Fault-Tolerant Flight Control System Design with Application to a Bell-205 Helicopter

Thesis submitted for the degree of
Doctor of Philosophy
at the University of Leicester

by

Mohammad F. Al-Malki
B.Sc. EE (UPM), M. Sc. Electrical & Computer Eng (KAAU)
Department of Engineering
University of Leicester

June 2004

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

Abstract

This thesis addresses the topic of fault-tolerant flight control system (FTFCS) design and focuses on its application to the Bell-205 helicopter. In this context, a fault detection, isolation, and accommodation (FDIA) system has been constructed using artificial neural networks (ANNs) and real flight test data (FTD). The construction of the ANN-based FDIA system considers all the feedback sensors but does not use any of the sensor measurements in the input space of the ANNs. This latter feature increases the reliability of detection and isolation of faults. Desktop simulations of the FDIA system have shown highly acceptable performance. Robust controllers have been designed for the lateral and longitudinal dynamics using the H_∞ mixed-sensitivity approach. The controllers were then integrated with the aforementioned FDIA systems and tested in simulation. The inspection of various uncertainties, including those due to the presence of the FDIA in the feedback loop, indicated that μ-synthesis may give better results than the H_∞ design. Therefore, an improved FTFCS system was designed using μ-synthesis. The functioning of the integrated systems is acceptable but their accuracy needs to be further verified on the nonlinear model. Use of μ-synthesis helped to identify areas for further improvement.

In addition to the design work that was carried out in the thesis, a theoretical investigation was conducted to study the impact of the faults on the Algebraic Riccati Equations (AREs) that are normally solved when finding stabilizing H_∞ controllers. Accordingly, a new FTFCS scheme which is based on solving a new set of AREs is proposed. The solutions of the Riccati equations are corrected adaptively. The controller has a particular structure and only certain parts of it need updating. The advantages of this approach include possible smooth transfer when updating and less computations when compared with switched controllers.

Acknowledgment

I am greatly indebted to my supervisor Dr. Da-Wei Gu for his highly constructive and greatly influential leadership throughout my PhD study. He provided me with a lot of support and generously sorted out many technical problems. His approach has enlightened the way I do research.He instilled in me a great deal of research knowledge and experience that will last and grow in the rest of my life.

As well, I'd like to thank my co-supervisor Dr. Declan G. Bates for his valuable comments, guidance and informative discussion sessions that have opened many avenues which have benefited the research project. I'd like to acknowledge the valuable assistance, encouragement, and continuous support of my local supervisor Dr. Mustafa Al-Shafie at KFUPM. He has extended unlimited support and enjoyable discussion hours with him in his office and at his home. Thanks are extended to Dr. AbdulWahid Saif (KFUPM) for technical assistance and for the literature he provided. I'd like to thank Bill Gubbels from NRC flight research lab (Canada) for detailed clarification of the content of the flight test data which helped to finalize many tasks pertaining to fault detection and isolation design.

Thanks go to Dr. M Turner for his assistance in preparing the Bell-205 flight test data (FTD) and time availed to discuss certain aspects of the FTD and flight testing in general.

My colleagues in the research lab are greatly thanked for their valuable assistance. Particularly, I like to thank Khalid Khan for his valuable LaTex assistance, Dr.Husam Baki for valuable discussions on helicopter dynamics when I first joined, and Dr. Giduo Herman for his keen interest to help and for valuable discussions with him. I have benefitted from a lot of experts in the field with whom I have communicated over the net; I'd like to thank

them all.

It 's my great pleasure to be in this academic community which was ranked 5A in RAE2001, 4^{th} Engineering Dept in UK, and is BAe's preferred Academic Capability Partner in control systems.

Finally, I would like to express my gratitude to my wife and sons for their valuable patience, encouragement, and sacrifice without which it would not be possible to complete the thesis. My son Moaath deserves special thanks for preparing some drawings.

Contents

1 **Fault-Tolerant Flight Control System Overview** 15

1.1 The FTFCS Concept 16

1.2 Overview of FTFCS Subsystems 18

1.3 FTFCS Methods and Tools 19

1.3.1 Model-based vs. Model-free FDIA 20

1.3.2 Fixed Controller FTFCS vs. Adaptive FTFCS 21

1.4 Application of FTFCS Concept 21

1.5 Robust Control vs Fault-Tolerant Control 23

1.6 Contribution and Thesis Organization 23

1.6.1 Thesis Contribution 23

1.6.2 Thesis Organization 24

2 **Bell-205 Helicopter** 27

2.1 Helicopter Dynamics 27

2.1.1 Nonlinear Model 29

2.1.2 Linearization of Nonlinear Model 39

2.2 Bell-205 Helicopter 41

2.2.1 NASA Model 43

2.2.2 DERA Model 45

2.3 Model Validation 47

2.3.1 Flight Test Data - FTD 48

2.4 Conclusion 49

3 Artificial Neural Networks (ANNs) 50

3.1 ANN Architecture 50

3.1.1 Structure of a Single Neuron 51

3.1.2 MLP Structure 52

3.1.3 Recurrent Neural Network (RNN) 53

3.2 ANN Training 54

3.3 Function Approximation 57

3.4 Application of ANN in Control Systems 58

3.4.1 Neuro Modelling 59

3.4.2 Neuro Modelling Guideline 60

3.5 Why use ANN in FDIA? 62

3.6 Conclusion 63

4 Fault Detection, Isolation and Accommodation(FDIA) 64

4.1 FDIA Concepts 65

4.2 FDIA Design 65

4.2.1 Model-based FDI 66

4.2.2 Model-free FDI 67

4.3 Systems Identification Principles 68

4.4 ANN-Based FDI 71

4.4.1 Fault Accommodation 73

4.4.2 Neural Network Architecture for Fault Detection and Isolation (FDI) 74

4.4.3 NN Input Space 76

4.5 ANN-based FDI for Bell-205 Helicopter 77

4.5.1 ANN-based FDIA for Bell-205 Helicopter - Lateral Case 78

4.5.2 The Lateral Decision Logic Module (DLM) 82

4.5.3 ANN-Based FDIA for Bell-205 Helicopter - Longitudinal Case . 84

4.5.4 The Longitudinal Decision Logic Module (DLM) 86

4.5.5 Threshold Selection 86

4.6 ANN Training and Testing Results 88

4.7 Statistical Analysis 96

4.7.1 Lateral ANN-based FDIA 97

4.7.2 Longitudinal ANN-based FDIA 101

4.8 FDIA Simulation Results 105

4.9 Conclusion 109

5 Integration of ANN-Based FDI and H_∞ Controller 111

5.1 H_∞ controller Design 112

5.1.1 Mixed-Sensitivity Design 115

5.1.2 Bell-205 Helicopter Lateral Controller 116

5.1.3 Bell-205 Longitudinal Controller 120

5.2 Integrating Controllers with the FDI 122

5.2.1 Lateral Case 122

5.2.2 Longitudinal Case . . . 126

5.3 ANN-based FDI using Linear Model Data . . . 129

5.3.1 Limited Version of FTFCS Simulation Results . . . 132

5.4 Performance and Robustness Analysis . . . 134

5.5 Conclusion . . . 138

6 Fault-Tolerant Controller using μ Analysis and Synthesis 139

6.1 Introduction . . . 139

6.2 μ Synthesis . . . 141

6.3 μ Controller Design for Bell-205 Helicopter . . . 143

6.3.1 Uncertainty Modelling . . . 144

6.3.2 Weighting function Selection . . . 146

6.3.3 Performance and Stability Analysis . . . 149

6.4 Simulation Results . . . 153

6.5 Conclusion . . . 159

7 ARE-Based Adaptive FTFCS 161

7.1 ARE and H_∞ control . . . 161

7.2 ARE Solutions . . . 163

7.2.1 Invariant Subspace Method . . . 163

7.2.2 Deflating Subspace Method . . . 165

7.2.3 Sign Matrix Method . . . 166

7.2.4 Iterative Method . . . 166

7.3 Controller Reconfiguration in FTFCS . . . 167

7.3.1 On-Line Design and Switching . . . 168

7.3.2 An Adaptive Compensation 171

7.3.3 Adaptive Scheme Selection 177

7.4 Conclusion 179

8 Conclusion 180

8.1 Achievements 180

8.2 Recommendations For Future Research 181

A ANN Details 184

A.1 Introduction 184

A.2 Lateral Dynamics FDIA Neural Netorks Details 184

A.3 Longitudinal Dynamics FDI Neural Networks Details 192

B Simulation Environment 199

B.1 Computing Facility 199

B.2 Testing and Validation Files 200

C Flight Test Data (FTD) Details 202

D A Result on the Rank of Solutions to a Kind of Algebraic Riccati Equation 208

List of Figures

1.1 Fault-Tolerant Control System Architecture 18

2.1 Helicopter Subsystems with Forces and Moments 28

2.2 Coordinate Systems 32

3.1 A Single Neuron Diagram showing inputs, weights, bias and mapping function 51

3.2 MLP Architecture 52

3.3 RNN Architecture 54

3.4 BP Algorithm Diagram 56

3.5 Generic Neural Model 60

3.6 Comprehensive Neural Model 61

4.1 Parallel Identifier Model 70

4.2 Parallel-Series Identifier Model 71

4.3 Fault Representation 80

4.4 Lateral Dynamics FDIA Simulink Model 81

4.5 Decision Logic Module (DLM) Flowchart 83

4.6 Longitudinal Channel Simulink Model 85

4.7 Master ANN for Lateral Dynamics Model (LANOF) - after training vs. FTD 89

4.8 MISO ANN used to model Roll Attitude ϕ to mimic phi-sensor output (phiNN) After training vs. FTD 90

4.9 MISO ANN used to model Roll rate p to mimic p-sensor output (pNN) After training vs. FTD 90

4.10 MISO ANN used to model Yaw Rate r to mimic r-sensor output (rNN) After training vs. FTD 91

4.11 Master ANN for Lateral Dynamics Model (LANOF) - after validation vs. FTD 92

4.12 MISO ANN used to model Roll Attiude ϕ to mimic phi-sensor output (phiNN) after validation vs. FTD 92

4.13 MISO ANN used to model Roll Rate p to mimic p-sensor output (pNN) after validation vs. FTD 93

4.14 Master ANN for Longitudinal Dynamics Model (LONOF) - after training vs. FTD 93

4.15 MISO ANN used to model Pitch Rate q to mimic q-sensor output (qNN) after training vs. FTD 94

4.16 MISO ANN used to model Pitch Attiude θ to mimic θ-sensor output (thetaNN) after training vs. FTD 94

4.17 Master ANN for Longitudinal Dynamics Model (LONOF) - after validation vs. FTD 95

4.18 Effects of Noise on Quality of Prediction 96

4.19 Typical Error Autocorrelation Plot 97

4.20 LANOF Error and Error Autocorrelation Plot 97

4.21 LANOF Error and Error Autocorrelation Plot - Collective' 98

4.22 phiNN Error and Error Autocorrelation Plot - Collective' 99

4.23 pNN Error and Error Autocorrelation Plot - Collective' 100

4.24 rNN Error and Error Autocorrelation Plot - Collective' 101

4.25 LONOF Error and Error Autocorrelation Plot 102

4.26 LONOF Error Plot and Error Autocorrelation Plot 103

4.27 Error Plot and Error Autocorrelation Plot for thetaNN 104

4.28 Error Plot and Error Autocorrelation Plot for qNN 105

4.29 Lateral Sensors' Measuremnts with one fault occurs at time t= 2 seconds . 106

4.30 Lateral Sensors' Measuremnts with two Simultaneous faults occurring at time t= 2 seconds 107

4.31 Successfull Fault Detection, Isolation, and Accommodation of Multiple Simultaneous Faults 107

4.33 Successfull Fault Detection, Isolation, and Accommodation of θ signal . 108

4.32 Longitudinal Sensors' Measuremnts with two Simultaneous Faults 108

5.1 Plant-Controller Architecture with Sensor FDIA in the loop 112

5.2 The Standard H_∞ Configuration 114

5.3 Mixed-sensitivity H_∞ Controller 116

5.4 Lateral Frequency Response 119

5.5 Lateral Unit Step Response 119

5.6 Longitudinal Frequency Response 121

5.7 Longitudinal Unit Step Response 122

5.8 Lateral FTFCS for Bell-205 Helicopter 123

5.9 Successful FDIA with the H_∞ Controller in the loop 124

5.10 Comparison of Linear Model and Flight Test Data 126

5.11 Longitudinal FTFCS for Bell-205 Helicopter 127

5.12 Closed Loop System without ANN-based FDIA 128

5.13 Simulation Results of Closed loop without ANN-Based FDIA 128

5.14 Master ANN (LONOF) Outputs After Training vs FTD 130

5.15 Pitch Attitude ANN (thetaANN) Output After Training vs. FTD 131

5.16 Pitch Rate ANN (qNN) Output vs FTD 131

5.17 Simulation Result of Fault at t=0 sec with Random Magnitude on pitch channel 132

5.18 Simulation Result of Fault at t=50 sec with Magnitude=1.5 on Pitch Attitude Channel 133

5.19 Simulation Result of Fault at t=50 sec with Magnitude=1.5 on Pitch Rate channel 134

5.20 Fault-Tolerant Control System - Fixed Controller Case 135

5.21 Closed-Loop with FDIA Model 136

5.22 Closed-Loop with FDIA Model - SGT view 137

5.23 μ Analysis Robust Stability Output 138

6.1 General Control Configuration 141

6.2 M-Δ Configuration 142

6.3 Design Problem Formulation Diagram 144

6.4 General Block Diagram 149

6.5 μ Controller with all uncertainties Performance and Robustness Analysis Results 150

6.6 H_∞ Controller Performance and Robustness Analysis 150

6.7 Nominal and Robust Performance and Stability with 50% Improvement in FDI) 152

6.8 Nominal and Robust Performance and Stability with No Delay 152

6.9 Nominal and Robust Performance and Stability with 50% improvement in Roll Channel 153

6.10 Integration of ANN-based FDI with μ Controller - Lateral Case 154

6.11 Master ANN (LANOF) Performance vs Linear Model 154

6.12 Roll Attitude ANN (phiNN) performance vs Linear Model 155

6.13 Roll Rate ANN (pNN) performance vs Linear Model 155

6.14 Yaw Rate ANN (rNN) performance vs Linear Model 156

6.15 Simulation Results of faults occurred at t=9.985 sec with Magnitude=2.0 . 157

6.16 Simulation Results of faults occurred at t=1 sec with Magnitude=2.0 . . . 158

6.17 Simulation Results of faults occurred at t=10 sec with Magnitude=2.0 . . 159

7.1 The Standard H_∞ Configuration 162

7.2 H_∞ S/KS Mixed-sensitivity Problem Configuration 169

7.3 General Control Configuration 170

7.4 Adaptive FTFCS Controller Overall Structure 175

7.5 Adaptive FTFCS - First Controller (K_1) 176

7.6 Adaptive FTFCS - Second Controller (K_2) 176

7.7 Adaptive FTFCS - Third Controller (K_3) 177

7.8 Adaptive Schemes 178

List of Tables

2.1 States, Input, and Output Symbols 36

2.2 Input/Output Pairing Relationship 42

2.3 Sensors's Minimum and Maximum Values 42

2.4 The NASA Linear Model trim angles 44

2.5 NASA Linear Model trim velocities 44

2.6 Longitudinal Sates, Outputs, and Inputs 46

2.7 Lateral States, Outputs, and Inputs 47

4.1 Bell-205 Helicopter Lateral FDIA Design Parameters 79

4.2 Bell-205 Helicopter Lateral FDIA Neural Networks Input Space 79

4.3 Bell-205 Helicopter Longitudinal FDIA Design Parameters 84

4.4 Bell-205 Helicopter FDIA Neural Networks Input Space 84

4.5 Neural Networks Threshold Values 88

4.6 Error bound for LANOF ANN 98

5.1 Input Space for ANN-based FDIA using Linear Model Data 130

6.1 H_∞ and μ Controllers Comparison 151

B.1 Training and Validation Flight Test Data Sets 200

C.1 FTD Variables - Continued 205

C.2 FTD Variables - Continued 206

C.3 FTD Variables - Continued 207

Symbols and Abbreviations

Mathematical Symbols

Symbol	Meaning
$A > 0$	A is positive definite
$A \geq$	A is positive semi-definite
$A < 0$	A is negative definite
$A \leq 0$	A is negative semi-definite
I	Identity matrix of appropriate dimension
A^T	Transpose of matrix A
$\|G\|_\infty$	$sup_\omega \bar{\sigma}(G(j\omega))$ where G is a transfer function of jw
$\left[\begin{array}{c\|c} \mathrm{A} & \mathrm{B} \\ \hline \mathrm{C} & \mathrm{D} \end{array}\right]$	State-space representation of a continuous system $C(sI - A)^{-1}B + D$

Abbreviations

Abbreviation	Meaning
FDI	Fault Detection and Isolation
FDIA	Fault Detection, Isolation and Accommodation
FTC	Fault Tolerant Control
FCS	Flight Control System
FTFCS	Fault Tolerant Flight Control System
ARE	Algebraic Riccati Equation
LFT	Linear Fraction Transformation
MIMO	Multi Inputs and Multi Outputs
MISO	Multi Inputs and Single Output
MSE	Mean-Square Error

Modelling Symbols

Symbol	Meaning
V_T	Tangential Velocity
F	Force Vector
M	Moment Vector
H	Angular Momentum
ω	Angular Velocity
p,q and r	Angular velocity components around the x, y, and z-axis
u, v, and w	Linear velocity components along the x, y, and z-axis
R	Main Rotor Radius
X_{hw}, Y_{hw} Z_{hw}	Main Rotor force components with respect to hub-wind axis
L_{HQ} ,M_{HQ}, and N_h	Main Rotor moments about the hub-wind axis
X_T, Y_T Z_T	Tail Rotor force components
L_T ,M_T, and N_T	Tail Rotor moments
T_T	Tail Rotor Thrust
Q_T	Tail Rotor torque
l and d	Lift and Drag force of the blades
N_b	Number of blades
r_b	Blade radius β
Blade flap angle	
β_{1c}	First harmonic cosine
β_{1s}	First harmonic sine
α_f	Incidence of resultant velocity to fuselage (rad)
β_f	Sideslip angle at fuselage (rad)
f_y and f_z	Main Rotor in-plane and out-of-plane aerodynamic load on rotor blades
$\dot{\Omega}$	angular acceleration
Q_R	Main Rotor Torque

Chapter 1

Fault-Tolerant Flight Control System Overview

Fault-tolerant control has been motivated mainly by aerospace applications and been applied to highly sensitive industrial applications such as chemical plants. With the increased interest in making safe flights, the reliability of various components of the aircraft have been re-engineered with this reliability issue in mind. Various techniques have been applied ranging from physical redundancy to software-based methods. Though physical redundancy has been in use in the aerospace industry and petrochemical plants, such a solution is not of great appeal to some aerospace applications or in small chemical plants where the cost of physical redundancy is prohibitive. This gives more motivation to researchers to focus on software-based approaches that besides their relatively lower cost of development are functionally promising. In this chapter, we aim to cover the basic principles of FTFCS and its relationship to other disciplines such as robust control. The fundamental components will be explained and linked to the appropriate chapter of the thesis.

1.1 The FTFCS Concept

Faults that are likely to occur in an aircraft are either related to its sensors, actuators or the components of the aircraft itself (e.g. the airframe). Each has its impact on the performance and safety of the aircraft. It is expected that not all failures are of the same degree of urgency nor that they all lead to catastrophic events. Here below we will touch on the impact of failure on system reliability and operation.

