG,HHH,INFLOW,FI,VELZ,FFKENG

VARIABLE INERT,QLAG,FLAG,FDAMP,FDENG,FEENG,FINDD,FINDE,FINDF

VARIABLE CORR1,CORR2,CORR3,CORR4,CORR5,CORR6,CORR7,CORR8

VARIABLE

CCCOR1,CCCOR2,CCCOR3,CCCOR4,CCCOR5,CCCOR6,CCCOR7,CCCOR8

VARIABLE

DDCOR1,DDCOR2,DDCOR3,DDCOR4,DDCOR5,DDCOR6,DDCOR7,DDCOR8

VARIABLE

EECOR1,EECOR2,EECOR3,EECOR4,EECOR5,EECOR6,EECOR7,EECOR8

VARIABLE

FFCOR1,FFCOR2,FFCOR3,FFCOR4,FFCOR5,FFCOR6,FFCOR7,FFCOR8

INTEGER COUNT

VARIABLE DL1,DL2,TQWGT,RPMWGT,TH0TW,CR1,CR2,CR3,NUMSTP

C

REAL*4 DELAY1,DELAY2

```

c      COMMON /UGDCMN/ DELAY1(0:100,2),DELAY2(0:100,2)
C
C Define system parameters
C
C -- helicopter configuration
C
      VARIABLE RESID,X7X8,ROTLAG,DISK,DET,FDT,DTRM,DETA,ROTDAT
      VARIABLE CONDAT,LAGDAT,TQDAT,NORDAT,RPMDAT,CONRAD,GRAV,HH
      VARIABLE A,C,NBLADE,OMEGA,R,RKMOM,SOL,LOCK,DELTA,C1,C2
      VARIABLE MBTA,MASS,ENGGAI,ENGTCO,RHO,IBTA0,IR,TEMP4,TEMP5
      VARIABLE C0,DEL0,TAUAM,HOFF,KMBTA,AZETA,BZETA,CZETA,DAMPC
      VARIABLE SFP,CDF,LAG,OPTBEG,OPTEND,ENGFF
      VARIABLE TAU01,TAU30,TAU31,TAU20,TAU21,LAG0,DAT1,DAT2
C
      VARIABLE F3,F4,F5,F6,F8
C
C External definitions.
C
      EXTERNAL 1 4100
      EXTERNAL 2 4200
      EXTERNAL 3 4300
C      EXTERNAL 4 4400
      EXTERNAL 6 4600
      EXTERNAL 7 4700
      EXTERNAL 8 4800
      EXTERNAL 11 5100
      EXTERNAL 12 5200
      EXTERNAL 13 5300
      EXTERNAL 15 5500
      EXTERNAL 17 5700      .
      EXTERNAL 18 5800
      EXTERNAL 19 5900
      EXTERNAL 20 6000
      EXTERNAL 21 6100
      EXTERNAL 22 6200
      EXTERNAL 23 6300
      EXTERNAL 24 6400

```

```
EXTERNAL 25 6500
external 26 6600
c    EXTERNAL 27 6700
C    EXTERNAL 30 4350
C
    WAIT
C
C Initialisation
C
C
    IF (INIT.EQ.1) THEN
        LOCK = RHO*A*C*R**4/IBTA0
        SOL = NBLADE*C*R/(3.1415927*R**2)
C    HHOLFA = 0.0
C    FCHHOLD = 0.0
C    FDHHOLD = 0.0
C    FDX1=X1
C    FDX2=X2
C    FDX3=X3
C    FDX4=X4
C    FDX5=X5
C    FDX6=X6
C    FDX7=X7
C    FDX8=X8
C    FEX1=X1
C    FEX2=X2
C    FEX3=X3
C    FEX4=X4
C    FEX5=X5
C    FEX6=X6
C    FEX7=X7
C    FEX8=X8
C    FFX1=X1
C    FFX2=X2
C    FFX3=X3
C    FFX4=X4
C    FFX5=X5
```

```

C      FFX6=X6
C      FFX7=X7
C      FFX8=X8
      ENDIF
C      OPTIONS TO INCLUDE FILES FOR DIFFERENT PARAMETER ESTIMATION
PROBLEMS
      INCLUDE 'HOVMAINOLDMOM.SIM'
C      INCLUDE 'HOVMAINOLDDFDMOM.SIM'
C      INCLUDE 'HOVMAINOLDENGGA1.SIM'
C      INCLUDE 'HOVMAINOLDENGTCO.SIM'
C      INCLUDE 'HOVCOSENGGA1.SIM'
C      INCLUDE 'HOVCOSENGtco.SIM'
      INCLUDE 'HOVCOSoldAZ.SIM'
      INCLUDE 'HOVCOSoldcZ.SIM'
      INCLUDE 'HOVCOSolddc.SIM'
C      INCLUDE 'FINDIFFSENSCOEFF.SIM'
      INCLUDE 'ESTIMATE3.sim'
c      SIMSTOP
C
      INCLUDE 'EXTERNALSMULTI.INC'
C
      END
C      INCLUDE 'DELAY.ICL'

```

Program listing for HOVMAINOLD.SIM

```

C      This simulation simulates the main rotor on a PUMA helicopter
and is
C      part of a parameter estimation program ROTORPARA.SIM.
C
C      Written May 1990
C
C      Inflow equation - version using apparent additional air mass
C
      SX1 = ((3.0/16.0*A*C*NBLADE*(X5)/(3.1415927*R)
1      *(2.0/3.0*(X5)*R*(TH0)
1      -(X1)+(X4)-2.0/3.0*R*X3)

```

```

1      -1.5/R*1.0/(1.0+A*SOL/(16.0*X1/X5/R))
1      *(X1)/RKMOM
1      *((X1)/RKMOM-(X4)
1      +2.0/3.0*R*X3))/0.637)/C0
C
C
C   Coning kinematics
C
      SX2 = X3
C
C   Flapping equation (From SSH European Forum Paper)
C   Modified to include contribution from lag
C
      SX3 = SX4*MBTA/IBTA0
1      +2.0*X5*X8*X2
1      +X2*X7*SX5
1      -X3*(X5-X8)
1      *LOCK/8.0*1.0/(1.0+3.142*SOL/(4.0*X1/X5/R))
1      +X4/(6.0*R)*(X5-X8)
1      *LOCK*1.0/(1.0+3.142*SOL/(4.0*X1/X5/R))
1      -X1/(6.0*R)*(X5-X8)
1      *LOCK*1.0/(1.0+3.142*SOL/(4.0*X1/X5/R))
1      +TH0*(X5-X8)**2
1      *LOCK/8.0*1.0/(1.0+3.142*SOL/(4.0*X1/X5/R))
1      - X5**2*(1.0+HOFF*R*MBTA/IBTA0)*X2
C
C   Flapping Equation (ATMcC - Similar to Blackwell, Feik & Perrin)
C
C      SX3 = -X5**2.0*X2*(1.0+HOFF*R*MBTA/IBTA0)
C      1      +2.0*X5*X8*X2
C      1      +MBTA/IBTA0*SX4
C      1      -SX5*X7*X2*HOFF*R*MBTA/IBTA0
C      1      -X8**2.0*X2
C      1      +(AERODYNAMIC TERMS)
C
C
C   Lagging Equation (ATMcC - Same as Blackwell, Feik & Perrin)