- **Sensor faults**: Sensors are devices that convey extensive amounts of information to the panels of the pilot as well as to the flight computer system. The most critical pieces of information are those pertaining to the feedback loops. A flight control system, as a closed-loop, relies heavily on the measurements of the outputs. Based on these measurements the control signals are modified to cope with the pilot inceptors' commands. Of course, the pilot uses his eyes to watch the impact of his commands but that can by no means be considered as a replacement of the measured signals. His observation may alert him of occurrence of a problem but his feelings can not be quantified and injected properly into the automatic control loop. In some cases, the malfunctioning of sensors can lead to catastrophic situations especially when flying under hard weather conditions and/or at high altitudes. Despite the fact that modern sensors are of high quality and their Mean Time Between Failure (MTBF) is high, still there is a possibility of sensor failure.
- **Actuator faults**: This is another source of prime reasons for aircraft crashes. If certain control services can not be controlled then certain manoeuvres can not be made. In a combat aircraft, if this happens the pilot will not only face the impact of being unable to perform aggressive maneuvering but also be unable to make right move at a specific and crucial point of time. One can imagine the end result which may be either a complete crash or becoming an easy target of attack. Another example, from the helicopter domain, is the failure of actuators responsible for controlling the rods of the blades of the main rotor. As an option, in this case, the pilot has to do autorotation; i.e. to switch off the power plant and bring the helicopter to a safe landing. Other option could be to switch to mechanical control

if the helicopter actuators are equipped with dual control; i.e. electronically as well as mechanically. In the case of a fixed-wing aircraft, the existence of a solution to address the failure of control services actuators is of practical value. Such a solution relies on what is called the analytic redundancy of the aircraft. In the case of a helicopter, the presence of actuator failure detection system may not have so much practical appeal except alerting the pilot to switch to mechanical control, if ever possible.

- **Airframe damage**: There are various reasons for considering airframe damage. Not only fighters in the battle field do face the risk of partial damages (say, e.g. wing or tail partial damage) but also civil aircraft may face such kind of trouble. Solutions have been sought and promising results have been reported (see e.g. [27]).

Each aspect of the above is of prime importance and implementation of a solution for each depends on the type of aircraft under consideration. For fixed-wing aircraft, it is quite applicable and feasible to implement a system for each category. However, for helicopters, actuator faults may not be feasible to be recovered fully, as in the case for fixed-wing aircraft due to the limited mechanical architecture of the helicopter. Degraded operation of actuators may have to be considered.

The main objective of FTFCS is to enhance **reliability** of the system under the presence of faults. For that reason, the term FTFCS is used exchangeably with Reliable Control. Other terminology used are restructuring and reconfiguration. The terms have emerged as a result of the approach used in dealing with the faults which involves restructuring or reconfiguration of the controller. This reconfiguration/restructuring may take place on-line or may be designed off-line and controllers are switched depending on the signature of faults.

In view of the above, the focus of our research will be on sensor faults especially for the Bell-205 Helicopter which is not equipped with any physical redundant sensors. As has been said above, this does not eliminate the need to consider the other two aspects. For example, the actuator degradation can be considered. Also, the airframe damage is of value especially for military helicopters. However, this option was eliminated at an early stage of the project as it was not clear that access to the nonlinear model (owned by QinetiQ) and also the modelling tools of the helicopter would be available. If airframe damage occurs, the nonlinear model no longer represents the aircraft as the stability derivatives and control derivatives no longer coincide with the operating condition of the aircraft. Some fast tools have to be used on-line to estimate these derivatives and to use the new estimates to generate a new model that will help to save the aircraft and the crew though the performance may be degraded. Either one of these two other aspects is worthy of further independent research effort.

1.2 Overview of FTFCS Subsystems

By now the objectives and functionality of the FTFCS are becoming clear. The FTFCS has to be able to deal with all or part of the above sources of failures or damages. Thus, a FTFCS can be represented by the following diagram (Figure 1.1):

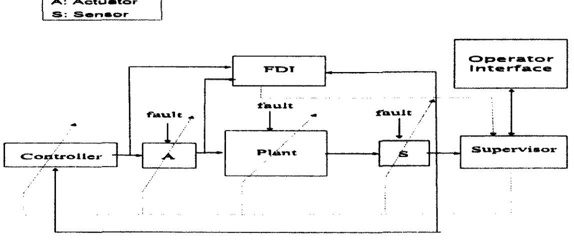


Figure 1.1: Fault-Tolerant Control System Architecture

From the diagram, it is clear that for a flight control system or any control system in general to be fault-tolerant it needs to be able to detect faults, identify their sources and take remedial actions promptly. The fault detection and isolation is handled by a Fault Detection and Isolation (FDI) subsystem. The accommodation can either be handled in the FDI subsystem or be embedded in the controller. The FDIA can communicate with a system operator who is in our case the pilot and various alarms can be produced in the Head Up Display (HUD). In regard to the FDI system, the field is quite matured and various methods have been developed. The designer may choose either model-based or model-free techniques where under each category various methods are available. Most of the well developed methods in the FDI field are linear and use linear models. Other methods do not require a model and have been used widely in process control and in some aerospace applications such as large transport aircraft. The controller which is the second major component in the setup, can be designed using robust controller methods or using adaptive techniques. If robust control is used, the resultant controller is a fixed controller and, thus, the fault accommodation is handled by the FDI subsystem. In the case of adaptive control, the resultant controller is flexible as some of its parameters are updated depending on the fault signatures constructed by fault identification and isolation modules that are conveyed to the FTFCS decision logic.

1.3 FTFCS Methods and Tools

There are many approaches to remedy the faulty case and bring the flight control system to a level of fault-tolerance. The field itself is intra-control disciplinary. Many researchers with different backgrounds have participated in the effort to shape the field over the last three decades. In the survey paper on Fault Tolerant Control (FTC) [71], the author has divided the research areas in the various fields pertaining to FTC into six major areas. These are:

- **FDI**: where the approaches vary from physical redundancy with voting schemes to model-based techniques and all efforts in this field contribute indirectly to Fault Tolerant Control (FTC). Currently, focus is on nonlinear FDI and modern techniques such as neural networks (see e.g. [13, 71]).
- **Robust Control**: Improvements in this field have contributed minimally to bringing in new design methods that produce reliable controllers in the presence of limited

type and number of faults. Examples of the work in this field are the ones reported in [52, 53, 74, 79, 83]. The activities in this area try to extend the concept of controller robustness to coincide with the concept of reliability.

- **Reconfiguration**: Controller reconfiguration is dominated by adaptive techniques that use feedback linearization or model-following. Another alternative is to gain schedule several controllers that are based on fault analysis.
- **Robustness**: Robustness issues related to controller reconfiguration especially the time taken to reconfigure controllers on-line are of prime importance. Few studies have addressed the design of robust reconfigurable controllers.
- **Integration**: The integration of robust controller and robust estimation is considered a vital area for future research where the fault-tolerance is dependent on the integration of both fields. An example of this approach is [65] plus some other examples cited in the references therein.
- **Supervision**: As indicated in the paper [71], very few researchers have shown interest in this area though it plays a central role in active[1] FTC.

1.3.1 Model-based vs. Model-free FDIA

Model-based techniques for FDIA systems are mainly based on linear or linearized models. There exists considerable research on nonlinear FDI but the achievement in this arena is not as mature as the linear FDI tools and algorithms. On the other hand model-free FDI methods are those based on physical redundancy with voting schemes, system identification, or neural networks. The model-based techniques require a high fidelity mathematical model in order to build an effective FDIA system. Lack of a proper model results in a poor FDIA that may not detect faults or may treat false alarms as faults [35]. Model-free FDIA systems that are based on physical redundancy may not be applicable in all situations due

[1] FTC is divided into *active* where controller reconfiguration takes place and *passive* where no controller reconfiguration is involved

to cost or space limitations. Model-free approaches that are based on system identification or neural networks have the advantage that they do not require a mathematical model which may not exist anyway. However, they do require laboratory data for building the FDIA system. Examples of research in this category include [2, 10, 44, 54, 66].

1.3.2 Fixed Controller FTFCS vs. Adaptive FTFCS

The reported work in the recent literature can be categorized into two groups. The first group of work considers fixed controllers integrated with some version of FDIA system. In this category, depending on the nature of the FDIA system, the controller will be reconfigured or left untouched. If the FDIA is used as an alarm system, the information conveyed by the FDIA is used to redesign the controller on line. If the FDIA is used to detect, isolate, and accommodate the faults then there is no reconfiguration/redesign of the controller as the faults are accommodated by the FDIA system.

The second category is totally different and in some cases the FDIA as classically described in the literature is not used at all. Instead, some model of the plant using system identification or neural networks are used to adjust the controller (see e.g. [9, 17, 85]). Other example of this type of FTFCS is what is proposed in Chapter 7, of the thesis where the fault characterization generated by the FDI system is used to adjust the adaptive controller.

1.4 Application of FTFCS Concept

The approaches taken to implement FTC in the domain of Flight Control System (FTFCS) as reported in the literature varry depending on the aircraft under consideration.

- **Fixed-wing Civil and Fighter Aircraft**: Such type of aircraft are typical candidates for FTFCS concept. The FTFCS can be designed to address sensor, actuator and airframe damage as well. However, due to the differences in the construction and mission of these aircraft different approaches are taken. For example, in civil aircraft it is possible to duplicate, triplicate or even quadruplicate the sensors. Of course there is an increase in the cost and space. It has been reported that there is an increase of 18% in the weight and 42% in the cost in the case of quadruplex redundancy of sensors [71]. This flexibility does not exit in fighter aircraft as the weight and space are quite limited. Thus, an alternative to physical redundancy is sought. On the actuator level, both types of aircraft are possible targets for consideration. Airframe damage is of prime importance to fighter aircraft. So it is not surprising that a lot of what have been reported in the field of FTFCS is directed to these types of aircraft. Examples of recent work for civil and fighter aircraft are given in [11, 12, 26, 28, 32, 48, 67].

- **Rotary-wing Aircraft**: Rotary-wing aircraft carry the characteristics of both fixed-wing and helicopters with the additional feature of high nonlinearity due to the switching of configurations from fixed-wing to helicopter or vice versa. Significant published work is available. An example of a paper dedicated to rotor-wing aircraft is [36].

- **Helicopters**: To the best of my knowledge, there is no published work on the application of FTFCS to helicopters. That may be due to the limitation of the configuration of the helicopter. However, it is possible to design an FTFCS for helicopters that addresses mainly the sensor failures and could extend that to include the degraded performance of the actuators and airframe damage for military helicopters. It is envisaged that the development of FTFCS for helicopters is no less important than others and the development of FTFCS for fixed-wing aircraft can benefit the FTFCS for helicopter. This can be observed via the growing interest in Unmanned Air Vehicles (UAV) that can be either fixed-wing or helicopter. Fault-tolerance for a UAV is of prime importance due to the absence of a human pilot.

1.5 Robust Control vs Fault-Tolerant Control

Robust control theory is centred around the robustness of the controller in the face of model uncertainty and disturbances. Extensive work has been conducted in this regard but none of the researchers has claimed that the design method is robust against faults except very few publications that have been mentioned previously and those few cases are limited to certain types and number of faults. On the other hand, FTC is making use of the advancements in robust control theory and integrating it with FDIA to construct a FTC system. From the discussion above and the partitioning of the research areas into six major fields, the relationship between robust control and FTC is clear.

1.6 Contribution and Thesis Organization

1.6.1 Thesis Contribution

The thesis has attained the following objectives:

- We have investigated the applicability of an ANN-based technique to the development of an FDI system for a Bell-205 Helicopter. Based on the analysis and the flight test data available at the Engineering Department Control Systems Lab of Leicester University, the design of an FDI system has been carried out and the results of desktop simulation are encouraging. The design is reported in Chapter 4.
- In order to design an FTFCS for the Bell-205, we have designed H_∞ controllers for the lateral and longitudinal dynamics. The design was carried out using a DERA (now QinetiQ) Linear Model around 20 knots. The time and frequency response analysis shows the fidelity of the design. The design was carried on using the H_∞ S/KS algorithm.
- We have integrated the above mentioned lateral and longitudinal controllers with the respective lateral and longitudinal FDIA systems and carried out various simulations and tests to assess the first FTFCS scheme centred around the concept of

a fixed structure FTFCS. Results with nonlinear FDI which were based on actual Flight Test Data (FTD) have highlighted the importance of accumulating rich FTD with sufficient documentation. To confirm the functionality of the FTFCS concept, an ANN-based FDI scheme using linear model data was developed and tested. The results were satisfactory and provided solid evidence of true functionality of the scheme though testing against the nonlinear model is still needed.

- Further, as per the recommendation of recent studies in the field of fault-tolerant control, the controller and the FDIA are better if designed together. Based on the various uncertainties concerning the presence of the FDIA in the feedback loop, μ-synthesis was used for the design of another FTFCS scheme that addresses the two systems together. Use of the μ-synthesis tool, has assisted in identifying the area for improvement of the overall system in terms of robustness and stability. Desktop simulation has shown again the functionality of the system but the accuracy needs to be checked with the full nonlinear model in place. This is reported in Chapter 6.

- In addition to the practical design work that was carried out in the thesis, a theoretical investigation was completed to study the impact of the faults on the Algebraic Riccati Equations (AREs) that are normally to be solved in finding stabilizing H_∞ controllers. Accordingly, a new FTFCS which is based on solving a new set of AREs is proposed. The solutions of the Riccati equations are used to update the controller that will compensate for single or multiple faults. The advantages of this approach have been highlighted.

1.6.2 Thesis Organization

The thesis is organized into eight chapters including this chapter and four appendices. A summary of each is given below:

- Chapter 1: This is the current chapter in which introductory and relevant general material is presented about the concept of FTFCS and the difference between this approach and the robust control methodology. In addition, a literature survey is presented in respect of the different parts of the thesis. The scope of the thesis is outlined.

- Chapter 2: This chapter includes a detailed presentation of the Bell-205 Helicopter, the plant on which the concepts of FTFCS are applied. The mathematical models are described though the nonlinear model was not available and, thus, it will not be presented.

- Chapter 3: Neural Networks have been used to design FDIA systems for the Bell-205. Thus, it is appropriate to bring into context the main and most important features of ANN. The architecture of ANN, its training, and its usage in control systems engineering is discussed. A detailed coverage of neuro modelling is given but neuro control is only considered briefly. However, the interested reader may consult some of the references in the literature survey or those cited in this chapter.

- Chapter 4: This chapter is about FDIA in general and later in the chapter details on ANN-based FDIA approaches are given. Based on the background given, the ANN-based FDIA for the Bell-205 Helicopter is presented with all of its fine details. The details in the chapter plus the supporting material in Appendix-A and Appendix-B give enough depth on the subject. The FDIA design basis, the ANN training, the design parameters, testing and validation of the neural networks used are presented. The pros and cons of using ANN are highlighted as well.

- Chapter 5: This chapter presents the concepts of the robust control paradigm and, then, uses it to design the lateral and longitudinal controllers. The fault-tolerance concept has been implemented in this chapter based on the integration of H_∞ controllers and the FDIA systems described in the previous chapter. The concept has been tested and the results are presented followed by some analysis of the robustness of the integrated system.

- Chapter 6: In the previous chapter, the H_∞ mixed-sensitivity controllers were designed based on the nominal linearized model model of the Bell-205 helicopter around a 20 *Knots* operating point. The design did not take into consideration the presence of the FDIA with its computational load and estimation discrepancies in case of failure occurrence. In this chapter, the design of the controller is based on the μ analysis and synthesis procedure. Additionally, the FDIA has been modelled and taken into consideration. The results and benefits of the approach have been analyzed in the framework of FTFCS.

- Chapter 7: In this chapter, analytic work is given which centres around the study of the impact of sensor faults on the AREs usually solved in finding a stabilizing controller. A new set of AREs are developed for the fault case and a new structured controller is designed. It is claimed, in view of its architecture, that it may be almost bumpless when a fault occurs. This constitutes a solid ground for further research.

- Chapter 8: This is a summary of the achieved results and some thoughts on future research.

- Appendix-A: This appendix describes the details of the architecture parameters of all ANNs designed for the FDIA system.

- Appendix-B: This appendix describes the simulation environment and the flight test data sets used in the project for the ANN training and validation.

- Appendix-C: This appendix gives a detailed overview of the flight test data structure. Some comments on flight test data have been mentioned earlier in the section on ANN training and during the discussion of the simulation results of Chapters 5 and 6.

- Appendix-D: This is a supplement to Chapter 7 where the rank of a solution to ARE is proved.

Chapter 2

Bell-205 Helicopter

The aim of this chapter is to give details on helicopter dynamics in general and the Bell-205 dynamics in particular. Nonlinear models, linear models, sensors, and actuators are covered.

2.1 Helicopter Dynamics

Helicopters are nonlinear, highly coupled, complex systems. The complexity of helicopters is attributed to various interacting subsystems. These subsystems include: the main rotor, tail rotor, powerplant, airframe, fin assembly, and empennage. Producing a nonlinear model requires the engineer to be aware of the dynamics and aerodynamics of various components and to integrate them together under plausible assumptions. The modelling of each component aims at producing the governing equations that relate the force and moment components to the subsystem parameters. The integration of components produces the nonlinear model [61, 64, 70, 73].The overall system is described by nonlinear differential equations as described hereafter in Section 2.1.1. The figure below (Figure 2.1) shows the various subsystems and axes of motion.

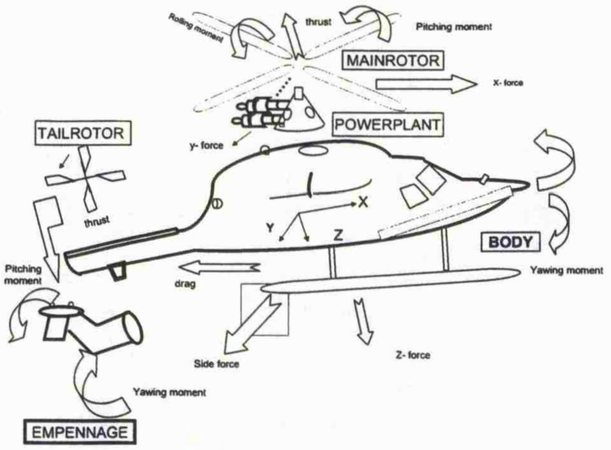


Figure 2.1: Helicopter Subsystems with Forces and Moments

Main Rotor (MR) Assembly [5, 80]

The MR has various structures. The most complicated one is the ***fully articulated*** rotor in which blades are free to move up and down. This type of movement is called *flapping*. The blades can twist about the span-wise axis in what is known as *feathering* and they can move back and forth in what is called *dragging*. This type of rotor is usually fitted with three blades equally spaced. Another type of MR is the **semi-rigid** one which is fitted with two blades that are rigidly attached to the MR mast. This arrangements allows the two blades to flap together where one flaps up and the other flaps down. This type of MR system uses the see-saw principles where one blade flaps up the other flaps down around a gimbal hinge sometimes called a *teetering* hinge. A third type of MR is the **rigid rotor** where the blades are free only to feather. Stability of this type is hard to achieve and, thus, it is equipped with a computer control to overcome the stability issue. From the above discussion, it is clear that each blade is a rotating wing with each blade having its own degrees of freedom. The extra degrees of motion in addition to the 6DOF fuselage complicate the modelling of the helicopter.

MR Blades

The blades' orientations are controlled via rods linked to the control system via mechanical linkages. Various controls can be exerted on the blades to affect the desired motion. For example, when the lever is moved up or down the blades' pitch angles are collectively increased or decreased accordingly which results in applying the MR thrust power to ascend or descend. If a combination of lever and longitudinal cyclic stick are applied, the extra lift power will be used to gain extra speed on descend or conserve the current speed and gain extra height by pitching up. In this combination, the pilot may need to increase or decrease the engine speed by manipulating the throttle to maintain constant revolutions per minute (RPM). Besides these basic motions, the helicopter can hover and move backward. Hovering, which is a fundamental feature of helicopter is a prerequisite for safe landing and aims at maintaining a given position to the ground at constant hight. It can take place at various flight conditions and the cushion it creates beneath the aircraft is influenced by the ground terrain. Under all these conditions there are various stability and performance issues that need to be analyzed carefully.

Tail Rotor (TR) Role

The MR normally moves left with respect to the fuselage and due to the asymmetry of the helicopter, the fuselage tends to move in the reverse direction . To overcome this problem, a single main rotor helicopter is fitted with a Tail Rotor (TR) to balance the MR torque by producing torque in the reverse direction and for that reason the TR is considered as an anti-torque system.

2.1.1 Nonlinear Model

The nonlinear model normally covers all the helicopter subsystems which include airframe, actuator dynamics, main rotor dynamics, empennage and tail rotor dynamics. In this section, we will derive the equations of motion in detail as we are going to use them later on when we come to linearization. However, the equations describing the dynamics of main rotor, tail rotor, and the empennage are only mentioned for completeness of the presentation without getting into the details of their derivation. Additionally, extra modelling features such as MR inflow will only be mentioned without getting into the details of deriving the governing equations.

The nonlinear differential equations that describe the model are of the form:

$$\frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}, \mathbf{U}, t) \qquad (2.1)$$

$$\mathbf{Y} = \mathbf{G}(\mathbf{X}, \mathbf{U}) \qquad (2.2)$$

where $\mathbf{X}$ is the state vector, $\mathbf{U}$ is the control input vector, $\mathbf{F}$ is a nonlinear vector-valued function of both variables, $\mathbf{Y}$ is the output vector while $\mathbf{G}$ is a vector-valued function mapping the states and control inputs to the outputs.

Equations of Motion

Under the following assumptions the nonlinear model is derived for the translational and rotational degrees of freedom:

- Assumption 1: The aircraft is assumed rigid while in reality it is not. This assumption implies that all points in the aircraft are assumed to maintain fixed relative positions in space at all time.
- Assumption 2: The mass of the aircraft is assumed constant throughout its flight. Of course, the mass of the aircraft varies due to fuel consumption in real flight.

Without the assumptions above, additional degrees of freedom must be added to account for flexible modes and variation of mass, which result in further complexity of the model.

The following equations are used to derive the nonlinear equations of motion (see e.g. [8, 21, 64]):

$$\sum \mathbf{F} = \frac{d}{dt}(m\mathbf{V}_T)|_I \quad \text{.... force summation} \qquad (2.3)$$

$$\sum \mathbf{M} = \frac{d\mathbf{H}}{dt}|_I \quad \text{.... moment summation} \qquad (2.4)$$

The first equation states that the summation of all forces (**F**) acting on the aircraft body equals the rate of change of the momentum of the body and the second equation states that the external momentum, **M**, of the aircraft equals the rate of change of moment of momentum (angular momentum), **H**.

The equations are re-written to indicate explicitly that the total forces and moments are those pertaining to steady-state and perturbations:

$$\begin{aligned} \sum \mathbf{F} &= \sum \mathbf{F}_0 + \sum \triangle\mathbf{F} \qquad (2.5) \\ \sum \mathbf{M} &= \sum \mathbf{M}_0 + \sum \triangle\mathbf{M} \qquad (2.6) \end{aligned}$$

In the last two equations, the first term is steady-state and the second is due to a perturbation. Initially the aircraft is assumed unaccelerated and the controls are locked which make the steady-state forces and moments equal to zero which lead to the following:

$$\begin{aligned} \sum \triangle\mathbf{F} &= \frac{d}{dt}(m\mathbf{V}_T) \qquad (2.7) \\ \sum \triangle\mathbf{M} &= \frac{d}{dt}\mathbf{H} \qquad (2.8) \end{aligned}$$

The rate of change of velocity vector, $\mathbf{V_T}$, can be expressed with respect to the earth using the following vector differential equation:

$$\frac{dV_T}{dt}|_E = \mathbf{1}_{VT}\frac{dV_T}{dt}|_B + \omega \times \mathbf{V}_T \qquad (2.9)$$

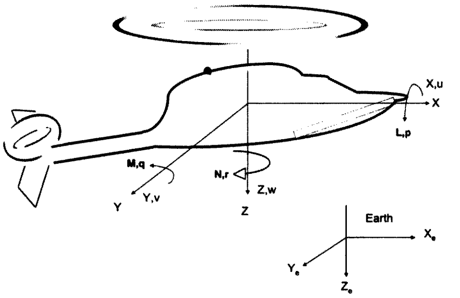


Figure 2.2: Coordinate Systems

where the subscript E indicates the Earth reference system and the subscript B refers to the coordinate system fixed to the aircraft body (see Figure 2.2). The first term in this equation (eq- 2.9) is the linear velocity resulting from a change in the length of the position vector while the second term is the tangential velocity of the vector due to the aircraft angular velocity. Using the following relations:

$$\mathbf{V_T} = iu + jv + kw \tag{2.10}$$

$$\omega = ip + jq + kr \tag{2.11}$$

$$\dot{\mathbf{V}}_\mathbf{T} = i\dot{u} + j\dot{v} + k\dot{w} \tag{2.12}$$

$$\triangle\mathbf{F} = i\triangle F_x + j\triangle F_y + k\triangle F_z \tag{2.13}$$

$$\omega \times \mathbf{V_T} = \begin{pmatrix} i & j & k \\ p & q & r \\ u & v & w \end{pmatrix} \tag{2.14}$$

where u, v and w are velocity components along the x, y, and z-axes respectively and ω is the angular velocity vector that has p, q, and r as its components around the same axes. By expanding the vector cross product $\omega \times \mathbf{V_T}$ in the last equation and collecting terms together, the following force equations are derived:

$$\begin{aligned} \triangle F_x &= m(\dot{u} + wq - vr) && (2.15) \\ \triangle F_y &= m(\dot{v} + ur - pw) && (2.16) \\ \triangle F_x &= m(\dot{w} + pv - uq) && (2.17) \end{aligned}$$

Following the same procedure, the rotational equations of motion are derived. The moment is equated to time rate of change of angular momentum, **H**:

$$\sum \triangle \mathbf{M} = \frac{d\mathbf{H}}{dt} \qquad (2.18)$$

where the moment of the partial mass dm due to angular velocity ω equals the tangential velocity times dm where the tangential velocity is defined as:

$$V_T = \omega \times \mathbf{r} \qquad (2.19)$$

Thus, partial moment is defined as:

$$d\mathbf{M} = (\omega \times \mathbf{r})\mathbf{dm} \qquad (2.20)$$

The moment of momentum (angular momentum), **H**, equals the moment lever times the moment:

$$d\mathbf{H} = \mathbf{rx}(\omega \times \mathbf{r})dm \qquad (2.21)$$

Recalling that the vector equations for the angular velocity and position vectors are given by:

$$\begin{aligned} \omega &= ip + jq + kr \\ \mathbf{r} &= ix + jy + kz \end{aligned}$$

the vector multiplication $\mathbf{r} \times (\omega \times \mathbf{r})$ yields:

$$\begin{aligned} \mathbf{r} \times (\omega \times \mathbf{r}) = &\, i[(y^2 + z^2)p - xyq - xzr] \\ &+ j[(z^2 + x^2)q - yzr - xyp] \\ &+ k[(x^2 + y^2)r - xzp - yzq] \end{aligned} \tag{2.22}$$

The angular momentum, **H**, is defined as:

$$\mathbf{H} = \int \mathbf{r} \times (\omega \times \mathbf{r})\, dm \tag{2.23}$$

where the integration is performed over the entire mass of the aircraft. By using the vector cross product of $\mathbf{r} \times (\omega \times \mathbf{r})$ given above we get:

$$\begin{aligned} \mathbf{H} = \int \mathbf{r} \times (\omega \times \mathbf{r}) dm = &\int i[(y^2 + z^2)p - xyq - xzr]\, dm \\ &+ \int j[(z^2 + x^2)q - yzr - xyp]\, dm \\ &+ \int k[(x^2 + y^2)r - xzp - yzq]\, dm \end{aligned} \tag{2.24}$$

and by decomposing it into the three components, we get:

$$\begin{aligned} H_x &= pI_{xx} - qI_{xy} - rI_{xz} & (2.25) \\ H_y &= -pI_{yx} - qI_{yy} - rI_{yz} & (2.26) \\ H_z &= -pI_{xz} - qI_{zy} + rI_{zz} & (2.27) \end{aligned}$$

where I_{xx}, I_{yy}, and I_{zz} are moments of inertia while the rest are products of inertia. Due to the symmetry of aircraft, products of inertia about the XZ-plane are zero which lead to the following simplified equations:

$$\begin{aligned} H_x &= pI_{xx} - rI_{xz} & (2.28) \\ H_y &= qI_{yy} & (2.29) \\ H_z &= rI_{zz} - pI_{xz} & (2.30) \end{aligned}$$

Using the above equations and $\frac{d}{dt}(\mathbf{H})|_E = \dot{\mathbf{H}} + \omega \times \mathbf{H}$, then the equations for the moments in decomposed form are:

$$\begin{aligned} \Delta M_x &= I_{xx}\dot{p} - I_{xz}(\dot{r} + pq) + qr(I_{zz} - I_{yy}) & (2.31) \\ \Delta M_y &= I_{yy}\dot{q} - I_{xz}(p^2 - r^2) + pr(I_{xx} - I_{zz}) & (2.32) \\ \Delta M_z &= I_{zz}\dot{r} - I_{xz}\dot{p} + pq(I_{yy} - I_{xx}) + I_{zz}qr & (2.33) \end{aligned}$$

To be in line with the notations conventionally used in many standard references, the following symbols will be used instead:

$$F_x \rightarrowtail X$$
$$\mathrm{F}_y \rightarrowtail Y$$

$$\begin{aligned} \mathrm{F}_z &\rightarrowtail Z \\ \mathrm{M}_x &\rightarrowtail L \\ \mathrm{M}_y &\rightarrowtail M \\ \mathrm{M}_z &\rightarrowtail N \end{aligned}$$

The force equations (Equations 1.15 - 1.17) and the moment equations (1.32 -1.34) constitute the rigid body dynamical equations. Of course, solving nonlinear differential equations is quite complicated and in some cases may not be possible. On the other hand, unlike numerical solutions, analytical solutions help to gain insights into the plant physics.

The variables for the states, inputs and outputs are defined in Table- 2.1 below:

Table 2.1: States, Input, and Output Symbols

Symbol	Meaning
u	Translational velocity along x-axis
p	Roll Rate
v	Translational velocity along y-axis
q	Pitch Rate
w	Translational velocity along z-axis
r	Yaw Rate
X	Gravity force along x-axis
L	Rolling Moment
Y	Gravity force along y-axis
M	Pitching Moment
Z	Gravity force along z-axis
N	Yawing Moment

Main Rotor (MR) Modelling

Modelling of the MR is normally done on three levels of complexity. The Levels are differentiated by the degree of the physics considered which is represented by the involvement of the aerodynamics and dynamics equations which vary from linear 2-D to nonlinear 3-D in terms of aerodynamics. On the level of dynamics, the mathematical representations vary from 3DOF/6DOF plus consideration of flapping and lag to consideration of elastic modes and detailed structural representations. The application of these levels is quite specific. Level-1 is targeted for low bandwidth control while Level-2 is for medium bandwidth and appropriate to high gain active flight control. Level-3 is the most complex one and is targeted for rotor design and analysis, vibration analysis and MR stability analysis. With the aid of the definitions cited in the symbol table at the beginning of the thesis, the equations for forces with respect to hub-wind frame (X_{hw}, Y_{hw} and Z_{hw}) and moments (L_{HQ}, M_{HQ} and N_h) are [70]:

$$\begin{aligned}
X_{hw} &= \Sigma_{i=1}^{N_b} \int_0^R \{-(f_Z - ma_{zb})_i \beta_i cos\psi_i - (f_y - ma_{yb})_i sin\psi_i \\
&+ ma_{xb} cos\psi_i\}\, dr_b \qquad (2.34) \\
Y_{hw} &= \Sigma_{i=1}^{N_b} \int_0^R \{-(f_Z - ma_{zb})_i \beta_i sin\psi_i - (f_y - ma_{yb})_i cos\psi_i \\
&+ ma_{xb} sin\psi_i\}\, dr_b \qquad (2.35) \\
Z_{hw} &= \Sigma_{i=1}^{N_b} \int_0^R \{-(f_Z - ma_{zb} + m_{xb})_i \beta_i)_i\, dr_b \qquad (2.36) \\
L_{HQ} &= -\frac{Q_R}{2}\beta_{1c} \qquad (2.37) \\
M_{HQ} &= \frac{Q_R}{2}\beta 1s \qquad (2.38) \\
N_h &= \Sigma_{i=1}^{N_b} \int_0^R \{r_b(d - l\phi) + I_R\dot{\Omega}\} \qquad (2.39)
\end{aligned}$$

Tail Rotor (TR) Modelling

The TR is modelled in the same way as the MR. However, the behaviour at low speed is different from high speed. The wake of the MR and the disturbed air from the MR hub creates local inflow that has to be taken care of especially for high fidelity models. The equations for forces and moments are:

$$
\begin{aligned}
X_T &\approx T_T\beta_{1cT} && (2.40a)\\
Y_T &= T_T && (2.40b)\\
Z_T &= -T_T\beta_{1sT} && (2.40c)\\
L_T &= h_T Y_T && (2.40d)\\
M_T &= (l_T + x_{cg})Z_T - Q_T && (2.40e)\\
N_T &= -(l_T + x_{cg})Y_T && (2.40f)
\end{aligned}
$$

where:

X_T, Y_T and Z_T	: Forces due to TR along x,y and z-directions.
L_T, M_T, and N_T	: Moments due to TR around x,y, and z-directions.
lc_T	: TR Lateral Cyclic
Q_T	: TR torque.
T_T	: TR Thrust.

Fin and Empennage

The flow around the fuselage and the empennage is quite complex and the forces and moments due to surface pressure and skin friction are functions of speed and direction. The force and moment equations are [70]:

$$
\begin{aligned}
X_f &= \frac{1}{2}\rho V_f^2 S_p C_{xf}(\alpha_f, \beta_f) && (2.41a)\\
Z_f &= \frac{1}{2}\rho V_f^2 S_p C_{zf}(\alpha_f, \beta_f) && (2.41b)\\
M_f &= \frac{1}{2}\rho V_f^2 S_p l_f C_{mf}(\alpha_f, \beta_f) && (2.41c)\\
Y_f &= \frac{1}{2}\rho V_f^2 S_s C_{yf}(\alpha_f, \beta_f) && (2.41d)\\
L_f &= \frac{1}{2}\rho V_f^2 S_s l_f C_{lf}(\alpha_f, \beta_f) && (2.41e)\\
N_f &= \frac{1}{2}\rho V_f^2 S_s l_f C_{nf}(\alpha_f, \beta_f) && (2.41f)
\end{aligned}
$$

where:

X_f, Y_f, and Z_f: Forces due to fin along the x,y, and z-directions.

M_f, L_f, and N_f: Moments around x,y, and z-directions.

ρ: air density.

S_s and S_P: Side and Planner area of helicopter.

C_{xx}: are force and moments coefficients usually stored in look-up tables.

Actuators and Sensors

The Bell-205 helicopter is equipped with a set of sensors to measure the various readings as indicated in Appendix-A. Clearly, no **redundancy** in sensors is available. The available sensors measure various attitude and rate quantities that include for example the positions, velocities, accelerations, pressure, height, and actuator positions ...etc. As far as controller design is concerned, the sensors are considered linear. Regarding the actuators, they are the standard set available in most helicopter that are linked to the cockpit inceptors via mechanical linkages. In control design they are treated as 2^{nd} order transfer functions.

The nonlinear model equations are obtained by grouping the force and momentum components contributed by the rigid body and the various components together.

2.1.2 Linearization of Nonlinear Model

Physical systems are mostly nonlinear and are represented by nonlinear differential equations and the resultant model is quite rich as it encapsulates most of the physical features of the plant . Despite the nonlinear model complexity, it may still fall short of the actual system. Solving nonlinear differential equations is usually not an easy task and may turn out to be impossible in some cases. Thus, in some engineering problems (e.g. controller design), nonlinear systems are approximated by linear models that are considered good enough to represent the system in the vicinity of some operating point. For that purpose, small*perturbation theory* and *Taylor's theorem* for analytic functions are used to linearize nonlinear systems.

The small perturbation theory states that an object in disturbed motion can be described as a perturbation from its trim. Mathematically, it is expressed as:

$$\mathrm{x} = \mathrm{x}_e + \delta \mathrm{x} \tag{2.42}$$

Taylor's theorem states that if a function and its derivatives are known at a certain point, then the behavior of the function can be estimated from an expansion of the function in a series about the known points. In mathematical terms, the expansion for the function $f(x)$ around $x = 0$ is given by:

$$f(x) = f(0) + f'(0)(x) + \frac{f''(0)}{2!}(x^2) + ... + \frac{f^{(n)}(0)}{n!}(x^n) + ... \tag{2.43}$$

As more terms are included, the estimation of the function will be more accurate. Expanding the function $f(x)$ using Taylor's series (i.e. approximating it around $x = 0$ and in the presence of a perturbation), we get:

$$f(x_0 + \Delta x) = f(x_0) + f'(x_0)(\Delta x) + \frac{f''(x_0)}{2!}(\Delta x^2) + ... + \frac{f^{(n)}(x_0)}{n!}(\Delta x^n) + ... \tag{2.44}$$

For example, the force along the x-axis is expressed as:

$$X = X_e + (\frac{\partial X}{\partial u}\delta u + \frac{\partial X}{\partial w}\delta w + ... + \frac{\partial X}{\partial \theta_0}\delta \theta_0) + ... \tag{2.45}$$

Ignoring all but the first order terms of the series expansion will leave only the linear terms which are dominant around a trim point. Further set points as e are such that the function values are zero; i.e. $X_e = 0$. Similarly, the rest of the force and moment equations are linearized in the same manner. Defining the following:

$\frac{\partial X}{\partial u} = X_u, \frac{\partial X}{\partial w} = X_w...$ etc.

which lead to a set of linear equations as will be presented in the following section. The set points around which the nonlinear system is linearized, are normally the ***points of equilibrium***. Though it sounds simple and straightforward to linearize nonlinear models, it must be used with caution as a linear model is only good around the equilibrium points with a small perturbation. As the perturbation becomes bigger, the linear model fails to represent the system.