```

C

$$SX7 = X8$$

$$SX8 = 2.0 * X2 * X8 * X3$$

$$\begin{aligned}
 1 \quad & -2.0 * X5 * X3 * X2 \\
 1 \quad & + SX5 * (1.0 + HOFF * R * MBTA / IBTA0) \\
 1 \quad & - X5 ** 2.0 * X7 * HOFF * R * MBTA / IBTA0 \\
 1 \quad & + 1.0 * (LOCK / (6.0 * R) * (X5 - X8) * (X1 - X4) * TH0 \\
 1 \quad & \quad + LOCK / 8.0 * (X5 - X8) * X3 * TH0 \\
 1 \quad & \quad - LOCK / (4.0 * R ** 2) * (X1 - X4) ** 2 \\
 1 \quad & \quad - LOCK / (3.0 * R) * (X1 - X4) * X3 \\
 1 \quad & \quad - LOCK / 8.0 * X3 ** 2 \\
 1 \quad & \quad + LOCK / (8.0 * A) * DEL0 * (X5 - X8) ** 2)
 \end{aligned}$$

C

C Plus lag damper terms

C

$$1 \quad - AZETA * X8 - CZETA * X7 - SIGN(1.0, X8) * DAMPC$$

C

C Hub Force Equation (From SSH European Forum Paper)

C

$$\begin{aligned}
 SX4 = & X5 * X3 * NBLADE * IBTA0 * LOCK / (6.0 * MASS * R) \\
 1 \quad & + SX3 * NBLADE * MBTA * KMBTA / MASS \\
 1 \quad & + X5 * X1 * NBLADE * IBTA0 * LOCK / (4.0 * MASS * R ** 2) \\
 1 \quad & - X5 * X4 * NBLADE * IBTA0 * LOCK / (4.0 * MASS * R ** 2) \\
 1 \quad & - X5 ** 2 * TH0 * NBLADE * IBTA0 * LOCK / (6.0 * MASS * R) + 9.81
 \end{aligned}$$

C

C Plus fuselage, tailplane, wing etc drag

C

$$1 \quad + 0.5 * RHO * (X1 - X4) ** 2 * SFP * CDF / MASS$$

C

C Rotorspeed degree of freedom (ATMcC Same as Blackwell, Feik & Perrin)

C

$$\begin{aligned}
 SX5 = & NBLADE / IR * ((IBTA0 + HOFF * R * MBTA) \\
 1 \quad & \quad * (SX8 + 2.0 * X3 * X2 * (X5 - X8)) \\
 1 \quad & \quad + HOFF * R * MBTA * X7 \\
 1 \quad & \quad * (2.0 * X5 * X8 - X2 * SX3 - X3 ** 2 - X8 ** 2)) \\
 1 \quad & - (4.0 / 3.0 * RHO * 3.1415927 * R ** 3 * SX1
 \end{aligned}$$


```

1      +2.0*RHO*3.1415927*R**2
1      *1.0/(1.0+A*SOL/(16.0*X1/X5/R))
1      *(X1)/RKMOM
1      *((X1)/RKMOM
1      -X4+2.0/3.0*R*X3))
1      *((X1)-X4)
1      /(X5)/IR+X6/IR-TEMP4

C
C Plus profile part
C
1      -RHO*3.1415927*R**4*X5**3*(DEL0*SOL/8.0)/X5/IR
C
C BLADE ELEMENT ROTOR SPEED DYNAMICS
C
C BLADE ELEMENT FUNCTION FOR BLADE TWIST
C
C SX5:=INT(DRAG,R)+INT(LIFT,R);
C
C      LIFTDG=-(C*R*RHO*X5**2*(3.*A*R**3*X3**2+3.*A*R**3*X3*TH0
C      . *X8-3.*A*R**3*X3*TH0*X5-8.*A*R**2*X4*X3-4.*A*R**2*X4
C      . *TH0*X8+4.*A*R**2*X4*TH0*X5-4.*A*R**2*X3**2*HOFF+8.*
C      . A*R**2*X3*X1-4.*A*R**2*X3*HOFF*TH0*X8+4.*A*R**2*X3*
C      . HOFF*TH0*X5+4.*A*R**2*X1*TH0*X8-4.*A*R**2*X1*TH0*X5+
C      . 6.*A*R*X4**2+12.*A*R*X4*X3*HOFF-12.*A*R*X4*X1+6.*A*R*
C      . X4*HOFF*TH0*X8-6.*A*R*X4*HOFF*TH0*X5-12.*A*R*X3*X1*
CC     . HOFF+6.*A*R*X1**2-6.*A*R*X1*HOFF*TH0*X8+6.*A*R*X1*
CC     . HOFF*TH0*X5-12.*A*X4**2*HOFF+24.*A*X4*X1*HOFF-12.*A*
C      . X1**2*HOFF-3.*R**3*X8**2*DEL0+6.*R**3*X8*DEL0*X5-3.*
C      . R**3*DEL0*X5**2))/(24.*(X8**2-2.*X8*X5+X5**2))
C
C      LIFTER=-(A*C*(R/4.)*RHO*X5**2*(3.*(R/4.)***3*X3**2
C      . +3.*(R/4.)***3*X3*
C      . TH0*X8-3.*(R/4.)***3*X3*TH0*X5-8.*(R/4.)***2*X4*X3-4.
C      . *(R/4.)***2*X4*
C      . TH0*X8+4.*(R/4.)***2*X4*TH0*X5-4.*(R/4.)***2*X3**2*HOFF+8.
C      . *(R/4.)***2*
C      . X3*X1-
4.*(R/4.)***2*X3*HOFF*TH0*X8+4.*(R/4.)***2*X3*HOFF*TH0*X5+
C      . 4.*(R/4.)***2*X1*TH0*X8-4.*(R/4.)***2*X1*TH0*X5+6.*(R/4.)*X4**2

```

```

C      . +12.*(R/4.)*
C      . X4*X3*HOFF-12.*(R/4.)*X4*X1+6.*(R/4.)*X4*HOFF*TH0*X8-
6.*(R/4.)
C      . *X4*
C      . HOFF*TH0*X5-12.*(R/4.)*X3*X1*HOFF+6.*(R/4.)*X1**2-6.*(R/4.)
C      . *X1*HOFF
C      . *TH0*X8+6.*(R/4.)*X1*HOFF*TH0*X5-12.*X4**2*HOFF+24.*X4*X1
C      . *HOFF-12.*X1**2*HOFF))/(24.*(X8**2-2.*X8*X5+X5**2))
C
C      SX5=(-(LIFTDG-LIFTER
C      1          ))*NBLADE/IR
C      1          +X6/IR-TEMP5
CC
C      UPUT=(X1+R*X3-X4)/(X5-X8)/R
C      CD=A*(TH0-UPUT)*UPUT+DEL0
C      FDAMP=-AZETA*X8-CZETA*X7-SIGN(1.0,X8)*DAMPC
C      SX5=NBLADE/IR*((IBTA0+HOFF*R*MBTA)
C      1          *(SX8 + 2.0*X3*X2*(X5-X8))
C      1          +HOFF*R*MBTA*X7
C      1          *(2.0*X5*X8 - X2*SX3 - X3**2 - X8**2))
c      aero drag 10/3/92 with twist
C      1      -(0.5*RHO*C*CD*(X5-X8)**2*(R**4/4.0-HOFF*R**3/3.0)
C
CC
C      1      -(0.5*RHO*C*(X5-X8)**2
C      1          *(A*(X1+R*X3-X4)/(X5-X8)/R*((TH0-(X1+R*X3-X4)
C      1          /(X5-X8)/R)*R**4/4.0)
C      1          +DEL0*(R**4/4.0-H*R**3/3.0)
C      1          -A*HOFF*(X1+R*X3-X4)/(X5-X8)/R
C      1          *((TH0-(X1+R*X3-X4)/(X5-X8)/R)*R**3/3.0))
Cc
CC      1      -0.5*RHO*C*(A*(X5-X8)*(TH0*(X1-X4)*R**2/2.0+TH0*X3*
CC      1          R**3/3.0+TH0TW*(X1-X4)*R**3/3.0+TH0TW*X3*R**4/4.0)
CC      1          -A*((X1-X4)**2*R+X3*(X1-X4)*R**2+X3**2*R**3/3.0)
CC      1          +DEL0*(X5-X8)**2*R**3/3.0)*HOFF*R*NBLADE/IR
C
C

```

```

C
C Engine as lag
C
      SX6 = ENGGAI*(X5-OMEGA) + ENGTCO*X6
C
C with anticipator term
C
      1      + ENGFF*SX5
C
C Height integrator
C
      SH = - X4
C
C End of equations defining system

```

Program listing for HOVCOSAZ.SIM

```

C      This program simulates the cosystem of the helicopter rotor and
C      is an
C      INCLUDE file in the parameter estimation program ROTORPARA.SIM.
C      The parameter is AZETA.
C
C      Written May 1990
C      Modified June 1990
C
C      Inflow equation - version using apparent additional air mass
C
      CCTH0=TH0
C
      SCCX1 = ((3.0/16.0*A*C*NBLADE*(CCX5)/(3.1415927*R)
1          *(2.0/3.0*X5*R*CCTH0-(X1)+(X4)-2.0/3.0*R*X3)
1          +3.0/16.0*A*C*NBLADE*(X5)/(3.1415927*R)
1          *(2.0/3.0*CCX5*R*CCTH0-(CCX1)+(CCX4)-2.0/3.0*R*CCX3)
1          +1.5/R*1.0/(1.0+A*SOL/(16.0*X1/X5/R))**2
1          *A*SOL*R/16.0*(X1*CCX5-X5*CCX1)/X1**2
1          *(X1)/RKMOM
1          *((X1)/RKMOM-(X4)

```

```

1      +2.0/3.0*R*X3)
1      -1.5/R*1.0/(1.0+A*SOL/(16.0*X1/X5/R))
1      *(CCX1)/RKMOM
1      *((X1)/RKMOM-(X4)
1      +2.0/3.0*R*X3)
1      -1.5/R*1.0/(1.0+A*SOL/(16.0*X1/X5/R))
1      *(X1)/RKMOM
1      *((CCX1)/RKMOM-(CCX4)
1      +2.0/3.0*R*CCX3))/0.637)/C0
C
C Coning kinematics
C
      SCCX2 = CCX3
C
C Flapping equation (From SSH European Forum Paper)
C Modified to include contribution from lag
C
      CCF3 =+2.0*(CCX5*X8*X2+X5*CCX8*X2+X5*X8*CCX2)
1      -CCX3*(X5-X8)
1      *LOCK/8.0*1.0/(1.0+3.142*SOL/(4.0*X1/X5/R))
1      -X3*(CCX5-CCX8)
1      *LOCK/8.0*1.0/(1.0+3.142*SOL/(4.0*X1/X5/R))
1      -X3*(X5-X8)
1      *LOCK/8.0*(-1.0)/(1.0+3.142*SOL/(4.0*X1/X5/R))**2
1      *3.142*SOL*R/4.0*(CCX5*X1-X5*CCX1)/X1**2
1      +CCX4/(6.0*R)*(X5-X8)
1      *LOCK*1.0/(1.0+3.142*SOL/(4.0*X1/X5/R))
1      +X4/(6.0*R)*(CCX5-CCX8)
1      *LOCK*1.0/(1.0+3.142*SOL/(4.0*X1/X5/R))
1      +X4/(6.0*R)*(X5-X8)
1      *LOCK*(-1.0)/(1.0+3.142*SOL/(4.0*X1/X5/R))**2
1      *3.142*SOL*R/4.0*(CCX5*X1-X5*CCX1)/X1**2
1      -CCX1/(6.0*R)*(X5-X8)
1      *LOCK*1.0/(1.0+3.142*SOL/(4.0*X1/X5/R))
1      -X1/(6.0*R)*(CCX5-CCX8)
1      *LOCK*1.0/(1.0+3.142*SOL/(4.0*X1/X5/R))
1      -X1/(6.0*R)*(X5-X8)

```

```

1      *LOCK*(-1.0)/(1.0+3.142*SOL/(4.0*X1/X5/R))**2
1      *3.142*SOL*R/4.0*(CCX5*X1-X5*CCX1)/X1**2
1      +CCTH0*2.0*(X5-X8)*(CCX5-CCX8)
1      *LOCK/8.0*1.0/(1.0+3.142*SOL/(4.0*X1/X5/R))
1      +CCTH0*(X5-X8)**2
1      *LOCK/8.0*(-1.0)/(1.0+3.142*SOL/(4.0*X1/X5/R))**2
1      *3.142*SOL*R/4.0*(CCX5*X1-X5*CCX1)/X1**2
1      - 2*X5*CCX5*(1.0+HOFF*R*MBTA/IBTA0)*X2
1      - X5**2*(1.0+HOFF*R*MBTA/IBTA0)*CCX2

```

C

```

      SCCX7 = CCX8
      CCF8 = 2.0*CCX2*X8*X3
1      +2.0*X2*CCX8*X3
1      +2.0*X2*X8*CCX3
1      -2.0*CCX5*X3*X2
1      -2.0*X5*CCX3*X2
1      -2.0*X5*X3*CCX2
1      -2.0*X5*CCX5*X7*HOFF*R*MBTA/IBTA0
1      -X5**2*CCX7*HOFF*R*MBTA/IBTA0
1      +1.0*( LOCK/(6.0*R))*((CCX5-CCX8)*(X1-X4)+(X5-X8)*
1      (CCX1-CCX4))*CCTH0
1      + LOCK/8.0*CCTH0*((CCX5-CCX8)*X3+(X5-X8)*CCX3)
1      - LOCK/(4.0*R**2)*2.0*(X1-X4)*(CCX1-CCX4)
1      - LOCK/(3.0*R)*(CCX1-CCX4)*X3
1      - LOCK/(3.0*R)*(X1-X4)*CCX3
1      - LOCK/8.0*X3*2*CCX3
1      + LOCK/(8.0*A)*DEL0*(X5-X8)*2.0*(CCX5-CCX8) )