2.2 Bell-205 Helicopter

The Bell-205 Helicopter is an advanced experimental fly-by-wire 3.629 tons helicopter. It has two pilot seats; the evaluation pilot (EP) on the right side can fly the helicopter with the assistance of an experimental Full-authority Fly-By-Wire (FBW), simplex and fail-safe system while the Safety Pilot (SP) on the left seat can override the control of EP through direct mechanical control of the helicopter. Further general detail (including main rotor, tail rotor, elevator, and vertical stabilizer) can be found in [14, 77]. The flight conditions cover many cases and are distinguished by the speed and trim parameters.

The helicopter has undergone some physical modification [1] which may indicate some discrepancy between the linear models available and the actual helicopter. This modification increased the usefulness of the Flight Test Data (FTD) as a reference of the available linear models [4, 76]. The models available for the unmodified helicopter are either produced by the Defence Research and Evaluation Agency (DERA, now QinetiQ) and it is called hereafter DERA Model or it is the general model produced by NASA, and hereafter will be referred to as NASA Model.These models will be described in succeeding section.

The helicopter is controlled via a collective stick, a cyclic (2-axis) lever and foot pedals. The helicopter is highly coupled and a command in one channel affects other channels as well. However, with this in mind one can describe the input/output relationship as per the table below, Table- 2.2

[1] The stablizing bar has been removed

Table 2.2: Input/Output Pairing Relationship

Pilot/Controller Command	Helicopter Response
Main Rotor Collective via stick	Heave velocity
Longitudinal cyclic via stick	Pitch attitude
Lateral Cyclic via stick	Roll attitude
Tail Rotor Collective via pedal	Heading rate

The sensors' minimum and maximum readings are as follows:

Table 2.3: Sensors's Minimum and Maximum Values

Sensor	**Symbol**	**min**	**max**
Pitch Rate	q	-7.65386	8.123258
Pitch Attitude	θ	-5.01567	16.9034
Roll Attitude	ϕ	-14.6088	10.62586
Yaw Rate	r	-11.6803	6.993779
Roll Rate	p	-13.9777	15.53333

The state-space model for the Bell-205 is represented by the following set of equations [76]:

$$
\begin{array}{llcll}
X: & \dot{u} & = & X_u u + X_w w + X_q q + X_v v + X_p p + X_r r & \\
 & & + & gcos(\theta_0)sin(\phi_0) & (2.46a) \\
Y: & \dot{v} & = & Y_u u + Y_w w + Y_q q + Y_v v + Y_p p + Y_r r & \\
 & & + & gcos(\phi_0)sin(\theta_0) & (2.46b) \\
Z: & \dot{w} & = & Z_u u + Z_w w + Z_q q + Z_v v + Z_p p + Z_r r & \\
 & & + & gcos(\theta_0)cos(\phi_0) & (2.46c) \\
M: & \dot{q} & = & M_u u + M_w w + M_q q + M_v v + M_p p + M_r r & (2.46d) \\
N: & \dot{r} & = & N_u u + N_w w + N_q q + N_v v + N_p p + N_r r & (2.46e) \\
L: & \dot{p} & = & L_u u + L_w w + L_q q + L_v v + L_p p + L_r r & (2.46f) \\
 & \dot{\theta} & = & q & (2.46g) \\
 & \dot{\phi} & = & p & (2.46h)
\end{array}
$$

where the state vector (x), the input vector (u), and the output (y) are:

$$x = [\ u \quad v \quad w \quad q \quad r \quad p \quad q \quad \phi\ ]^T$$

$$u = [\ \delta_{ls} \quad \delta_{lc} \quad \delta_{tr}\ ]^T$$

$$y = [\ \phi \quad \theta \quad r \quad p \quad q\ ]^T$$

For the Bell-205 the model has been produced by incorporating flight test data into the HELISIM [2] model.

2.2.1 NASA Model

The NASA Models are a collection of linear models obtained around several operating points in the flight envelope and documented in a contract report [38]. The trim angles are :

[2] HELISIM stands for Helicopter Simulation model built around Padfield flight mechanic model

Table 2.4: The NASA Linear Model trim angles

ϕ:	The Roll attitude
θ:	The Pitch attitude
α:	The Angle of Attack (AoA)
β:	The Side Slip Angle
γ:	The Heading angle attitude

The trim velocities are:

Table 2.5: NASA Linear Model trim velocities

U_o	velocity x-direction
V_o	velocity y-direction
W_o	velocity z-direction

In these models, no engine dynamics modelling is considered. Rotor speed is assumed constant and any rotor torques are instantaneous. The models have been evaluated for control law analysis and design and the findings reflect some shortcomings when validated in time and frequency domain as shown in [14, 77]. These models can be described by the **linear** matrix equations:

$$x(k+1) = Ax(k) + Bu(k) \tag{2.47a}$$

$$y(k) = Cx(k) + Du(k) \tag{2.47b}$$

where A, B, C, and D are the stability derivatives, control derivatives, and output matrices respectively which are very well defined for many operating points in the flight envelope.

The inputs are the lateral cyclic (FDA), longitudinal cyclic (FDE), and tail rotor collective (FDR). The Main Rotor (MR) collective is not part of the flight control system. The states are: the roll attitude (ϕ), pitch attitude (θ), roll rate (p), yaw rate (r), longitudinal velocity (u), lateral velocity (v), and vertical velocity (w). The outputs are: roll attitude (ϕ), pitch attitude (θ), yaw rate (r), roll rate (p), and pitch rate (q).

The set of linear models is documented in detail in [38]. There are sixty two flight conditions that cover the full flight envelope. The model is of 6^{th} order and the trimming is done at sea level for all flight conditions except for two of them which are meant to cover operation at altitude.

2.2.2 DERA Model

DERA (now QinetiQ) has produced a nonlinear model which was originally implemented in the TSIM environment of the dynamic equations of motion using Padfield's theoretical model of helicopter flight mechanics. The nonlinear model is built using the mathematical formulation mentioned in Section 2.1.1 where all the subsystems are modelled and the equations are presented to SIMULINK as polynomials and a set of look-up tables. The forces and moments from all subsystems are accumulated and used to represent the total forces and moments acting on the helicopter. The data on which the DERA model was based was taken from a contractor report prepared for NASA [38].The stability and control derivatives were originally provided by the manufacturer.

Afterwards, the model is used to generate a set of linearized models which are used to develop linear controllers. Another usage of the nonlinear model is for testing controllers prior to piloted flight test.

The DERA nonlinear model considers the Main Rotor (MR), the Tail Rotor (TR), the fuselage, and the fin. The MR is modelled using centre- spring-disk approximation to the rotor states with quasi-steady flapping and inflow dynamics [70]. The fuselage is a one-dimensional look-up table and polynomial function where incidence on sideslip angle is an independent variable. The DERA nonlinear model is described by a set of nonlinear discrete time equations of the form:

$$\mathbf{x}(k+1) = \phi(\mathbf{x}(k), \mathbf{u}(k)) \tag{2.48a}$$

$$\mathbf{y}(k+1) = \psi(\mathbf{x}(k)) \tag{2.48b}$$

where ϕ and ψ are nonlinear functions. $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{y}$ are the state, input and output vectors of the helicopter.

In addition to the nonlinear model, DERA has produced 8DOF and 12DOF linear models. The states, inputs and outputs are as seen in Table- 2.6 and Table- 2.7. DERA linear models are available in two versions; i.e. coupled and decoupled forms [3]. The decoupled models can be used separately for the design and evaluation of lateral and longitudinal controllers and the coupled model is used for the testing and evaluation of the combined controllers as has been done, for example, in [76, 77]. We have followed almost the same approach in our design of a fault-tolerant flight control system for the Bell-205 helicopter. The states, inputs, and outputs are grouped into lateral and longitudinal. The input for the longitudinal channel is longitudinal cyclic (δ_{ls}) while the states and outputs are given by Table 2.6:

Table 2.6: Longitudinal Sates, Outputs, and Inputs

	Symbol	Meaning
States	θ	pitch attitude
	q	pitch rate
	u	longitudinal velocity
	w	vertical velocity
	$X_{\delta ls}$	Longitudinal cyclic actuator
Outputs	θ	pitch attitude
	q	pitch rate
Input	δ_{ls}	longitudinal cyclic actuator input

The inputs for the lateral channel are: lateral cyclic (δ_{lc}), and tail rotor collective (δ_{tr}).The states and outputs are as shown below:

[3] Dynamic motion equations are originally coupled which represent the actual interaction between the various axes. For the sake of the design, they can be decoupled; i.e. separated into lateral and longitudinal dynamics (see e.g. [64, 70])

Table 2.7: Lateral States, Outputs, and Inputs

	Symbol	Meaning
States	ϕ	roll attitude
	p	roll rate
	r	yaw rate
	v	lateral velocity
	w	lateral velocity z-direction
	$X_{\delta lc}$	Lateral cyclic actuator
	$X_{\delta tr}$	Tail rotor actuator
Outputs	ϕ	roll attitude
	p	roll rate
	r	yaw rate
Inputs	δ_{lc}	lateral cyclic actuator input
	δ_{tr}	lateral tail rotor actuator input

As a result of a comparative evaluation with the above mentioned NASA models, it has been found that the DERA Models are of superior quality and the latter model(i.e. the 12DOF) has been recommended for use in control law analysis and future control laws synthesis [14]. The model was analyzed extensively and certain improvements have been recommended as indicated in [14]. One of the main points that has been highlighted in [77] is pertaining to poor indication of off-axis coupling which is extremely important for helicopter controller design. More details are given in the next section.

2.3 Model Validation

DERA linear models (8DOF and 12DOF) and the NASA 6DOF models have been compared against Flight Test Data (FTD). The FTD has been preprocessed by cropping some records to isolate the short term response to the control input which is approximately (3-4 seconds in the case of time-domain analysis and 20-40 seconds for frequency analysis). Also, the data was zero mean detrended to remove any trim offset and the inputs were linear detrended to remove the effects of drift caused by change of flight conditions. The comparisons were made in the time-domain and frequency domain. The flight data are used to excite both models and given below are the major conclusions [14]:

- The DERA 6DOF and NASA 6DOF models roughly capture the salient rigid body modes of the Bell-205 helicopter in the frequency range 1-10 rad/s. It is been suggested that additional engine modelling be made to improve the model fidelity.
- Fidelity of the DERA model can be further enhanced by considering the 12DOF model which incorporates conning and flapping dynamics and by including an inflow correction and tail fin blockage.
- With the improvements of the 12DOF model, there is still some uncertainty that shows significant discrepancy between the model and the aircraft. Uncertainty varies from one channel to another and can reach 70% to 80%.
- Predicting on-axis response was good overall with some minor exceptions.
- Though the pilot subjective assessments show that the 6DOF simulation was representative of the Bell-205 helicopter it has been recommended by Leicester University to use the 12DOF model for analysis of existing control laws and for the synthesis of future control laws.

2.3.1 Flight Test Data - FTD

Many controllers have been designed for the Bell-205 helicopter over the past five years. The controllers, after desktop simulation, have been coded and loaded into the Bell-205 helicopter on-board simulator and flight tested. During flight test, various data entities have been recorded for later analysis. The data collected can be grouped into:

- Sensor Measurements

- Actuator readings
- Dynamics related data: pressure, hight, true air speed etc.
- Safety readings
- Controller inputs
- Small perturbation readings for decoupling
- Miscellaneous.

Two main comments need to be made on the FTD. These are:

- The FTD was collected solely for controller performance and stability analysis and, thus, the data was collected during certain manoeuvres.
- The on-board simulator has a limited storage and, thus, the FTD are limited in size and coverage is not wide.

The above comments are very important when thinking about deploying an ANN-based FDI system into a real-time platform when the ANN-based FDI is designed using this FTD. Further details on the FTD are given in Appendix-A.

2.4 Conclusion

We have covered helicopter dynamics in general and the Bell-205 helicopter in particular. The background developed in this chapter will be used later on in Chapters 5 and 6 during the design of the controllers. The development of the nonlinear model is quite involved and the linear models are only approximations of the actual plant around specific operating points. As will be seen in Chapters 3 and 4, developing nonlinear models using ANNs is less expensive and more attainable provided that the flight test data is rich enough.

Chapter 3

Artificial Neural Networks (ANNs)

In our research project we used ANNs to build FDIA systems for the Bell-205 helicopter. Thus, in this chapter, it is intended to give necessary details that cover the mathematical background of ANN and its application in control systems design. Due to its biological origin, terms such neuron and learning are used while the term synapse has been replaced by weight which reflects its realistic role but in a mathematical sense. The chapter will give details on these architectural aspects as well as on learning capability.

3.1 ANN Architecture

Basically, Artificial Neural Networks (ANNs) are either of feed-forward or of recurrent structure. Multi-layer Feed-forward ANN (MFNN), which is equivalently called Multi Layer Perceptron (MLP), is the most widely used ANN architecture in control systems applications. However, recurrent neural network (RNN) is claimed to be more superior to MFNN due to its inherent feedback property [57]. These two architectural schemes are the major ones. However, the field is expanding rapidly with other paradigms, which are variations of the two or have some new features, emerging.

3.1.1 Structure of a Single Neuron

ANNs are flexible numerical structures; i.e. they have adjustable parameters. These parameters are the weights that interconnect two nodes (neurons)and the bias which is an additional variable to each neuron. The structure of the neuron is as seen in Figure 3.1 while the function that governs the behaviour of a single neuron is represented by the following:

$$\begin{aligned} a_j &= f(\sum_i x_i * w_{ij} + b_j) \\ &= f(v) \end{aligned} \tag{3.1}$$

Figure 3.1: A Single Neuron Diagram showing inputs, weights, bias and mapping function

where x_i is the input, w_{ij} is the weight, b_j is the bias and f is the mapping function that can be either linear or nonlinear.The mathematical foundations of ANNs and their connection to modelling will be further explored in the function approximation section later on in this chapter.

When an input is presented to a neuron, the output is produced pending on the current weight, bias, and mapping function.

3.1.2 MLP Structure

Multi Layer Perceptron (MLP) comes in a variety of structures and has the neuron as its basic building block. Mainly, it consists of an input layer, one or more hidden layers, and an output layer. The input and hidden layers have as many neurons as required by the application at hand. The number of neurons in the output layer is dictated by the actual number of the outputs of the physical system. The diagram below (Figure 3.2) depicts the overall structure of a fully connected MLP of two inputs, three outputs and one hidden layer.

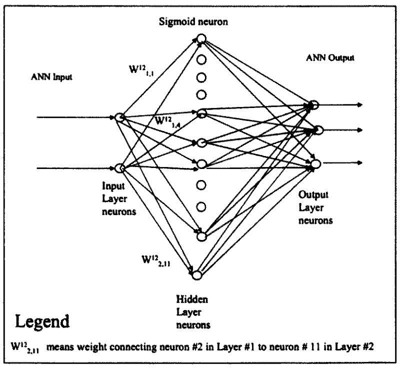


Figure 3.2: MLP Architecture

The MLP is suitable for static modelling and with appropriate time-delay units it can handle complex dynamical systems. The governing mathematical equation for an MLP (without time-delay) is given by:

$$y = f^n(W_{n,n-1}f^{n-1}(W_{n-1,n-2}...f^1(W_{1,0}x_i + b_1) + b_2)... + b_j) \qquad (3.2)$$

where :

$W_{i,j}$ represents the weight vector connecting neurons of two consecutive layers ,

b_j Column vector of bias at ith layer,

f^n Column of Mapping functions

x is the input vector

y is the output vector

Conventionally, for the first and second (first hidden) layers, each has a nonlinear mapping function and the last layer has a linear mapping function. If MLP is to be nonlinear at least one of its layers must have a nonlinear mapping function.

3.1.3 Recurrent Neural Network (RNN)

The RNN is similar to the MLP but it has an extra feature due to its dynamic feedback where some or all of the outputs are fed back to other neurons in other preceeding layers. Accordingly, the MLP with Time-Delay units in its inputs or/and outputs may also be considered as an RNN.

The mathematical representation of a RNN is the same as for the MLP but with additional terms to account for the feedback which could vary from one RNN to another depending on the architecture chosen for the problem. The diagram in Figure 3.3 shows an example of a RNN. Recurrent neural networks can be fully connected which is known as FRNN. In contrast to FRNN, partially connected RNN (PRNN) may have some of the feedback connections broken.

The mathematical equation of this RNN (Figure 3.3) is give by the following equation:

$$y = \psi[W_a^1(k) \circ X_a(k) + W_a^2(k) \circ Y_a(k-1)] \tag{3.3}$$

where:

ψ is the nonlinear activation operator (function)

$X_a(k)$ is the augmented neural input vector

$Y_a(k-1)$ is the augmented forward synaptic weight

$W_a^2(k)$ is the feedback augmented weight matrix

$\circ$ is the dot product

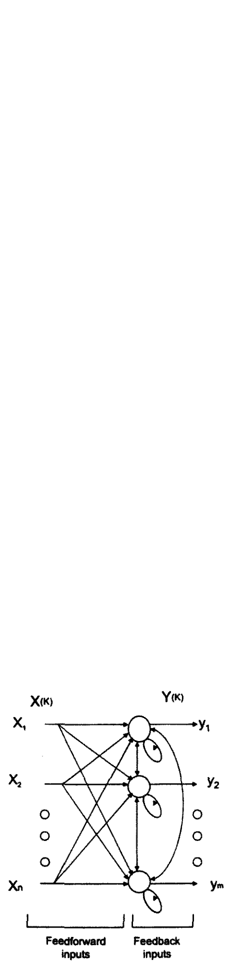


Figure 3.3: RNN Architecture

3.2 ANN Training

As we have seen the ANNs are characterized by their architectural components; namely, the weights and biases which are flexible to have any value. In order for the network to produce certain outputs, it must learn the relationship between the patterns of inputs and of outputs. The learning process is also known as training which can be conducted in two forms. The first and the most popular one is off-line training while the other is on-line training. In off-line training, the designed network is presented with a data set that contains pairs of inputs and outputs. The whole set is processed by the neural network and at the end, the ANN weights and biases are updated. A certain error function is evaluated to check that the ANN outputs are close enough to the desired outputs. If so, the training is stopped or else it continues till a better match is reached. The mechanism via which the training is controlled is known as the training algorithm. Back-propagation is the popular and de facto standard for ANN training. The mathematical basis of this algorithm will be explained later. In contrast to off-line training is on-line training which has fundamental differences. In on-line training, the weights and biases are updated upon processing of a single pair (alternatively called pattern) of data and, thus, it is sometimes called incremental training [15, 37, 56]. Normally, on-line training is used to improve the fidelity of previously produced ANN-based models or neuro controllers that were off-line trained. An example of this, is a research aircraft ANN-based model (or neural model)

[20, 46, 47] which was trained during typical flights to fine tune the model. Similarly, an aircraft neuro controller can be fine tuned during typical flight conditions to enhance the performance of the controller.

Backpropagation (BPTraining Algorithm)

The BP algorithm is so far the most popular algorithm which is based on the gradient descent method. It has many variations that aim to satisfy the conflicting requirements of boosting the speed of convergence while reducing the memory consumption as much as possible. Many references on this subject give enough details of this algorithm and its variations (e.g. [37, 41, 56, 57, 63]).

The algorithm is described in many references (e.g. [57]) and is given by the following equations:

$$w_{k+1} = w_k + \Delta_{wk} \tag{3.4}$$

$$\begin{aligned} \Delta_{wk} &= -\eta g_k \\ &= -\eta \frac{\partial E}{\partial w} \end{aligned} \tag{3.5}$$

where w_k is the weight in the k^{th} iteration and Δ_{wk} is the change in the weight while g_k is the gradient and η is the step size (sometimes called learning rate). The gradient is calculated by finding the partial derivative of the sum of the instantaneous errors at a specific neuron j.

The BP algorithm works in two phases or passes as follows:

Forward Phase: In this phase the input is presented to the input layer nodes and the result is calculated and propagated to the hidden layer nodes which in turn do the same and propagate the output to the next hidden layer or to the output layer. The output layer processes the results linearly and presents it. This process is described by the equations below.

Backward Phase: This starts at the output layer and moves backwards; i.e., back propagates in the layers successively to adjust the weights and biases accordingly.

This two-phase processing is repeated until the cost function is minimized. The following diagram (Figure 3.4 taken from [57]) shows the dynamics of the BP algorithm where the solid line represents the forward pass and the dotted line represents the backward pass.

Recalling the structure of the neuron (Figure 3.1), the internal activity of the neurons is given by:

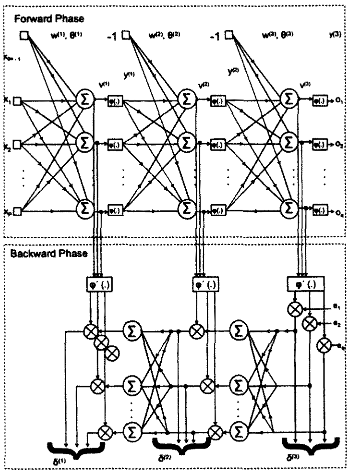


Figure 3.4: BP Algorithm Diagram

$$v_j(n) = \sum_{i=0}^{p} w_{ji}(n)x_i(n) \tag{3.6}$$

and the output of the neuron is given by:

$$y_j(n) = \varphi(v_j(n)) \tag{3.7}$$

Now using the chain rule and applying it to the gradient introduced in Equation 3.5, we get:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{3.8}$$

Now substituting the following relationship in Equation 3.8:

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \tag{3.9}$$

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \tag{3.10}$$

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \dot{\varphi}_j e_j(n) \tag{3.11}$$

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_j(n) \tag{3.12}$$

Equation 3.7 becomes:

$$\frac{\partial E(n)}{\partial e_j(n)} = -e_j(n)\dot{\varphi}_j e_j(n) y_j(n) \tag{3.13}$$

Recalling the correction in the weight given by Equation 3.5 and substituting Equation 3.13 in it gives us the equation for weight updating:

$$\Delta_{wk} = -\eta \delta_j(n) y_i(n) \tag{3.14}$$

where δ_j is called the local gradient and for the hidden layer is given by:

$$\delta_j(n) = -e_j(n)\frac{\partial E(n)}{\partial y_j(n)}\dot{\varphi}_j e_j(n) \tag{3.15}$$

For the output neuron, the local gradient is given by:

$$\delta_j(n) = e_j(n)\dot{\varphi}_j e_j(n) \tag{3.16}$$

3.3 Function Approximation

Physical systems are nonlinear and normally described by a set of nonlinear differential equations. Solving these equations may in some situations become very difficult and, thus, numerical approximation helps to solve them. Based on Approximation Theory, any smooth nonlinear function (i.e. a nonlinear function that is continuously differentiable everywhere) can be approximated by an MLP with a single hidden layer that has sufficient number of neurons. As shown in [54, 57], an ANN with a single hidden layer can approximate any nonlinear function provided it has a sufficient number of neurons in its hidden layer. The theorem below shows the structural linkage between neural networks and function approximation.

Theorem 3.1 *[57] Let $\varphi(.)$ be a nonconstant and continuous function, Let I_p denote the p-dimensional unit hypercube $[0,1]^p$. The space of continuous functions on I_p is denoted by $C(I_p)$. Then, given any function $f \in C(I_p)$ and $\epsilon > 0$, there exist an integer M and set of real constants α_i, θ_i and w_{ij}, where $i = 1,..M$ and $j = 1...p$ such that for*

$$F(x_1,..,x_p) = \sum_{i=1}^{M} \alpha_i \varphi(\sum_{j=1}^{p} (w_{ij} x_j + \theta_i)) \tag{3.17}$$

as an approximate realization of the function $f(.)$ in the sense that,

$$|F(x_1,...,x_p) - f(x_1,...,x_p)| < \epsilon \textit{ for all } x_1,..,x_p \in I_p$$

Here, the function φ can for example be the logistic function $\frac{1}{[1+exp(-v)]}$ which is used to represent the *nonlinearity* in the neuron model. The function is a nonconstant and bounded function. Also the above equation (3.17) resembles the equation of an MLP where the network has p input nodes, a single hidden layer consisting of M neurons, a hidden layer with synaptic weights $w_1...w_p$ and biases θ, and the network output is a linear combination of the the outputs of the hidden neurons with $\alpha_1,...,\alpha_p$ defining the coefficients of this combination. A detailed proof of the above theorem can be found in many references (e.g. see [57]).

3.4 Application of ANN in Control Systems

As we have seen, the ANN with its mathematical structure is very much suitable for function approximation. Real systems are mostly nonlinear and are represented by vector-valued nonlinear functions. Though developing nonlinear models takes lots of effort and time, the resultant models either may not be accurate enough to capture all physical features of the nonlinear plant or may have equations which are too difficult to solve. In either case, the input/output recording of the system can help to build better and easier models. ANN is a mathematical tool that has a proven record in this regard. ANN can be used as a modelling tool and the models can be built without great involvement in the plant mathematics that describe the dynamics of the system. Additionally, ANN can be used to build controllers for complex systems. These controllers are known in the literature as neurocontrollers. In this section we will give brief introduction to how ANNs are used in system modelling and later on in Chapter 4 we will describe in depth how ANNs are used in modelling, in general, and in building FDIA in particular. As we will not use ANN in building controllers in the thesis, no further details are presented on neurocontrollers.

3.4.1 Neuro Modelling

The basic background on which neuro modelling is based on is the ANN capability of approximating nonlinear functions. In Section 3.3,we saw how ANN with a single hidden layer is capable of approximating a nonlinear function with high degree of accuracy. As we recall from Chapter 2, in order to build a nonlinear model for a helicopter, in general, we need to derive all force and moment equations for all the subsystems. An alternative to that is to build neural models for the helicopter based on Flight Test Data (FTD). Of course, neural models may not give physical insight of the plant unlike the mathematical model. An example of a generic model for a complex system is, for example, building a neural model of lateral dynamics of an aircraft. This model can be built by selecting all, or a subset of, plant inputs and outputs and limiting the output to the physical quantities (namely the roll attitude, ϕ, the roll rate, p, and yaw the rate r). Similar models can be built for the longitudinal dynamics. However, an alternative approach is to build a more comprehensive model that consists of several neural networks, each neural network modelling a subsystem with one ANN. At the end, we may use all the other networks outputs in addition to some other inputs to generate various outputs. The two approaches are described by the following diagrams, Figure 3.5 and Figure 3.6. The factors affecting the accuracy of approximation are:

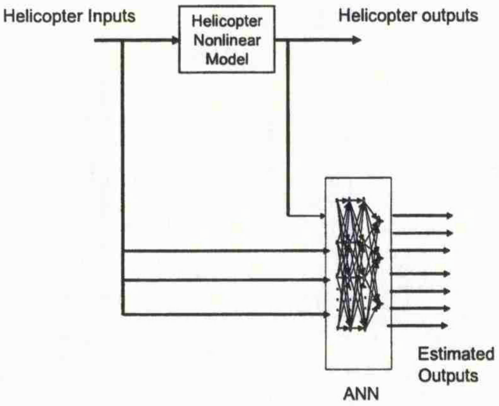


Figure 3.5: Generic Neural Model

- Size of hidden layer: the number of neurons, especially of the hidden layer.

- The number of hidden layers: Even though it has been stated that ANN with a single hidden layer can approximate any function with high level of accuracy, it has been found in some cases that adding more layers may improve the accuracy at the cost of computational overhead.

- Richness of the training data: Failure to collect sufficient data results in poor models or even models that fail to represent the plan in some operating conditions.

3.4.2 Neuro Modelling Guideline

The following guidelines may be found useful:

ANN Architecture Beside the fact that one has to decide whether to use MLP or RNN, the designer has to decide on the number of layers, neurons per layer, and the activation function for neurons at each layer. In regard to the size of the network, there are two approaches. Either to start with a small size of network and depending on performance, one increases the complexity of the network until optimal performance

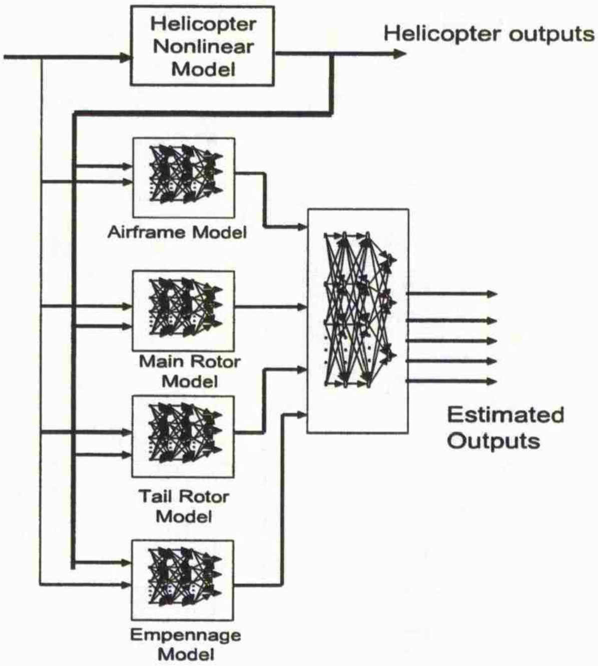


Figure 3.6: Comprehensive Neural Model

is reached. The other approach is technically more sound where one starts with a large size of network and by using either a genetic algorithm or pruning technique one gets rid of those nodes in the network that have minor contribution [57].

ANN input space One can blindly use the available data set and end up with either a poor performing network or an over-fitting network that results in poor performance once the network is used in real applications. Therefore, it is necessary to use some preprocessing techniques that normalize the training data or keep only the most contributive elements of the input vector [63]. Inputs in general can be chosen for a complex system such as the aircraft and its subsystems using physical insights.

3.5 Why use ANN in FDIA?

As has been reported widely in the literature, ANN can be used for either pattern classification or function approximation. It's been found that MFNN with single hidden layer of sufficient number of neurons can be used as a universal approximator for any smooth nonlinear function provided that it is trained using representative data sets. In some applications, adding more hidden layers help to speed the convergence to capture the features of the model. As indicated in the literature, FDI can be and has been implemented using for example the Luemberger observer, the Kalman filter, etc. In all these cases, a good model is required to detect robustly and isolate faults. The available mathematical models for the Bell-205 helicopter, especially the NASA and DERA 6DOF models, have certain shortcomings. ANN can produce good models provided high quality FTD are available. The accommodation is an extra task that can be implemented via ANN, fuzzy logic, or simple decision logic testing of instantaneous values [13, 35].

More importantly, all FDI methods that have proven records in tackling linear or linearized systems cannot generally be applied to nonlinear systems especially if they are ill-defined, complex, and/or have high degree of uncertainty in their models [13]. ANN has shown capability in handling such nonlinear mappings and, thus, has been used for nonlinear FDI systems as indicated in [13]. The field of nonlinear FDI is, unlike the linear FDI system, still under development while the latter has been well established since the late 80's and early 90's. ANN with its powerful architectural mapping and learning capabilities has shown strength to contribute to the development of nonlinear FDI. In some industrial applications, the powerful **parallelism** characteristic of ANN has been greatly utilized by using some neural chips or specialized processors that support parallelism such as the **TRANSPUTER** and its powerful language **OCAM**. This feature greatly enhances the ability of an ANN-based FTFCS.

3.6 Conclusion

We have in this chapter given a detailed overview of ANN architecture and training leaving out all variations but focusing on the architectures that will be used in our work. Many details, especially on training, are not included as it is considered out of scope but important remarks have been mentioned. We must stress that ANN does have its pros and cons. In what has been presented, we have seen the major pros of ANN techniques. However, the major shortcoming is pertaining to processing time in case of large and complex applications. However, it may not be a negative aspect of the ANN but rather, as recognized by many designers, that ANNs are parallel in nature which lends themselves to parallel machines. Thus, complex applications that result in large networks may take prolonged learning time. Learning time can be handled by special versions of the BP algorithm and use only off-line training. However, processing will be on-line and there may be some computational delay. One remedy is to use specialized hardware where there are varieties of specialized neural chips and boards that make the processing time well below normal expectation.

Chapter 4

Fault Detection, Isolation and Accommodation(FDIA)

Introduction

A system's tolerance to faults and operator errors is becoming of prime importance to ensure safety and reliability of plant operation. Faults can occur to plant sensors, actuators, and plant components. Occurrence of faults whether they are incipient or abrupt may lead to catastrophic results if not attended to in good time. In other cases, faults may result in major degradation of stability and/or performance of the plant. Thus, building systems that are fault-tolerant is of prime appeal in sophisticated industrial processes. Fault detection, isolation, and accommodation is the corner stone in the design of the fault-tolerant control systems.

At the beginning of the chapter we aim to give general coverage of the subject with emphasis on the ANN-based technique. The rest of the chapter is devoted to details of the design of ANN-based FDIAs for the Bell-205 Helicopter. The design details cover all the sensors of the lateral and longitudinal dynamics. Desktop simulation results and their analysis are presented.

4.1 FDIA Concepts

In the literature, the terminology in the FDI area is not fully unified. Thus, it is appropriate to start the chapter with definitions of the terms used in the thesis.

Fault Detection means the ability of the system to make a binary decision about the occurrence of a fault.

Fault Isolation means the ability of the system to identify the source and magnitude of the fault. Sometimes, determining the source of the fault is referred to as *isolation* while determining the magnitude of the fault is referred to as fault *identification*. In the thesis we used the former term for both cases.

Fault Accommodation means the system's ability to replace the faulty signal with an acceptable estimate.

Residual means the difference between a signal and its estimation. Fora a healthy plant, the residual vector should be zero. Otherwise, it indicates a fault occurrence if its magnitude diverges from zero by a prescribed range.

4.2 FDIA Design

There are basically two major approaches to the design of FDIA systems. These are:

- Physical Redundancy
- Analytic Redundancy [1]

In physical redundancy, the sensors, actuators, or plant components are duplicated or even triplicated. On the software level, a voting scheme is implemented to make decisions in cases of significant differences. For example, in an Airbus aircraft 300% redundancy is implemented in hardware. The same degree of redundancy is used in some parts of process control such as nuclear reactors. The benefits of such an approach can be justified in

[1] Analytic Redundancy (AR) exploits the implicit redundancy in the relationship between various measurements by using mathematical models

terms of increased safety and reliability of the system at hand. Of course, there is a significant increase in initial capital cost plus running cost for maintenance, and the extra space required to lay extra hardware. In some cases, such as with a combat aircraft where the space limitation is a major constraint or with plants with less hazards but where the cost minimization is a stringent requirement, other solutions have to be sought. Such plants can be diagnosed using analytic redundancy which use analytical or functional relationships among various plant variables to decide whether or not a fault has occurred.

Analytic redundancy requires mathematical models to start with. Since plants are mostly nonlinear this means that the FDIA is designed based on either nonlinear models or linearized models. Nonlinear FDIA, unlike linear FDI, is still an immature field. As analytic redundancy relies on mathematical models, they are sometimes called model-based. As there are actually some FDIA which are analytic but on the other hand do not require a mathematical model, we may consider the following grouping of analytic redundancy FDIA:

- Model-based approaches which include all methods that require mathematical models which are usually linear.
- Model-free approaches which include methods such as the ANN-based technique. Physical redundancy is model-free but is not analytic.

Model-based approaches have their own advantages but entail some cost. The cost is insignificant compared to the physical redundancy. Apart from development cost, model-based approaches require high fidelity mathematical models.

4.2.1 Model-based FDI

Model-based FDI requires a mathematical model for the system which is normally obtained by linearizing the nonlinear model around several operating points. As the model is going to be used for fault detection and isolation, the model has to be of high fidelity to avoid misleading detection and isolation information that may lead to wrong decisions.

On the one hand, the FDI system should be good enough to detect all type of faults whether they are abrupt or incipient and whether they have big or small magnitudes. On the other hand, the FDI system has to be robust enough to avoid false alarms. These conflicting requirements are made worse by significant mismatch of the model and the real plant. Many model-based FDI methods have been developed during the last three decades. Some of the methods deal with deterministic plants. Examples of these methods dealing with this type of plant are [13]:

- Observer-based FDI: The concept is to estimate the outputs of the system using an observer or a filter. A Luenberger observer is used for deterministic systems (see e.g. [13, 72]) while a Kalman filter is used for stochastic systems. For all dynamic systems there exists an observer-based residual generator because all inputs-outputs are observable [13].
- Parity Relation FDI: Here the idea is to check for consistency of system measurements .
- FDI via Parameter Estimation: The system parameters are estimated on-line using system identification techniques. The estimated parameters are compared against on-line measurements and once significant discrepancies take place fault occurrence is triggered.
- FDI using Statistical Techniques: The idea is based on statistical testing of residuals. Many methods have been reported such as Weighted Sum-Square Residual (WSSR) and Generalized Likelihood Ratio (GLR).
- Nonlinear Observer methods.

4.2.2 Model-free FDI

Model-free FDI requires no mathematical model. Examples of this type of FDI includes hardware redundancy (noting that they are not analytic) with voting scheme. Artificial

Neural Networks based technique is another kind of model-free FDI but with the following major advantages:

- ANN-based technique is actually a nonlinear modelling technique but does neither require a nonlinear model nor a set of linearized models to start with. All it requires are training data sets that are rich enough to represent the plant in its full set of operating conditions.
- If the training data is rich enough, then the ANN-based FDI built with it is more superior to FDI systems built using linear models. The first feature is pertaining to the fact that the FDI is based on the training data which is the closest representation of the actual physical plant. So discrepancies between the ANN model and the actual plant are less and subsequently false alarms are minimized. The second feature is pertaining to the fact that ANN-based FDI that are built with rich training data are actually nonlinear FDI systems. This means that switching between several FDI systems is not required.
- The cost of developing ANN-based redundancy and using it in fault detection and isolation is minimum compared to physical redundancy that requires initial investment in hardware and subsequent running costs for maintenance.
- There are various approaches according to which ANN is deployed in the FDI system. One of them, which we have used and will be explained later on in this chapter (Section 4.5), does not require faults' signatures to decide whether a fault has occurred or not. This is a useful feature as it does not require great amounts of knowledge of the system faults especially those faults which may not be anticipated but under certain conditions may occur.