```

C

C Plus lag damper terms

C

```

1      - AZETA*CCX8 - CZETA*CCX7

```

C

C Plus cosystem input

C

```

1      - X8

```

C

C Hub Force Equation (From SSH European Forum Paper)

C

```

CCF4 = CCX5*X3*NBLADE*IBTA0*LOCK/(6.0*MASS*R)
1      +X5*CCX3*NBLADE*IBTA0*LOCK/(6.0*MASS*R)
1      +CCX5*X1*NBLADE*IBTA0*LOCK/(4.0*MASS*R**2)
1      +X5*CCX1*NBLADE*IBTA0*LOCK/(4.0*MASS*R**2)
1      -CCX5*X4*NBLADE*IBTA0*LOCK/(4.0*MASS*R**2)
1      -X5*CCX4*NBLADE*IBTA0*LOCK/(4.0*MASS*R**2)
1      -2*CCX5*X5*CCTH0*NBLADE*IBTA0*LOCK/(6.0*MASS*R)

```

C

C Plus fuselage, tailplane, wing etc drag

C

```

1      +0.5*RHO*(X1-X4)*2.0*(CCX1-CCX4)*SFP*CDF/MASS

```

C

C Rotorspeed degree of freedom (ATMcC Same as Blackwell, Feik & Perrin)

C

C

```

CCF5 = NBLADE/IR*((IBTA0+HOFF*R*MBTA)
1      *( 2.0*CCX3*X2*(X5-X8)
1      + 2.0*X3*CCX2*(X5-X8)
1      + 2.0*X3*X2*(CCX5-CCX8))
1      +HOFF*R*MBTA*CCX7
1      *(2.0*X5*X8-X3**2-X8**2)
1      +HOFF*R*MBTA*X7
1      *(2.0*(CCX5*X8+X5*CCX8)
1      - X3*2.0*CCX3 - X8*2.0*CCX8))
1      -(
1      +2.0*RHO*3.1415927*R**2
1      *(1.0)/(1.0+A*SOL/(16.0*X1/X5/R))**2
1      *A*SOL*R/16.0*(CCX5*X1-X5*CCX1)/X1**2
1      *(X1)/RKMOM
1      *((X1)/RKMOM
1      -X4+2.0/3.0*R*X3))
1      *((X1)-X4)/X5/IR
1      -(2.0*RHO*3.1415927*R**2
1      *1.0/(1.0+A*SOL/(16.0*X1/X5/R))
1      *(CCX1)/RKMOM

```

```

1      * ( (X1) /RKMOM
1      -X4+2.0/3.0*R*X3) )
1      * ( (X1) -X4)
1      / (X5) / IR
1      - (2.0*RHO*3.1415927*R**2.0
1      *1.0/ (1.0+A*SOL/ (16.0*X1/X5/R) )
1      *X1/RKMOM
1      * ( (CCX1) /RKMOM
1      -CCX4+2.0/3.0*R*CCX3) )
1      * ( (X1) -X4)
1      / (X5) / IR
1      - (
1      +2.0*RHO*3.1415927*R**2.0
1      *1.0/ (1.0+A*SOL/ (16.0*X1/X5/R) )
1      * (X1) /RKMOM
1      * ( (X1) /RKMOM
1      -X4+2.0/3.0*R*X3) )
1      * ( (CCX1) -CCX4)
1      / (X5) / IR
1      - (4.0/3.0*RHO*3.1415927*R**3.0*SX1
1      +2.0*RHO*3.1415927*R**2.0
1      *1.0/ (1.0+A*SOL/ (16.0*X1/X5/R) )
1      * (X1) /RKMOM
1      * ( (X1) /RKMOM
1      -X4+2.0/3.0*R*X3) )
1      * ( (X1) -X4)
1      / ((-1.0)*X5**2)*CCX5/IR
1      +CCX6/IR

```

C

C Plus profile part

C

```

1      -RHO*3.142*(R**4.0)*3.0*(X5**2)*CCX5*(DEL0*SOL/8.0/X5/IR)
1      +RHO*3.142*R**4.0*X5**3.0*DEL0*SOL*CCX5/8.0/IR/(X5**2.0)

```

C

C

C Engine as lag

C

$$CCF6 = ENGGAI * (CCX5) + ENGTCO * CCX6$$

C

$$SCCTH0 = 0$$

C

$$X7X8 = X7 * X8$$

C

$$DETA = X2 * X7 * NBLADE / IR * HOFF * R * MBTA * X7 * X2 + 1 - MBTA / IBTA0 * NBLADE$$

$$1 \quad * MBTA * KMBTA / MASS$$

C

$$CCDETA = (CCX2 * X7 + X2 * CCX7) * NBLADE / IR * HOFF * R * MBTA * (CCX7 * X2 + X7$$

$$1 \quad * CCX2)$$

C

$$SCCX3 = -1.0 * CCDETA / (DETA ** 2) * (F3 - MBTA / IBTA0 * F4 + X2 * X7 * F5)$$

$$1 \quad + 1 / DETA * (CCF3 - MBTA / IBTA0 * CCF4 + CCX2 * X7 * F5 + X2 * CCX7$$

$$1 \quad * F5 + X2 * X7 * CCF5)$$

C

$$SCCX4 = -1.0 * CCDETA / (DETA ** 2) * (-NBLADE * MBTA * KMBTA / MASS * F3$$

$$1 \quad + (1 + X2 * X7 * NBLADE / IR * HOFF * R * MBTA * X7 * X2) * F4$$

$$1 \quad + (-X7 * X2 * NBLADE * MBTA * KMBTA / MASS) * F5)$$

C

$$1 \quad + 1 / DETA * (-NBLADE * MBTA * KMBTA / MASS * CCF3$$

$$1 \quad + ((CCX2 * X7 + X2 * CCX7) * NBLADE / IR * HOFF * R * MBTA *$$

$$1 \quad (CCX2 * X7 + X2 * CCX7)) * F4$$

$$1 \quad + (1 + X2 * X7 * NBLADE / IR * HOFF * R * MBTA * X7 * X2) * CCF4$$

$$1 \quad + (- (CCX2 * X7 + X2 * CCX7) * NBLADE * MBTA * KMBTA / MASS) * F5$$

$$1 \quad + (-X7 * X2 * NBLADE * MBTA * KMBTA / MASS) * CCF5)$$

C

$$SCCX5 = -1.0 * CCDETA / (DETA ** 2) * (-NBLADE / IR * HOFF * R * MBTA * X7 * X2 * F3$$

$$1 \quad + (MBTA / IBTA0 * NBLADE / IR * HOFF * R * MBTA * X7 * X2) * F4$$

$$1 \quad + (1 - MBTA / IBTA0 * NBLADE * MBTA * KMBTA / MASS) * F5)$$

$$1 \quad + 1 / DETA * (-NBLADE / IR * HOFF * R * MBTA * (CCX7 * X2 * F3$$

$$1 \quad + X7 * CCX2 * F3 + X7 * X2 * CCF3)$$

$$1 \quad + MBTA / IBTA0 * NBLADE / IR * HOFF * R * MBTA * (CCX7 * X2 * F4$$

$$1 \quad + X7 * CCX2 * F4 + X7 * X2 * CCF4)$$

$$1 \quad + (1 - MBTA / IBTA0 * NBLADE * MBTA * KMBTA / MASS) * CCF5)$$

C


```

SCCX6=-1.0*CCDETA/(DETA**2)*(ENGGAI*0.1*NBLADE/IR*HOFF*R
1      *MBTA*X7*X2*F3
1      +(-ENGGAI*0.1*MBTA/IBTA0*NBLADE/IR*HOFF*R*MBTA*X7*X2)*F4
1      +(ENGGAI*0.1*(1-MBTA/IBTA0*NBLADE*MBTA*KMBTA/MASS))*F5
1      +(X2*X7*NBLADE/IR*HOFF*R*MBTA*X7*X2+1-MBTA/IBTA0*NBLADE
1      *MBTA*KMBTA/MASS)*F6)
1      +1/DETA*(ENGGAI*0.1*NBLADE/IR*HOFF*R*MBTA*(CCX7*X2*F3
1      +X7*CCX2*F3+X7*X2*CCF3)
1      -ENGGAI*0.1*MBTA/IBTA0*NBLADE/IR*HOFF*R*MBTA*(CCX7*X2*F4
1      +X7*CCX2*F4+X7*X2*CCF4)
1      +(ENGGAI*0.1*(1-MBTA/IBTA0*NBLADE*MBTA*KMBTA/MASS))*CCF5
1      +((CCX2*X7+X2*CCX7)*NBLADE/IR*HOFF*R*MBTA*(CCX2*X7+X2
1      *CCX7)+1-MBTA/IBTA0*NBLADE
1      *MBTA*KMBTA/MASS)*CCF6)
C
SCCX8=-1.0*CCDETA/(DETA**2)*((1+HOFF*R*MBTA/IBTA0)*NBLADE
1      /IR*HOFF*R*MBTA*X7*X2*F3
1      +(-(1+HOFF*R*MBTA/IBTA0)*MBTA/IBTA0*NBLADE/IR*HOFF
1      *R*MBTA*X7*X2)*F4
1      +((1+HOFF*R*MBTA/IBTA0)*(1-
MBTA/IBTA0*NBLADE*MBTA*KMBTA
1      /MASS))*F5
1      +(X2*X7*NBLADE/IR*HOFF*R*MBTA*X7*X2+1-
MBTA/IBTA0*NBLADE
1      *MBTA*KMBTA/MASS)*F8)
1      +1/DETA*((1+HOFF*R*MBTA/IBTA0)*NBLADE/IR*HOFF*R*MBTA*
1      (CCX7*X2*F3+X7*CCX2*F3+X7*X2*CCF3)
1      -(1+HOFF*R*MBTA/IBTA0)*MBTA/IBTA0*NBLADE/IR*HOFF
1      *R*MBTA*(CCX7*X2*F4+X7*CCX2*F4+X7*X2*CCF4)
1      +((1+HOFF*R*MBTA/IBTA0)*(1-
MBTA/IBTA0*NBLADE*MBTA*KMBTA
1      /MASS))*CCF5
1
+((CCX2*X7+X2*CCX7)*NBLADE/IR*HOFF*R*MBTA*(CCX7*X2+CCX2
1      *X7))*F8
1      +(X2*X7*NBLADE/IR*HOFF*R*MBTA*X7*X2+1-
MBTA/IBTA0*NBLADE

```

```

1      *MBTA*KMBTA/MASS)*CCF8)
C
C
C End of equations defining system

```

Program listing for ESTIMATE3.SIM

```

C      This file calculates sums or errors and sensitivity
coefficients for
C      the optimisation routine in EXTERNALSCOMULTI.INC.
C      Initialise variables
IF (INIT.NE.0) THEN
    MERR2=0
    XE=0
    UU11=0
    UU12=0
    UU22=0
    UU23=0
    UU33=0
    UU13=0
    B1=0
    B2=0
    B3=0
    XSQ=0
ENDIF
IF (( ICHANG.EQ.1 ).AND. (RTIME.GT.0.0)) THEN
    U1=CCX7
    U2=DDX7
    U3=EEX7
    UU11=UU11+U1*U1
    UU12=UU12+U1*U2
    UU22=UU22+U2*U2
    UU23=UU23+U2*U3
    UU33=UU33+U3*U3
    UU13=UU13+U1*U3
    ERRR=LAGDAT-X7

```

```

RESID=ERRR
B1=B1+ERRR*U1
B2=B2+ERRR*U2
B3=B3+ERRR*U3
MERR2=MERR2+ERRR*ERRR
XXE=XE
XXSQ=XSQ
XMERR2=MERR2
ENDIF

```

Program listing for EXTERNALSMULTI.INC

```

C      INCLUDE file for helicopter optimisation example. This file is
included
C      in HELICOPTER.SIM.
C      It consists of several EXTERNALS,
C          1) Initialise ML variables
C          2) Calculate ML weights
C          3) Does parameter estimation for 2 parameters.
C          4) Discretise time delay parameter
C          5) Holds graphic screen until key is pressed and closes
disk
C          file.
C          6,7) AZETA=estma1
C              of sensitivity coefficients.
C          8) Finite difference variable assignment.
C          9,10) tbias=estma5
C          11,12) DAMPC=estma3
C          13) One parameter estimator
C          15) Three parameter estimator
C          17) Five parameter estimator
C          18,19) ENGGA1=estma1
C          20,21) CZETA=estma2
C          22,23) engtco=estma2
C          24) Four parameter estimator
C          25) Close file 27

```

```

C          26) Second stage of Box-Kanemasu algorithm
C
C      The Gauss-Newton method is used for the parameter estimation
searches
C
C
C      Initialise ML variables
4100  IF (TOT.EQ.0) THEN
        TOT=1.0
        WTOT1=1.0
        WTOT2=1.0
        WTOT3=1.0
        WTOT4=1.0
    ENDIF
    SIMSTOP
C
C      Calculate ML variables
4200  if (tot.ne.0.0) then
        W1=WTOT1/TOT
        W2=WTOT2/TOT
        W3=WTOT3/TOT
        W4=WTOT4/TOT
    else
        w1=1
        w2=1
        w3=1
        w4=1
    endif
    SIMSTOP
C
4250  FORMAT (G20.10)

4280  FORMAT (I5)
C
C      Get data from disk
C
C      Parameter estimation for three parameters