4.3 Systems Identification Principles

As indicated in [29, 54], system identification of linear systems is very well established where several models are available for direct use. Fortunately, similar models can be extended to nonlinear systems. These nonlinear models can be grouped into the following four models [54]:

Model I: which relates the output to the linear combination of the past outputs and nonlinearly to the past inputs via the function f which which is assumed to be differentiable of its argument as per the following equation:

$$y_p(k+1) = \sum_{i=0}^{n-1} \alpha_i y_p(k-i) + f[u(k), u(k-1), ..., u(k-m+1)] \tag{4.1}$$

Model II: which clearly relates the output linearly to the past inputs and nonlinearly to the past outputs via the function f which is assumed to be differentiable of its argument as per the following equation:

$$y_p(k+1) = f[y_p(k), y_p(k-1), ..., y_p(k-n+1)] + \sum_{i=0}^{m-1} \beta_i u(k-i) \tag{4.2}$$

Model III: which relates the output nonlinearly to the past inputs and outputs via two separable functions f and g which are also assumed to be differentiable of their argument as per the following equation

$$\begin{aligned} y_p(k+1) &= f[y_p(k),, y_p(k-1), ..., y_p(k-n+1] + \\ &\quad g[u(k), u(k-1), ..., u(k-m+1)] \end{aligned} \tag{4.3}$$

Model IV: which relates the output to the past inputs and outputs nonlinearly via one function f.

$$\begin{aligned} y_p(k+1) &= f[y_p(k),, y_p(k-1), ..., y_p(k-n+1); u(k), \\ &\quad u(k-1), ..., u(k-m+1)] \end{aligned} \tag{4.4}$$

The last model is the most general one and the first three models are considered special cases of it. For a nonlinear system described by one of the above models, two approaches are generally used in the identification:

Parallel Identification Model: In this model (Figure 4.1), the input to the identifier is the plant input vector $u(k)$ and past output vector of the identifier itself $\hat{y}(k)$. The stability of this architecture is quite questionable and conditions under which it is stable are not known. The governing equation is:

$$\hat{y}(k+1) = f(u(k), u(k-1), ..., u(k-n), \hat{y}(k), \hat{y}(k-1), ..., \hat{y}(k-n)) \quad (4.5)$$

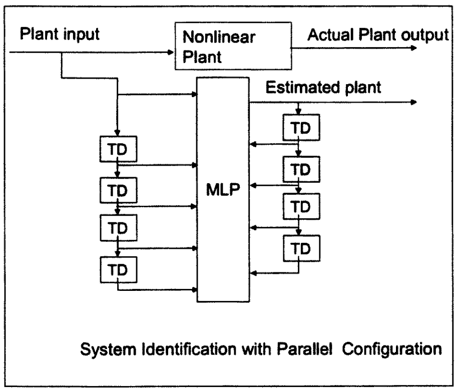


Figure 4.1: Parallel Identifier Model

Parallel-Series Model: Unlike the parallel model, the inputs to the identifier are the plant input vector $u(k)$ and the plant output vector $y(k)$. The model (Figure 4.2) is stable provided the modelled plant is Bounded Input Bounded Output (BIBO) stable. Here, the plant measurements (not the estimates) are used as input to the model.

The mathematical structure is described by the following equation:

$$y(k+1) = f(u(k), u(k-1), ..., u(k-n), f_s y(k), f_s y(k-1), ..., f_s y(k-n)) \quad (4.6)$$

where $f_s = 1$ if the sensor is healthy.

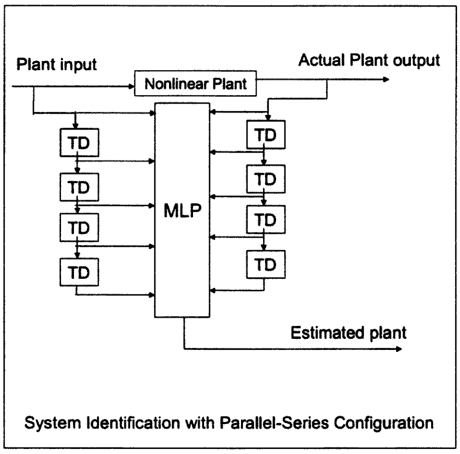


Figure 4.2: Parallel-Series Identifier Model

Both of the above models can be either handled by MLP or RNN where the mapping function f that maps the input to the output can be implemented via a neural network which has proved to be a good candidate for function approximation.

4.4 ANN-Based FDI

There are various approaches to designing an ANN-based FDI. The approaches may look similar to those used in linear FDI as far as the structure is concerned. However, ANN-based FDI is actually a nonlinear FDI system provided that the training data is quite representative of the operating conditions. This is a major difference. The structures we will discuss are:

- NN-based Residual Generator and Decision Logic.
- Two Stage NN-based FDI where one is used for FD and the other one for FI.
- One large ANN that combines FD and FI functions.
- Two Stage ANN-based FDI and a decision logic.

NN-based Residual Generator and Decision Logic: In this approach a neural network is constructed and trained to model the plant and generate estimates of all of its outputs or a subset of them depending on the scope of the diagnosis scope and generated residuals. The residuals are then processed by the decision logic to generate a fault indicating signal based on the residual values.

NN-based FD and NN-based FI This approach is similar to the previous one but rather than using a decision logic for the assessment of the fault, the output of the FD neural network is passed on to another neural network to act as a fault classifier. The neural classifier is trained to analyze the features of the incoming signals (outputs of the NN-based FD) and to make appropriate decisions.

One-Stage: Combined NN-based FD and NN-based FI functions This approach is similar to the previous one but rather than using two neural networks, one NN is used; i.e the two ANNs are combined together using a larger NN. This approach has been exemplified in [13].

Two Stage ANN and Decision Logic FDI In this approach two stages of ANN are used to model the plant outputs collectively via a single MIMO ANN and a set of ANNs that are used to model the individual plant outputs. This approach has been used in [31]. The main feature of this architecture is the reduction of false alarms.

We have used the last approach with two fundamental features [2]. The details will be given in Section 4.5. The reasons for selecting this scheme are:

- The architecture is excellent in minimizing false alarms as it has two thresholds one for the MIMO ANN and one for each individual MISO ANN. If we use only one neural network and based on that ANN threshold we make decisions then we either have to relax the threshold at the cost of increasing the risk of neglecting some faults or to tighten the threshold at the cost of treating many false alarms as faults. Neither case is desirable in robust FDI.

- The second scheme uses two ANNs one for FD and one for FI which means it has the same computational burden as the one we have selected but without the feature of the last scheme of eliminating the false alarms.

- The third scheme uses a larger ANN and normally when the ANN gets larger it takes lots of time to train and may not be satisfactory for an industrial-scale problem.

4.4.1 Fault Accommodation

In the last section, we have focused mainly on FDI as most of the reported work is concerned with FDI. However, the ultimate goal of FDI is to maintain the safe operation of the plant despite the various faults that may occur. There are several approaches that can be taken which will be summarized here and details will be given later on in this chapter and when describing the fault-tolerance flight control system in Chapter 5 and Chapter 6. The approaches in general are:

- Using a Decision Logic Module (DLM) where the output of which will be either the original signal of the plant if the checked signals are healthy or a combination of healthy signals and estimated signals if there is a single or multiple faults. The output is fed to the controller as the best representation of the system.

- Using Fuzzy Logic to decide on the feedback signals based on fault isolation results and some membership function. This is another format of the DLM but using the strict membership functions of fuzzy logic. Normally the human knowledge is coded which makes decisions more solid, [10, 13].
- The fault detection signal is used directly in an adaptive scheme to compensate for the faults by restructuring the controller on-line or by generating an adaptively compensating signal.

We have used the first scheme in our design of FTFCS as shown in Chapter 5 and Chapter 6. Also, we have also highlighted the usage of the last scheme in in the adaptive schemes developed in Chapter 7.

4.4.2 Neural Network Architecture for Fault Detection and Isolation (FDI)

The general Parallel-Series architecture is used to model the sensors and generate estimates of their outputs. The architecture of the neural network is not unique. That is to say; the topological features of the network in terms of the number of layers and number of neurons per layer could be different from one design to another. Optimum design of an ANN under the same circumstances can be obtained using the techniques mentioned in Chapter 2. Normally three layers are sufficient. However, it could be possible in some cases that an extra layer is needed as will be seen in the following section when designing e.g. the MIMO ANN (LANOF) for the lateral dynamics. It is necessary to mention that the extra layer was needed due to the nature of the flight test data used in the training. The flight test data (FTD) was recorded probably according to some practical characteristics. If the FTD was richer, the extra layer may not be needed. Consequently, the training would have been simpler. This conclusion is in line with the general theorem on ANN (shown in Chapter 2) where it has been that an ANN with a single hidden layer is sufficient to approximate any nonlinear function (plant). One may notice that the input for a MISO (say e.g. phiNN) is not a subset of its master ANN (MIMO ANN, LANOF). One may expect that the input to the former should be a subset of the later. This is not the case. The reason, in my opinion, is due to the nonlinear relationship and coupling of the various variables in the system. In addition, modelling the lateral channel collectively is more attainable with some subset of variables than for modelling a single sensor. This may be due to the fact that finding a nonlinear mapping of a single-output is harder than for the multi-output case. Empirical results confirm this phenomenon where training single output ANN sometimes take a longer time compared to multi-output ANN.

Another point that we want to clarify, is the benefits of the 2-stages approach. Clearly, a 2-stages scheme implies more computational cost which is justifiable in view of the benefits it possesses.

Before proceeding further, the term 2-stages as used in the literature will be first clarified. In some references (e.g. [13]), it is used to mean the presence of two ANNs where the first one is used to detect fault by comparing the plant output (y) and the ANN output ($\hat{y}$); i.e. the first ANN is used to generate what is called **residuals**. The second ANN is trained on patterns of these residuals to learn fault signatures. The 2-stages have been merged together and tested as shown in the above reference. However, the resultant ANN is bigger in size and there is no reporting on the comparison of performance during training and in operation. As a matter of fact, the problem exemplified by this approach has been illustrated by a simple dynamical system (the 3 tanks problem). It is anticipated that with a large dynamical nonlinear system, the two networks are relatively large in size and the effort paid to train them will be high. If the 2-stages are merged together,the size will be even bigger and consequently the training cost will be higher.

Another way of using 2-stages is as described in many references (e.g. [31]), where basically the first ANN is a MIMO ANN of the overall system and it does fault detection. The second ANN is a collection of MISO ANNs that each is producing estimates of the concerned variables for fault isolation. The isolation is handled by a decision logic. The fundamental feature of this hierarchial approach is reducing **false alarms**.

4.4.3 NN Input Space

Basically, a neural network is a learning system that correlates certain variables in a data set to another set of variables in the same data set provided a relation does exist between the two. The ANN is capable, via proper training, of extracting the features of the pairs and adjust its weights and biases to model the nonlinear relationship. The first set of variables need not be all the plant inputs and the other set need not to be all the plant outputs. To clarify, if we consider the ANN used for lateral dynamics (LANOF). In regard to its input space, we have selected the lateral cyclic (FDA), the pedal cyclic (FDR), the roll rate (PHI_DOT), acceleration in z-direction (AZ), the velocity component in x-direction (UDOT), and the velocity component in the z-direction (WDOT). The selection logic is a mix of prior knowledge of the impact of the variables on the output variables and trial-and-error. The trial-and-error has come into play when the ANN training started. By inspecting the learning performance curve, inclusion and/or exclusion of certain variables in addition to adding more layers or neurons can be made. The final version of the ANN is satisfactory as will be seen in the test and validation section at the end of this chapter (Section 4.6). As we can see the selection of the variables for lateral dynamics does not necessarily have to be all of the variables pertaining to lateral dynamics nor do they have to exclude all variables from the longitudinal dynamics. This certifies that the approach of splitting the FDI and consequently the FTFCS into lateral and longitudinal is not affecting our ANNs and, thus, the coupling of the dynamics is made use of in the design and training of the neural networks. Further, one may notice that the input space for the MISO ANN is not a subset of the MIMO ANN though the MISO ANN models a subset of the MIMO ANN. Additionally, the input space for a MISO ANN is not necessarily a subset of the MIMO ANN. For example, the lateral dynamics MISO ANN (phiNN) models the roll attitude which is part of the MIMO ANN (LANOF) output.

To conclude, the system identification principles discussed earlier are still in effect and constitute the general theoretical framework. However, there is no guaranteed formula establishing inputs and topology [82]. Thus, the general approach to select the input space for the neural networks involves these guidelines plus some engineering understanding of the problem and several iterations of training. Many design works have confirmed this approach (see e.g. [24, 31, 44]).

4.5 ANN-based FDI for Bell-205 Helicopter

In Chapter 3, we gave details on the architecture of ANN and its training. As well, the capability of an ANN in dealing with plant nonlinearities has been highlighted. Here we will present a detailed design of an ANN-based FDIA for the Bell-205 Helicopter.

In early stage of the design, we have made the decision to split the fault-tolerant flight control system into lateral and longitudinal. For that reason, the FDIA has been split as well into lateral FDIA and longitudinal FDIA. Both will be presented in detail hereafter. The concept we have applied capitalizes on the strengths of the ANN architecture and learning capabilities to generate the predictions required on the detection level and on the isolation level as well.The following features have been considered:

- We have intentionally selected to split the problem into lateral and longitudinal in order to keep the work in-line with future research work where the FTFCS will be decoupled. The reason has been stated clearly in [77] where off-axis coupling is poorly predicted by the DERA model that will be used for controller design. It has been demonstrated that decoupling lateral and longitudinal dynamics helps to remedy the situation. *However, in the design and training of the lateral and longitudinal ANNs, there was no restriction in the selection of the ANNs input variables.*

- We have considered the modelling of all the Bell-205 sensors that are involved in the lateral and longitudinal feedback loops. The detection is done via two MIMO MLP with Time-Delays; LANOF for the lateral dynamics sensors prediction. The isolation is done via MISO MLP for each sensor; namely, phiNN, pNN, and rNN. Similarly, the longitudinal FDIA uses the same structure. In the framework of fault-tolerant flight control system, many researchers have used ANN in modelling and control extensively (see e.g. [16–18, 30, 36, 43, 44, 60]). In the majority of the reported work, focus is on the three sensors; namely the pitch attitude attitude (θ), pitch rate q, and yaw rate (r). In our design, we have considered all the sensors in the feedback loops which makes the design of large-scale

- During the design, we have targeted the effort to maintain independence of the neural networks input space from other sensor readings to avoid faults in the other sensors impacting on the estimates generated by the neural network.

4.5.1 ANN-based FDIA for Bell-205 Helicopter - Lateral Case

As we have seen in Chapter 2, the lateral dynamics can be described by the following input/output relationships. The inputs are the: roll demand (DANET), yaw demand (DR-NET), and roll rate demand(DPNET). The outputs are: the roll magnitude (ϕ), the yaw rate (r), and the roll rate (p). As we saw in the terminology section- 4.1, the faults are detected via residuals vector checking. The consistency check of the plant signals against the estimates reveals the presence or absence of faults. In order to generate such residuals, the designer has to make use of the available signals.

In our design, we used the two-stage approach where the first one models the plant healthy measurements collectively and second one acts as a classifier of faults for every single output. The two-stages approach has been merged together and the approach has been demonstrated with the classical two tanks problem [71].

In the thesis, the whole work is based on MLP with Time-Delays which may be considered equivalent to RNN. MLP has been used as on-line estimator that mimicked the physical sensor reading and that is why it is called virtual sensor in some references. Even though we used the two-stage approach, we have targeted the effort towards distinctive features mentioned above.

With the concept explained in Section 4.4.2, the ANNs have the following architectural properties:

Table 4.1: Bell-205 Helicopter Lateral FDIA Design Parameters

ANN	**Architecture**	**No. of Time Delays**	**Hidden Layer Activation function**
LANOF	4-10-30-3	3	tangsig
phiNN	4-10-20-1	3	tangsig
pNN	7-35-1	2	tangsig
rNN	30-1	4	tangsig

Finally, as per the concepts in Section 4.4.3, the input variables for the various Neural Networks (NN) are as shown in the following table where for the definitions of the variables, one may refer to the appendix at the end of the thesis (Appendix-C).

Table 4.2: Bell-205 Helicopter Lateral FDIA Neural Networks Input Space

ANN	Input Space
LANOF	FDA, FDR, PHI-DOT, AZ, UDOT, WDOT
phiNN	FDA, FDR, AY,VDOT,BETA
pNN	FDA, FDR, PHI-DOT,AZ
rNN	FDA, FDR ,AY, VDOT, BETA

The relationship between the estimation generated by the ANN and the model output is depicted in the diagram below (Figure 4.3) where:

$$y_k = f(\phi_k, F)$$

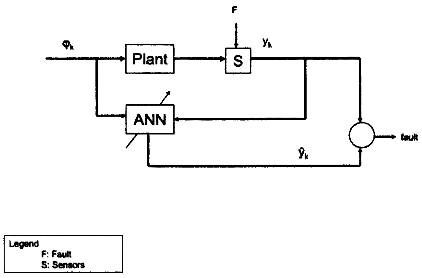


Figure 4.3: Fault Representation

The simulation model for the lateral channel is depicted in the following diagram where the interaction between the neural networks and the decision logic module (DLM) is apparent:

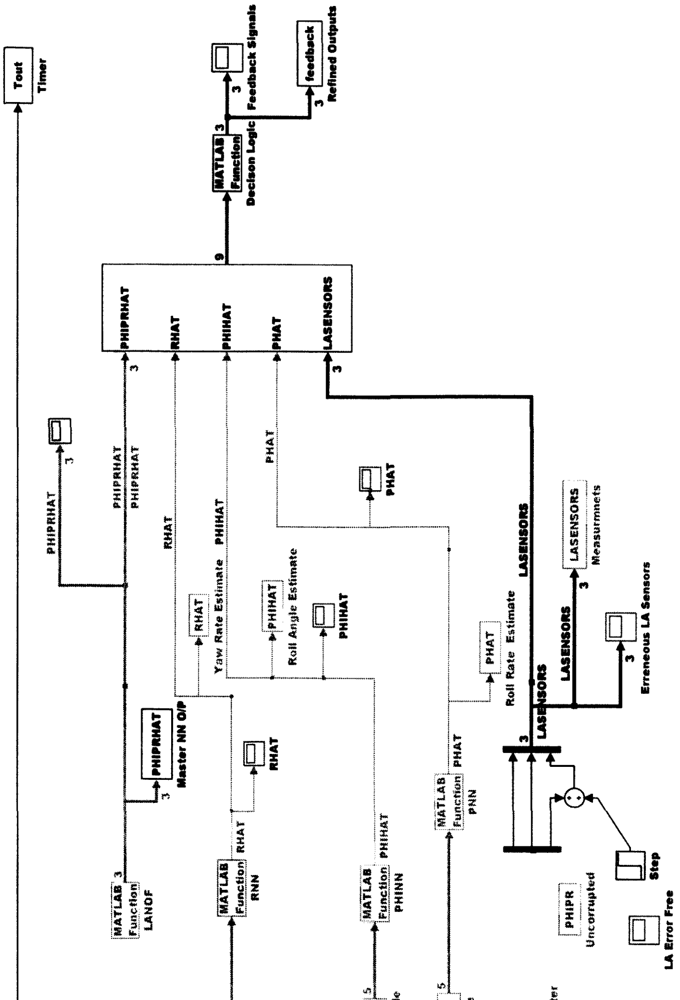


Figure 4.4: Lateral Dynamics FDIA Simulink Model

4.5.2 The Lateral Decision Logic Module (DLM)

The fault detection is handled by the master ANN while the fault isolation is handled by the individual ANNs. The control of this process in addition to the accommodation is via the **Decision Logic Module (DLM)** which is represented by the following equation:

$$y_{DLMi} = \begin{cases} y_i & \text{if } (\varepsilon < \epsilon_{max} \text{ or FAULT}_i = 0) \text{ or if } (\varepsilon < \epsilon_{min} \text{ or FLAG}_i \leq 32), \\ \hat{y}_i & \text{otherwise.} \end{cases} \qquad (4.7)$$

where:

y_{DLMi} refers to the DLM output with respect to the i^{th} channel
where $i = 1, 2, 3$

y_i refers to the i^{ith} channel output

$\hat{y}_i$ refers to the corresponding ANN estimate for the i^{th} channel

FAULT$_i$ A flag used to mark a sensor faulty

FLAG$_i$ A counter used to count the number of consecutive samples during which the discrepancy between the actual sensor output and the ANN estimate for the i^{th} channel has exceeded the specified threshold

The detailed of the DLM is depicted in the following diagram (Figure 4.5) for the lateral channel (ϕ, p, r)).