```

```

4300 DTRM=UU11*UU22-UU12*UU12
      IF (DTRM.EQ.0) THEN
        WRITE(*,4308)
        GOTO 4309
      ENDIF
4308      FORMAT(' SENSITIVITY COEFFICIENTS ARE ZERO')
      DQ1=1/DTRM*((UU22)*B1+(-UU12)*B2)
      DQ2=1/DTRM*((-UU12)*B1+(UU11)*B2)
      WRITE(5,4322)UU11,UU22,UU12,DQ1,DQ2
4322 FORMAT ('UU11,UU22,UU12,DQ1,DQ2=',5E15.5)
      CR1=(UU22/DTRM*MERR2/(numstp-1))
      CR2=(UU11/DTRM*MERR2/(NUMSTP-1))
      ESTMA1=ESTMA1+DQ1
      ESTMA2=ESTMA2+DQ2
4309 WRITE(5,4302) MERR2
4302 FORMAT(' SUM OF SQUARE ERRORS IS ',G20.10)
      WRITE(5,4303) ESTMA1,CR1
      WRITE(5,4304) ESTMA2,CR2
4303 FORMAT(' NEW ESTIMATE OF AZETA IS ',F20.10,
1          ' WITH ERROR OF ',E15.5)
4304 FORMAT(' NEW ESTIMATE OF CZETA IS ',F20.10,
1          ' WITH ERROR OF ',E15.5)
      S0=MERR2
      GK=DQ1*B1+DQ2*B2
      IF (GK.LT.0) THEN
        WRITE(*,4388)
4388 FORMAT(' GK IS LESS THAN ZERO')
        STOP
      ENDIF
      ALPHA=1.0
      AA=1.1
      SIMSTOP
C
4350 CONTINUE
      SIMSTOP
C
C      DISCRETISE PARAMETER DL1 - TIME DELAY

```

```

4400 DL1=7.8125E-3*INT((DL1+3.90625E-3)/7.8125E-3)
      IF (DL1.LT.0.0) DL1=0.0
      SIMSTOP
C
C      Hold graphics screen until key is pressed
4500 WRITE(5,4552)
4552 FORMAT(' CLEAR SCREEN, PRESS VDU KEY THEN RETURN.')
```

```

      READ(*,4553)
4553 FORMAT(A5)
      SIMSTOP
C
C      Azeta=estma1
4600      ESTMA1=AZETA
      SIMSTOP
C
4700 AZETA=ESTMA1
      SIMSTOP
C
C      calculations for finite difference calculation of sensitivity
coefficients
4800 FAAZET=AZETA+FINDA
      FBBIAS=BIAS+FINDB
      FCCZET=CZETA+FINDC
      FDENG=ENGGA1+FINDD
      FEENG=ENGTCO+FINDE
      FFKENG=KENG+FINDF
      SIMSTOP
C
4900      ESTMA5=TBIAS
      SIMSTOP
C
5000 TBIAS=ESTMA5
      SIMSTOP
C
5100      ESTMA3=DAMPC
      SIMSTOP
C
```

```

5200 DAMPC=ESTMA3
      SIMSTOP
C
C      One parameter parameter estimation
5300 HH=1.0
      IF (XSQ.EQ.0) THEN
          IF (UU11.EQ.0) THEN
              WRITE(*,5308)
              GOTO 5309
          ENDIF
          XSQ=UU11
          XE=B1
      ENDIF
5308      FORMAT(' SENSITIVITY COEFFICIENTS ARE ZERO')
      IF (XSQ.NE.0) THEN
          WRITE(*,5301)
          B1=XE
          DQ1=XE/XSQ
          ESTMA1=ESTMA1+DQ1
          CR1=1.0/XSQ*MERR2/NUMSTP
      ENDIF
5301 FORMAT(' TESTING...')
5309 WRITE(5,5302) MERR2
5302 FORMAT(' SUM OF SQUARE ERRORS IS ',G20.10)
      WRITE(5,5303) ESTMA1
      WRITE(5,5303) ESTMA1,CR1
5303 FORMAT(' NEW ESTIMATE OF AZETA IS ',F20.10,
1         ' WITH ERROR OF ',E15.5)
      S0=MERR2
      GK=DQ1*B1
      IF (GK.LT.0) THEN
          WRITE(*,5388)
5388 FORMAT(' GK IS LESS THAN ZERO')
          STOP
      ENDIF
      ALPHA=1.0
      AA=1.1

```

SIMSTOP

C

C Three parameter estimator

```

5500 DTRM=UU11*(UU22*UU33-UU32*UU23)-UU12*(UU21*UU33-UU23*UU31)+
      * UU13*(UU21*UU32-UU22*UU31)
      IF (DTRM.EQ.0) THEN
        WRITE(*,5508)
        GOTO 5509
      ENDIF
5508   FORMAT(' SENSITIVITY COEFFICIENTS ARE ZERO')
      DQ1=1/DTRM*((UU22*UU33-UU23*UU32)*B1+(UU23*UU31-UU21*UU33)*B2+
      * (UU21*UU32-UU22*UU31)*B3)
      DQ2=1/DTRM*((UU13*UU32-UU12*UU33)*B1+(UU11*UU33-UU13*UU31)*B2+
      * (UU31*UU12-UU11*UU32)*B3)
      DQ3=1/DTRM*((UU12*UU23-UU13*UU22)*B1+(UU13*UU21-
UU11*UU23)*B2+
      * (UU11*UU22-UU12*UU21)*B3)
      CR1=(UU22*UU33-UU23*UU32)/DTRM*MERR2/(NUMSTP-1)
      CR2=(UU11*UU33-UU13*UU31)/DTRM*MERR2/(NUMSTP-1)
      CR3=(UU11*UU22-UU12*UU21)/DTRM*MERR2/(NUMSTP-1)
      ESTMA1=ESTMA1+DQ1
      ESTMA2=ESTMA2+DQ2
      ESTMA3=ESTMA3+DQ3
5509 WRITE(5,5502) MERR2
5502 FORMAT(' SUM OF SQUARE ERRORS IS ',G20.10)
      WRITE(5,5503) ESTMA1,CR1
      WRITE(5,5504) ESTMA2,CR2
      WRITE(5,5505) ESTMA3,CR3
5503 FORMAT(' NEW ESTIMATE OF ENGGAI IS ',F20.10,' S.D.=',E15.5)
5504 FORMAT(' NEW ESTIMATE OF ENGTCO IS ',F20.10,' S.D.=',E15.5)
5505 FORMAT(' NEW ESTIMATE OF DAMPC IS ',F20.10,
1  ' S.D.=',E15.5)