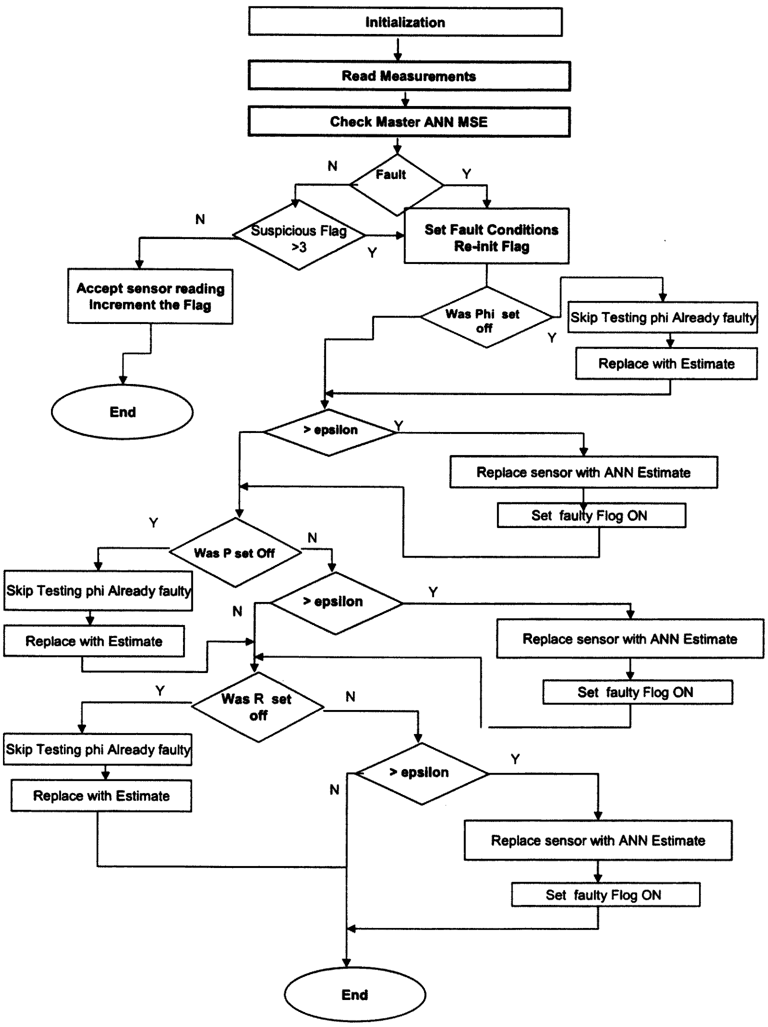


Figure 4.5: Decision Logic Module (DLM) Flowchart

4.5.3 ANN-Based FDIA for Bell-205 Helicopter - Longitudinal Case

We have considered the modelling of all Bell-205 sensors that are involved in the lateral and longitudinal feedback loops. The detection is done via two MIMO MLPs with Time-Delays; LONOF for longitudinal dynamics sensors prediction. The isolation is done via MISO MLP for each sensor: thetaNN, and qNN. The architectural properties of the ANNs have been embarked on as per the concepts laid out in Section 4.4.2 :

Table 4.3: Bell-205 Helicopter Longitudinal FDIA Design Parameters

ANN	Architecture	No. of	Hidden Layer
		Time Delays	Activation function
LONOF	7-30-2	4	tangsig
qNN	10-35-1	5	tangsig
thetaNN	7-30-1	4	tangsig

The input variables for the various neural networks are shown in the following table (Table- 4.4) where the guidelines of Section 4.4.3 are made. For the definitions of the variables, one may refer to the Appendix-C at the end of the thesis.

Table 4.4: Bell-205 Helicopter FDIA Neural Networks Input Space

ANN	Input Space
LONOF	FDE, FDA, BETA, UDOT, VDOT, WDOT
thetaNN	FDE, FDA, BETA, UDOT,VDOT, WDOT, AZ, AX
qNN	FDE, WDOT, AZ, AX, AZ./AX

The Simulink diagram for the longitudinal dynamics sensors FDIA is depicted in the following diagram:

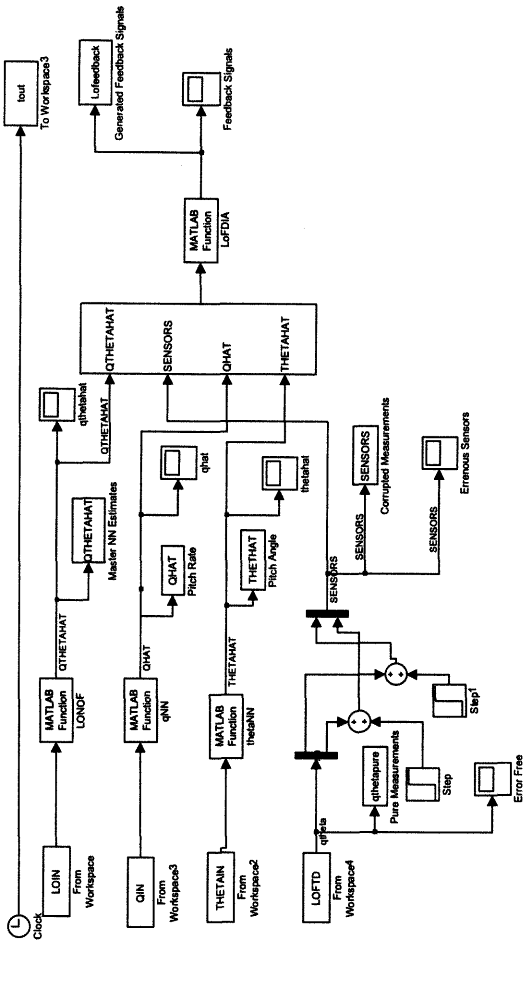


Figure 4.6: Longitudinal Channel Simulink Model

4.5.4 The Longitudinal Decision Logic Module (DLM)

The fault detection is handled by the master ANN while the fault isolation is handled by the individual ANNs in the same manner of the lateral dynamics. As we can see, once the master ANN threshold is exceeded a number of times, a fault is declared. At that point of time, it is determined to identify which sensor is faulty. It could be a single or multiple faults but it is irrelevant to identify the magnitude of the fault though it is possible. The fault identification process is performed by checking the threshold of all MISO ANNs. In the end, the DLM takes the following actions:

- Healthy signal is passed on to the feedback loop.
- The faulty sensor (the one with the threshold is exceeded) is replaced by its estimate.

If the sensor is faulty, its flag is set on, which helps in successive check cycles by just replacing the signal with its estimate rather than checking it every time. With this arrangement, it is possible to continue the operation of the helicopter till the end of its journey without any safety problems though the performance may be lightly degraded. The control of this process in addition to the accommodation is via the **Decision Logic Module** (DLM), which is similar to the one used by lateral channel.

4.5.5 Threshold Selection

The threshold value selection is very much dependent on the performance of the ANN estimator and the training data set. The selection of a threshold value is of prime importance prior to the full deployment of the ANN-based FDIA in a real application. In other words, these thresholds must be revised to make sure that they are based on representative training data.

During training, the following equations have been observed to judge the quality of the estimates:

$$\begin{aligned}
\epsilon_{LANOF} &= 1/2((\phi_i - \hat{\phi}_i)^2 + (p_i - \hat{p}_i)^2 + (r_i - \hat{r}_i)^2) && \text{(4.8a)}\\
\epsilon_{LONOF} &= 1/2((\theta_i - \hat{\theta}_i)^2 + (q_i - \hat{q}_i)^2) && \text{(4.8b)}\\
\epsilon_{phiNN} &= 1/2((\phi_i - \hat{\phi}_i)^2) && \text{(4.8c)}\\
\epsilon_{pNN} &= 1/2(p_i - \hat{p}_i)^2 && \text{(4.8d)}\\
\epsilon_{rNN} &= 1/2(r_i - \hat{r}_i)^2 && \text{(4.8e)}\\
\epsilon_{thetaNN} &= 1/2(\theta_i - \hat{\theta}_i)^2 && \text{(4.8f)}\\
\epsilon_{qNN} &= 1/2(q_i - \hat{q}_i)^2 && \text{(4.8g)}
\end{aligned}$$

The threshold values for MIMO ANNs (first stage ANNs) have minimum and maximum thresholds. The maximum threshold is selected by comparing the FTD against the MIMO ANN. Comparisons generate the minimum, mean and maximum. The maximum is used to indicate malfunctioning of the sensor. The minimum is used to indicate a possible fault and to put the sensors under monitoring for a number of samples. The threshold values for MISO ANNs are found in the same way. However, minimum values are considered adequate for fault isolation. If mean or maximum are chosen, then some faults may not be isolated. Monitoring the sensors by the MIMO ANNs may end up with a conclusion that it was only a false alarm. However, if the fault detection networks have decided that a fault has taken place, the MISO ANNs check which sensor is faulty. At the end it could be that one or more sensors are faulty.

Based on the FTD used for training, the threshold values considered in the **Decision Logic Module** are as follows:

Table 4.5: Neural Networks Threshold Values

ANN	**Threshold Value**
LONOF	1.405 (min)-2.855(max)
qNN	0.1032
thetaNN	0.82
LANOF	0.7974-2.19
pNN	0.0929
rNN	0.9879
phiNN	1.141

A final remark that we would like to emphasize, is about the DLM parameters. We have selected them to let the system take decisions after (32 samples) to avoid responding to false alarms and intermittent problems. Also, for the fault detection phase, the MIMO ANN is provided with two limits, an upper and lower one as explained above for this reason. It is worth mentioning that **the decision logic may be further revised and undergo several improvement and alterations.**

4.6 ANN Training and Testing Results

The outputs of the neural networks that we have used for detection and isolation are shown in the following figures. One can notice from the figures below that an acceptable accuracy can be attained provided the neural networks are properly designed and trained. Normally, once an ANN is designed, its performance is checked against the data used for the training. This is not sufficient. An additional training data set that has not been used during the training but fall within the same operating point is presented to the trained ANN to validate the behavior of the ANN. Satisfactory results confirm that the ANN is capable to represent the physical system in the used FTD operating range. Here below, we will present the results of the comparisons of ANNs performance with the training data and then with the validation data. All the comparisons are made against Flight Test Data (FTD).

The following set of plots show the outputs of the ANNs in the lateral channel compared to the FTD after being trained. Figure 4.7 shows the quality of the master ANN (LANOF) against FTD and as can be seen they are almost identical. Figure 4.8 shows the quality of phiNN compared to FTD where there is no significant discrepancy between the actual and estimate. However, there are some spikes at samples: 1000, 2500, and 2750 due to noise. Figure 4.9 shows the quality of pNN after being trained against FTD where the overall shape of the signal is captured. There are minor discrepancies at some samples but this should not be of great concern as the DLM is flexible to avoid responding to mismatch unless it is persistent. Finally, Figure 4.10 shows the quality of rNN vs. FTD where we can see again the impact of noise at some samples (e.g. at sample: 1400 and 3500).

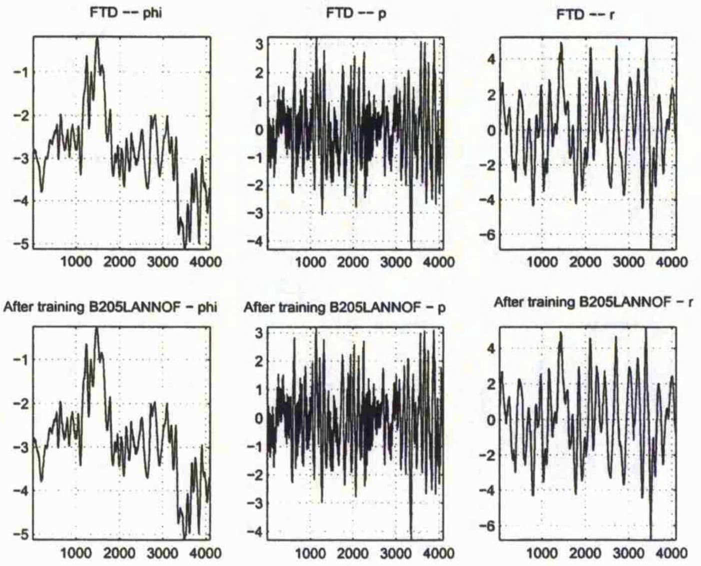


Figure 4.7: Master ANN for Lateral Dynamics Model (LANOF) - after training vs. FTD

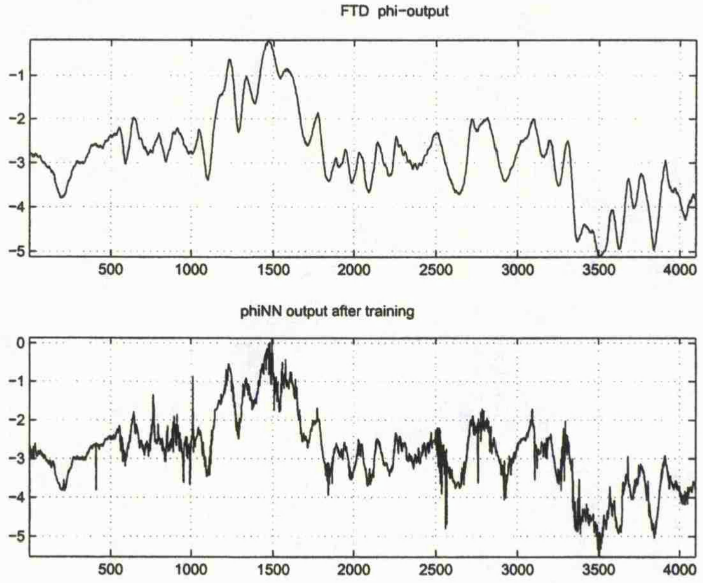


Figure 4.8: MISO ANN used to model Roll Attitude ϕ to mimic phi-sensor output (phiNN) After training vs. FTD

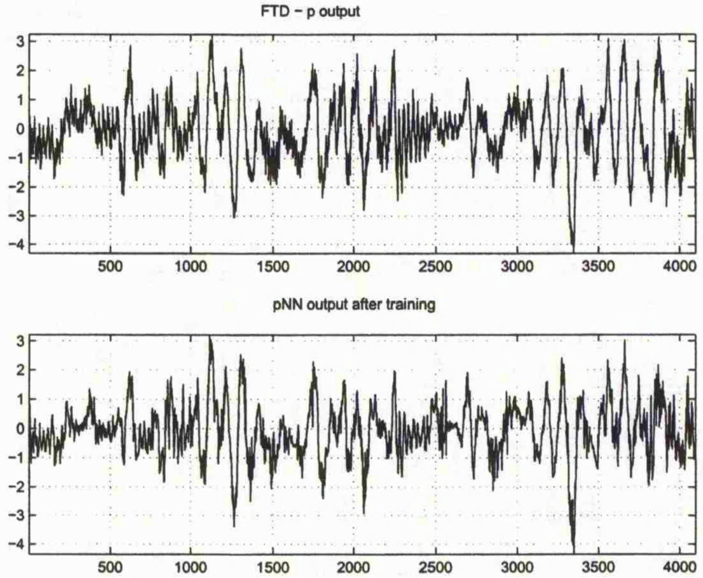


Figure 4.9: MISO ANN used to model Roll rate p to mimic p-sensor output (pNN) After training vs. FTD

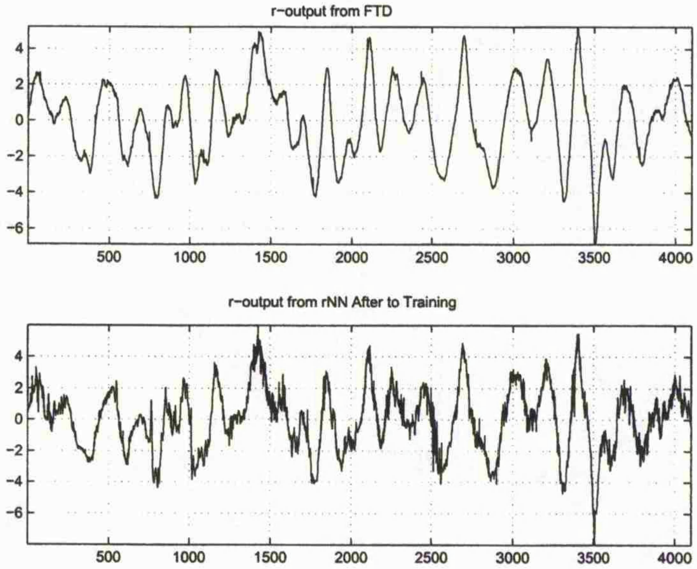


Figure 4.10: MISO ANN used to model Yaw Rate r to mimic r-sensor output (rNN) After training vs. FTD

Having finished the training, the four ANNs have been validated with a different data set. Here, the results of validation of the ANNs is demonstrated by the following figures from Figure 4.11 to Figure 4.13:

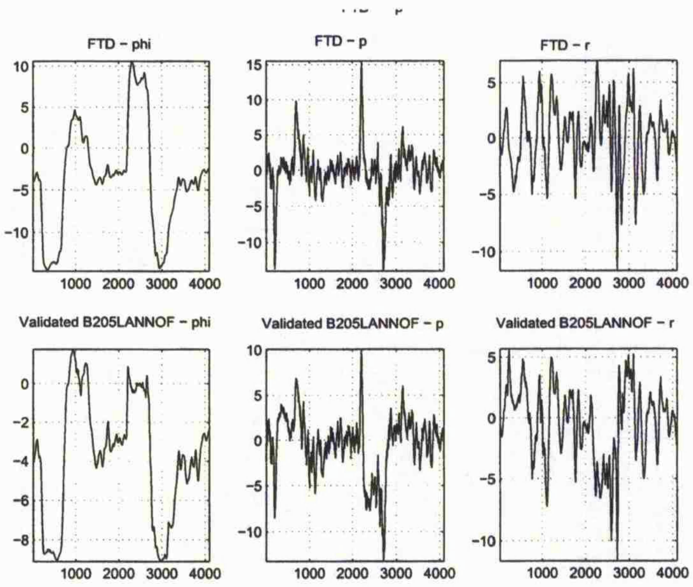


Figure 4.11: Master ANN for Lateral Dynamics Model (LANOF) - after validation vs. FTD

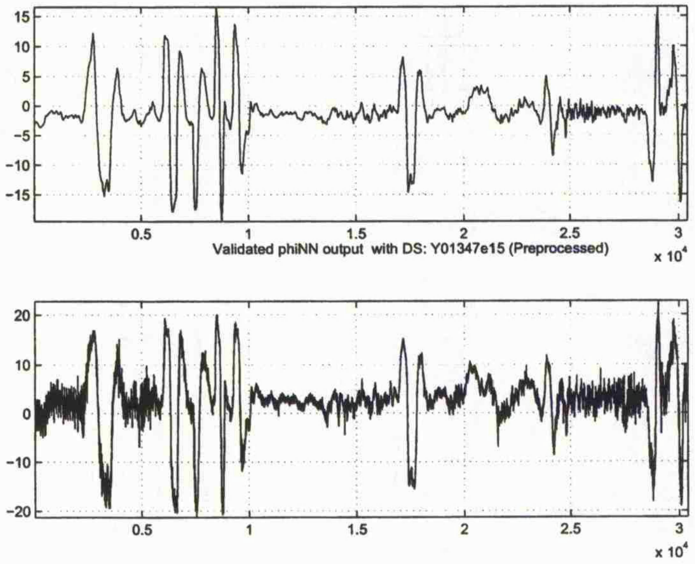


Figure 4.12: MISO ANN used to model Roll Attiude ϕ to mimic phi-sensor output (phiNN) after validation vs. FTD

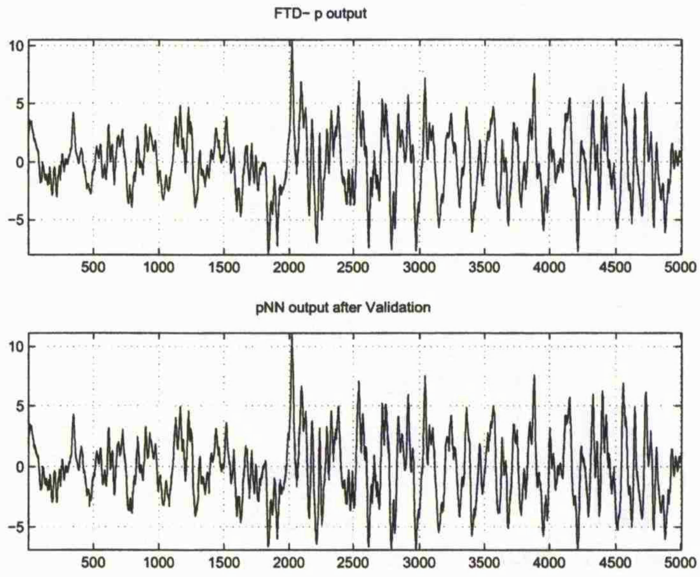


Figure 4.13: MISO ANN used to model Roll Rate p to mimic p-sensor output (pNN) after validation vs. FTD

On the longitudinal channel, the following set of plots demonstrate the quality of the ANNs after being trained against FTD.