```

C

C Additional code for Box-Kanemasu search

C

```

S0=MERR2
GK1=UU11*DQ1+UU12*DQ2+UU13*DQ3

```



```

GK2=UU21*DQ1+UU22*DQ2+UU23*DQ3
GK3=UU31*DQ1+UU32*DQ2+UU33*DQ3
GK=DQ1*GK1+DQ2*GK2+DQ3*GK3
IF (GK.LT.0) THEN
    WRITE(*,5588)
5588 FORMAT(' GK IS LESS THAN ZERO')
    SIMSTOP
ENDIF
ALPHA=1.0
AA=1.1
SIMSTOP
C
C      Five parameter estimator
5700 AAA(1,1)=UU11
    AAA(1,2)=UU12
    AAA(1,3)=UU13
    AAA(1,4)=UU14
    AAA(1,5)=UU15
    AAA(2,1)=UU12
    AAA(2,2)=UU22
    AAA(2,3)=UU23
    AAA(2,4)=UU24
    AAA(2,5)=UU25
    AAA(3,1)=UU13
    AAA(3,2)=UU23
    AAA(3,3)=UU33
    AAA(3,4)=UU34
    AAA(3,5)=UU35
    AAA(4,1)=UU14
    AAA(4,2)=UU24
    AAA(4,3)=UU34
    AAA(4,4)=UU44
    AAA(4,5)=UU45
    AAA(5,1)=UU15
    AAA(5,2)=UU25
    AAA(5,3)=UU35
    AAA(5,4)=UU45

```

```

      AAA(5,5)=UU55
C      MATRIX INVERSION ROUTINE
      N=5
      DO 5720 I=1,N
        DO 5710 J=1,N
5710      BBB(I,J)=0.0
5720 BBB(I,I)=1.0
        DO 5760 K=1,N
          DO 5740 I=1,N
            IF(I.NE.K) THEN
              Z=AAA(I,K)/AAA(K,K)
              DO 5730 J=1,N
                AAA(I,J)=AAA(I,J)-AAA(K,J)*Z
                BBB(I,J)=BBB(I,J)-BBB(K,J)*Z
5730      CONTINUE
              ENDIF
5740 CONTINUE
              Z=AAA(K,K)
              DO 5750 J=1,N
                AAA(K,J)=AAA(K,J)/Z
                BBB(K,J)=BBB(K,J)/Z
5750      CONTINUE
5760 CONTINUE
C
      DQ1=BBB(1,1)*B1+BBB(1,2)*B2+BBB(1,3)*B3+BBB(1,4)*B4
1    +BBB(1,5)*B5
      DQ2=BBB(2,1)*B1+BBB(2,2)*B2+BBB(2,3)*B3+BBB(2,4)*B4
1    +BBB(2,5)*B5
      DQ3=BBB(3,1)*B1+BBB(3,2)*B2+BBB(3,3)*B3+BBB(3,4)*B4
1    +BBB(3,5)*B5
      DQ4=BBB(4,1)*B1+BBB(4,2)*B2+BBB(4,3)*B3+BBB(4,4)*B4
1    +BBB(4,5)*B5
      DQ5=BBB(5,1)*B1+BBB(5,2)*B2+BBB(5,3)*B3+BBB(5,4)*B4
1    +BBB(5,5)*B5
C
      ESTMA1=ESTMA1+DQ1
      ESTMA2=ESTMA2+DQ2

```

```

ESTMA3=ESTMA3+DQ3
ESTMA4=ESTMA4+DQ4
ESTMA5=ESTMA5+DQ5
5709 WRITE(5,5702) MERR2
5702 FORMAT(' SUM OF SQUARE ERRORS IS ',G20.10)
WRITE(5,5703) ESTMA1
WRITE(5,5704) ESTMA2
WRITE(5,5705) ESTMA3
WRITE(5,5706) ESTMA4
WRITE(5,5707) ESTMA5
5703 FORMAT(' NEW ESTIMATE OF AZETA IS ',F20.10)
5704 FORMAT(' NEW ESTIMATE OF CZETA IS ',F20.10)
5705 FORMAT(' NEW ESTIMATE OF BIAS IS ',F20.10)
5706 FORMAT(' NEW ESTIMATE OF RBIAS IS ',F20.10)
5707 FORMAT(' NEW ESTIMATE OF TBIAS IS ',F20.10)
C
S0=MERR2
GK=DQ1*B1+DQ2*B2+DQ3*B3+DQ4*B4+DQ5*B5
IF (GK.LT.0.0) THEN
WRITE(*,5788)
5788 FORMAT (' GK IS LESS THAN ZERO')
STOP
ENDIF
ALPHA=1.0
AA=1.1
SIMSTOP
C
C      Clse data files
6500 CLOSE(27)
CLOSE(28)
CLOSE(29)
SIMSTOP
C
5800 ESTMA1=AZETA
SIMSTOP
C
5900 AZETA=ESTMA1

```

```
      SIMSTOP
C
6000   ESTMA3=DAMPC
      SIMSTOP
C
6100   DAMPC=ESTMA3
      SIMSTOP
C
6200   ESTMA2=CZETA
      SIMSTOP
C
6300   CZETA=ESTMA2
      SIMSTOP
C
C      Four parameter estimator
6400   HH=1
      AAA(1,1)=UU11
      AAA(1,2)=UU12
      AAA(1,3)=UU13
      AAA(1,4)=UU14
      AAA(2,1)=UU12
      AAA(2,2)=UU22
      AAA(2,3)=UU23
      AAA(2,4)=UU24
      AAA(3,1)=UU13
      AAA(3,2)=UU23
      AAA(3,3)=UU33
      AAA(3,4)=UU34
      AAA(4,1)=UU14
      AAA(4,2)=UU24
      AAA(4,3)=UU34
      AAA(4,4)=UU44
C      MATRIX INVERSION ROUTINE
      N=4
      DO 6420 I=1,N
        DO 6410 J=1,N
6410   BBB(I,J)=0.0
```

```

6420 BBB(I,I)=1.0
      DO 6460 K=1,N
        DO 6440 I=1,N
          IF(I.NE.K) THEN
            Z=AAA(I,K)/AAA(K,K)
            DO 6430 J=1,N
              AAA(I,J)=AAA(I,J)-AAA(K,J)*Z
              BBB(I,J)=BBB(I,J)-BBB(K,J)*Z
6430      CONTINUE
          ENDIF
6440 CONTINUE
        Z=AAA(K,K)
        DO 6450 J=1,N
          AAA(K,J)=AAA(K,J)/Z
          BBB(K,J)=BBB(K,J)/Z
6450      CONTINUE
6460 CONTINUE

```

C

```

DQ1=BBB(1,1)*B1+BBB(1,2)*B2+BBB(1,3)*B3+BBB(1,4)*B4
DQ2=BBB(2,1)*B1+BBB(2,2)*B2+BBB(2,3)*B3+BBB(2,4)*B4
DQ3=BBB(3,1)*B1+BBB(3,2)*B2+BBB(3,3)*B3+BBB(3,4)*B4
DQ4=BBB(4,1)*B1+BBB(4,2)*B2+BBB(4,3)*B3+BBB(4,4)*B4

```

C

```

      ESTMA1=ESTMA1+DQ1
      ESTMA2=ESTMA2+DQ2
      ESTMA3=ESTMA3+DQ3
      ESTMA4=ESTMA4+DQ4
6409 WRITE(5,6402) MERR2
6402 FORMAT(' SUM OF SQUARE ERRORS IS ',G20.10)
      WRITE(5,6403) ESTMA1
      WRITE(5,6404) ESTMA2
      WRITE(5,6405) ESTMA3
      WRITE(5,6406) ESTMA4
6403 FORMAT(' FIRST ESTIMATE OF ENGGAI IS ',F20.10)
6404 FORMAT(' FIRST ESTIMATE OF ENGTCO IS ',F20.10)
6405 FORMAT(' FIRST ESTIMATE OF ENGFF IS ',F20.10)
6406 FORMAT(' FIRST ESTIMATE OF DELAY IS ',F20.10)

```

```

S0=MERR2
GK1=UU11*DQ1+UU12*DQ2+UU13*DQ3+UU14*DQ4
GK2=UU12*DQ1+UU22*DQ2+UU23*DQ3+UU24*DQ4
GK3=UU13*DQ1+UU23*DQ2+UU33*DQ3+UU34*DQ4
GK4=UU14*DQ1+UU24*DQ2+UU34*DQ3+UU44*DQ4
    GK=DQ1*GK1+DQ2*GK2+DQ3*GK3+DQ4*GK4
    IF (GK.LT.0) THEN
        WRITE(*,6488)
6488 FORMAT(' GK IS LESS THAN ZERO')
    ENDIF
    ALPHA=1.0
    AA=1.1
    SIMSTOP
C
C      Second stage of Box-Kanemasu search algorithm
6600 S1=MERR2
    ESTMA1=ESTMA1-HH*DQ1
    ESTMA2=ESTMA2-HH*DQ2
    ESTMA3=ESTMA3-HH*DQ3
    ESTMA4=ESTMA4-HH*DQ4
    ESTMA5=ESTMA5-HH*DQ5
    IF (S1.GT.S0) THEN
        HH=0.5*HH
        ALPHA=HH
        WRITE(*,6668) HH
        IF (HH.LT.0.01) THEN
            WRITE (*,6666) HH,S0,S1,GK
            SIMSTOP
        ENDIF
        GOTO 6667
    ENDIF
6666 FORMAT(' ALPHA IS LESS THAN 0.01, ALPHA=',F20.10,' S0=',
1 F20.10,' S1=',F20.10,' GK=',F20.10)
6668 FORMAT(' DO AN @LOOP2, ERROR INCREASED, HH=',F20.10)
    HH=ALPHA*AA
    S2=S0-ALPHA*GK*(2-1/A)
    IF (S1.GE.S2) THEN

```

```

      HH=ALPHA**2*GK/(S1-S0+2.0*ALPHA*GK)
ENDIF
CR1=(UU22*UU33-UU23*UU32)/DTRM*MERR2/(NUMSTP-1)
CR2=(UU11*UU33-UU13*UU31)/DTRM*MERR2/(NUMSTP-1)
CR3=(UU11*UU22-UU12*UU21)/DTRM*MERR2/(NUMSTP-1)
ESTMA1=ESTMA1+HH*DQ1
ESTMA2=ESTMA2+HH*DQ2
ESTMA3=ESTMA3+HH*DQ3
6609 WRITE(5,6602) MERR2
6602 FORMAT(' SUM OF SQUARE ERRORS IS ',G20.10)
      WRITE(5,6603) ESTMA1,CR1
      WRITE(5,6604) ESTMA2,CR2
      WRITE(5,6605) ESTMA3,CR3
6603 FORMAT(' NEW ESTIMATE OF ENGGAI IS ',F20.10,' S.D.=' ,E15.5)
6604 FORMAT(' NEW ESTIMATE OF ENGTCO IS ',F20.10)
6605 FORMAT(' NEW ESTIMATE OF ENGINE FEEDFORWARD IS ',F20.10,
      1  ' S.D.=' ,E15.5)
6667 CONTINUE
      SIMSTOP
C
6700 HH=1.0
      SIMSTOP

```

Variables

x1	Induced velocity/ Inflow (normlised)
x2	Beta0, Multiblade Flapping angle
x3	Beta0dot. Multiblade flapping velocity
x4	Vertical Velocity (of hub and aircraft)
x5	Rotorspeed
x6	Engine Torque
x7	Zeta0, Multiblade Lag Angle
X8	Zeta0dot, Multiblade lag velocity
INIT	system flag indicating first pass of simulation
LOCK	Lock number
RHO	air density

A	lift curve slope
C	rotor blade chord
R	rotor blade radius
IBTA0	flapping moment of inertia of blade
NBLADE	number of rotor blades
MBTA	blade lag moment of inertia
MASS	aircraft mass
SOL	main rotor solidity
SX1	signifies time derivative of X1 (as with all states)
TH0	main rotor collective pitch angle
RKMOM	momentum correction factor used in inflow
C0	apparent additional air mass for inflow
HOFF	flapping hinge offset
DELO	blade profile drag coefficient
AZETA	lag damper damping coefficient
CZETA	lag damper spring constant
DAMPC	lag damper friction coefficient
KMBTA	inertia correction factor
SFP	effective aircraft frontal area in heave
CDF	aircraft drag coefficient
IR	main rotor moment of inertia
TEMP4	trim variable for rotor speed dynamics
UPUT	inflow velocity at blade element
CD	drag factor for blade element
FDAMP	lag force due to lag damper
ENGGA1	gain of rotor speed controller
ENGTCO	time constant of rotor speed controller and engine
H	aircraft altitude
CCX1	derivative of X1 with respect to parameter AZETA (same for all states)
MERR2	sum of squares of the error between model output and flight data
XE	sum of product of sensitivity coefficient and output error
XSQ	sum of square of sensitivity coefficient
Ux	sensitivity coefficient for parameter x
UUxy	sum of products of sensitivity coefficients x and y
Bx	product of sensitivity coefficient x and output error

ICHANG system flag indicating if first pass this integration interval
 RTIME time in simulation
 CCX7 sensitivity coefficient for parameter AZETA and output lag angle
 ERRR error between flight data and model output
 WTOT1 etc totals used in estimation process
 Wx maximum Likelihood parameter estimation weights
 DTRM determinant of sensitivity matrix
 DQx change in parameter calculated from parameter estimation
 CRx Cramer-Rao bound for parameter x
 ESTMAX parameter being estimated
 S0 sum of square errors
 DL1 discrete time delay
 BIAS bias error in data
 KENG additional engine parameter

Notes

States are defined as dynamic variables in the system which are integrated. Nodes are defined as input/output variables or summing junction variables. Equations are stated $S\langle\text{variable}\rangle = \langle\text{expression}\rangle$.

If the variable is a state this is a differential equation and the S represents the differential operator. If the variable is a node, it is evaluated as an algebraic expression.

References

- 1 *PRO-MATLAB user's guide*, The MathWorks, Inc., MA., pp3-84 - 3-88, June, 1986.
- 2 TSIM, Cambridge control Ltd.