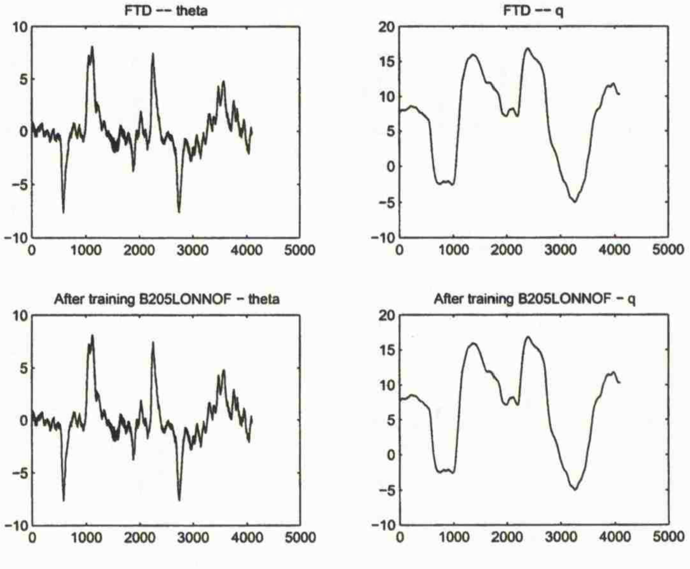


Figure 4.14: Master ANN for Longitudinal Dynamics Model (LONOF) - after training vs. FTD

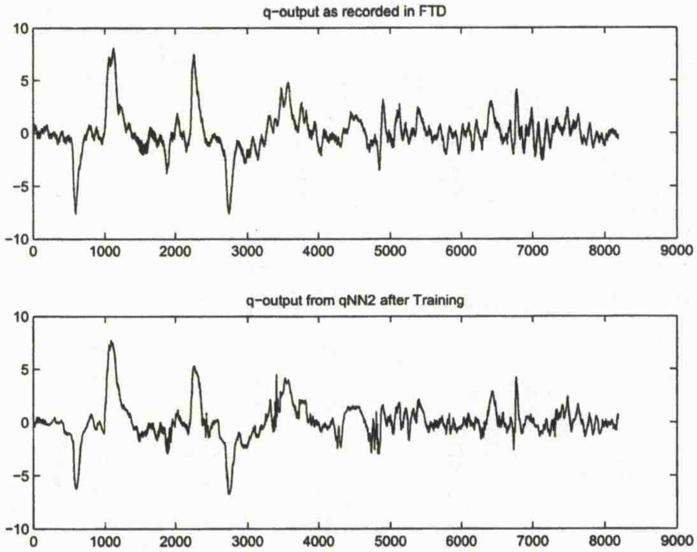


Figure 4.15: MISO ANN used to model Pitch Rate q to mimic q-sensor output (qNN) after training vs. FTD

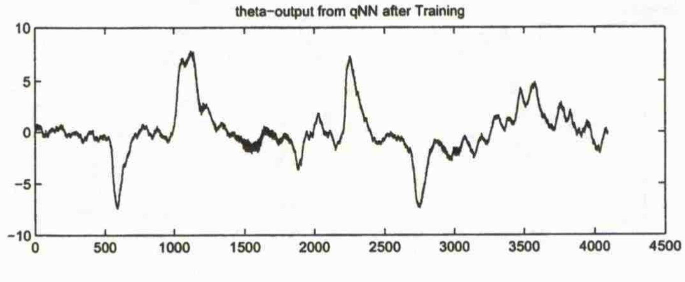


Figure 4.16: MISO ANN used to model Pitch Attiude θ to mimic θ-sensor output (thetaNN) after training vs. FTD

The following figures show the result of the validation:

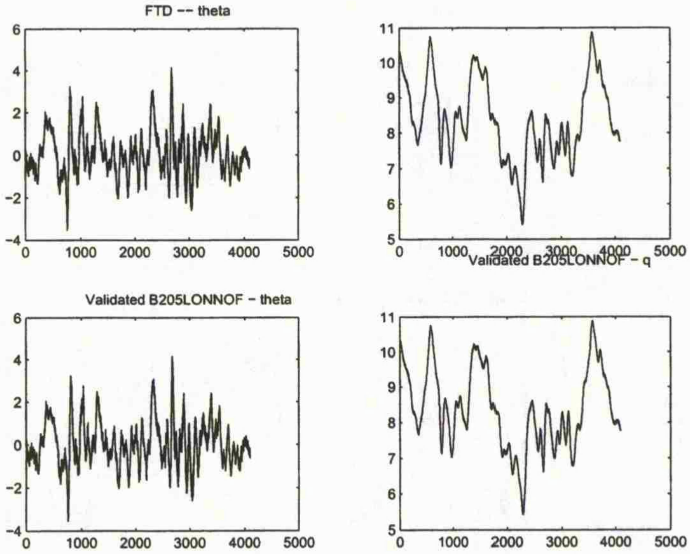


Figure 4.17: Master ANN for Longitudinal Dynamics Model (LONOF) - after validation vs. FTD

Before we leave this section, we want to point out that it has been stated in many references that ANN is noise-tolerant; i.e it can handle noise presence in the input data and still be able to produce highly acceptable results. The following figure (Figure 4.18) shows that though the ANN is able to capture the shape and magnitude of the signals, filtering the input data helps to improve the prediction quality by removing much of the spikes in the data caused by the noise.

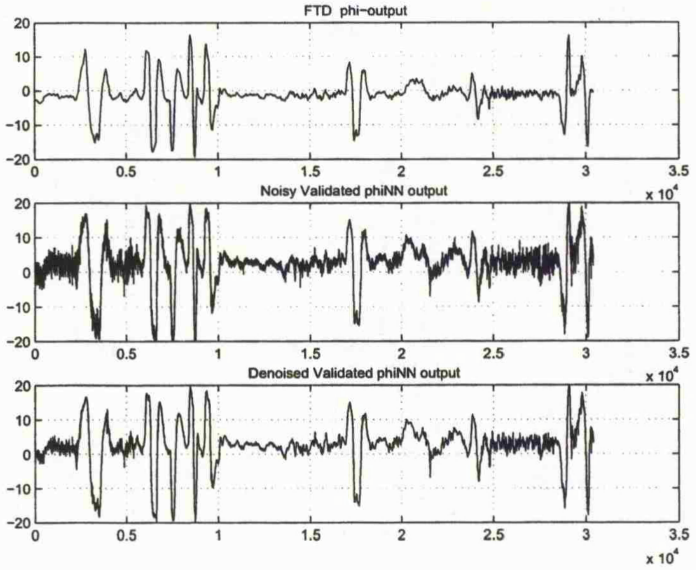


Figure 4.18: Effects of Noise on Quality of Prediction

4.7 Statistical Analysis

The results presented in previous sections have shown satisfactory performance of the various ANNs used in the lateral and longitudinal FDIA systems. This does not mean that the ANN-based FDIA system can be deployed prior to collecting rich FTD and retraining the ANNs. The quality of the designed ANNs has been judged on the basis of the shapes of the generated signals and some quantitative comparison. In order to accurately quantify the quality of these ANNs, the autocorrelation measure is used. The fundamental idea behind this statistical tool is that when the error is autocorrelated and plotted, the plot must show randomness between successive samples. This method is used in checking the quality of models against the actual plant. A typical shape for the error autocorrelation plot is depicted in Figure 4.19 below:

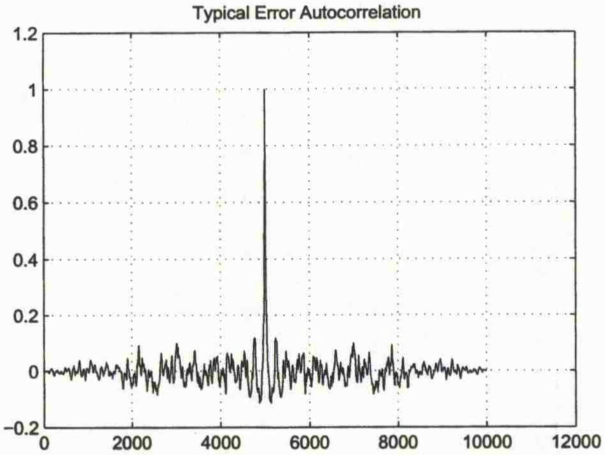


Figure 4.19: Typical Error Autocorrelation Plot

4.7.1 Lateral ANN-based FDIA

The statistical analysis is performed for the MIMO ANN (LANOF) on two levels. First, the individual output error is analyzed and, then, the collective error for the three signals is analyzed. In both cases, the reference is the flight test data. The error plot is shown on the upper pane of the diagram below (Figure 4.20) and the error autocorrelation is shown in the lower pane.

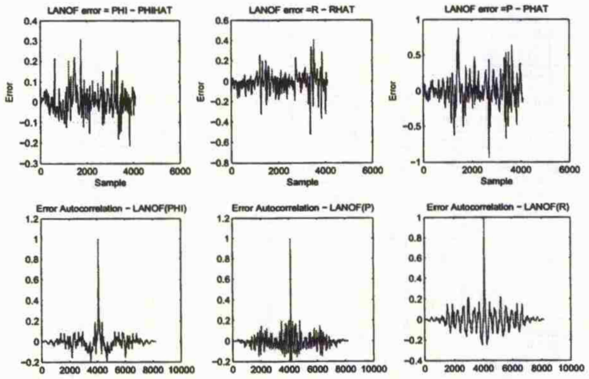


Figure 4.20: LANOF Error and Error Autocorrelation Plot

From the above figure (Figure 4.20), the following comments are drawn:

- The errors for all channels are bounded as shown in the following table (Table 4.6). These errors are contained in the threshold selected.

Table 4.6: Error bound for LANOF ANN

Channel	Absolute Error Bound
ϕ	0.3
r	0.4
p	0.9

- The autocorrelation plot is in line with the general shape of the typical plot shown in Figure 4.19. The negative values on the plot indicate a change of sign in the successive errors.

If LANOF outputs are compared collectively against the FTD, the following diagram (Figure 4.21) is obtained:

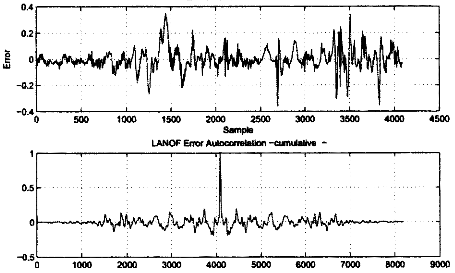


Figure 4.21: LANOF Error and Error Autocorrelation Plot - Collective'

From the above figure (Figure 4.21), the following remarks are drawn:

- The absolute error is bounded by an upper value of 0.4

- The autocorrelation plot is similar to the typical plot.

For the roll attitude ANN (phiNN), the following figure shows the error plot in the upper pane and the error autocorrelation in the lower pane:

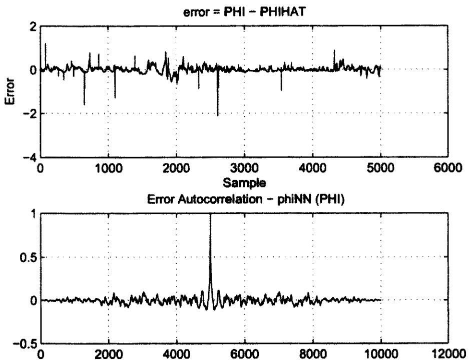


Figure 4.22: phiNN Error and Error Autocorrelation Plot - Collective'

From the above figure (Figure 4.22), the following remarks are drawn:

- The error plot shows that the absolute error is bounded by an upper value of 2.0. Actually, the error mostly within an upper bound of 1.0 and the spikes may be due to some outliers.
- The error function is quite similar to the typical autocorrelation plot which indicates that the errors of the successive samples are quite random.

For the roll rate ANN (pNN), the error and the error autocorrelation plots are depicted in the following diagram:

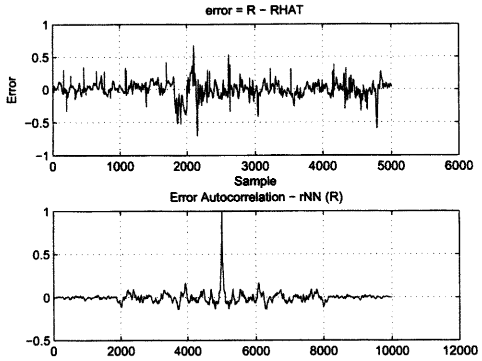


Figure 4.23: pNN Error and Error Autocorrelation Plot - Collective'

From the above figure (Figure 4.23), the following remarks are made:

- The absolute error upper pound is $\simeq$ 0.65 which is contained by the threshold value of the DLM.
- The error autocorrelation plot indicates randomness of the errors.

Finally, the yaw rate ANN (rNN) error and error autocorrelation is depicted in the following figure:

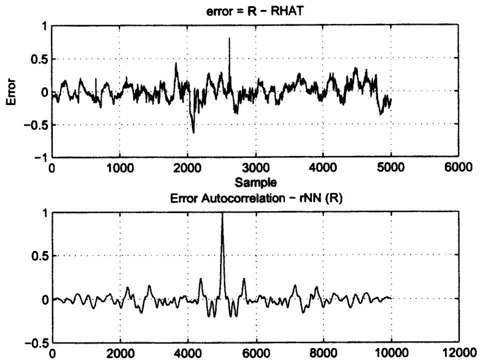


Figure 4.24: rNN Error and Error Autocorrelation Plot - Collective'

From the above figure (Figure 4.24), the following remarks are drawn:

- The absolute error is bounded by an upper value of 0.75 which is contained in the threshold value of the DLM.
- The error autocorrelation is quite typical and confirms error randomness among successive samples

4.7.2 Longitudinal ANN-based FDIA

The error function plot for the MIMO ANN (LONOF) and the error autocorrelation plot are shown in Figure 4.25. The upper panes show the difference between the ANN and the flight test data. The first pane in the left shows the error size pertaining to the pitch attitude and the second pane shows the error size pertaining to pitch rate.

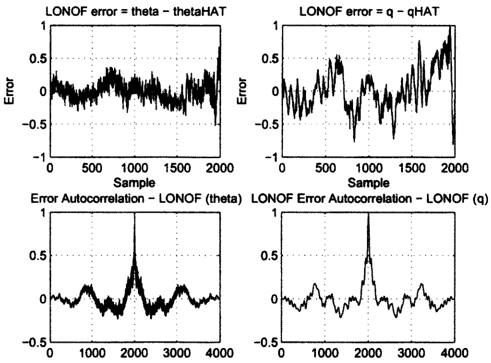


Figure 4.25: LONOF Error and Error Autocorrelation Plot

The following remarks are drawn from the figure:

- The absolute error for both signals (θ and q) is bounded by an upper value ≤ 1.
- The error autocorrelation plots for both signals are similar to the typical error autocorrelation plot (Figure 4.19) which mean that errors between the successive samples are random. This indicates that the ANNs have extracted all the features in the training data.
- There is some periodicity in the error autocorrelation plots which may be due to noise injected in the flight test data.

The MIMO ANN outputs are used collectively to check any fault in the longitudinal dynamics sensors. In other words, the outputs of the MIMO ANN (LONOF) are compared as a whole against the sensors as a whole. Thus, the following diagram (Figure 4.26) is obtained:

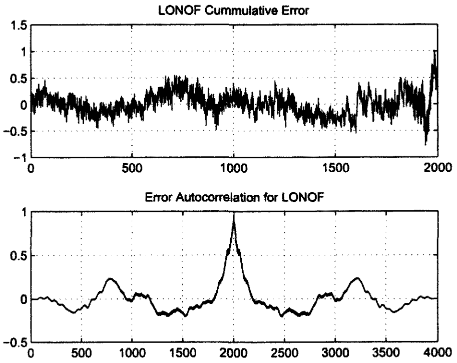


Figure 4.26: LONOF Error Plot and Error Autocorrelation Plot

The figure above confirms the conclusion regarding the bound of the error and the randomness of the errors. The error bound for the neural network is contained by the threshold values selected for this ANN.

The plots of the errors and the errors autocorrelation of the MISO ANNs (thetaNN and qNN) are shown consecutively below. The diagram below (Figure 4.27) shows the error plot and the autocorrelation plot for thetaNN:

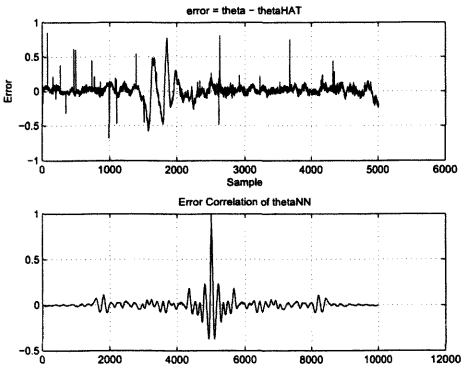


Figure 4.27: Error Plot and Error Autocorrelation Plot for thetaNN

From the figure, the following remarks are drawn on the quality of thetaNN output:

- The absolute error for the output is bounded by an upper value of 0.7.
- There are some spikes in the error which may be due to outliers in the training data. In spite of these spikes, the error is still within the threshold values.
- The error autocorrelation plot is similar to the typical error autocorrelation plot (Figure 4.19) which means that the errors between the successive samples are random which indicates that the ANN has extracted all the features in the training data.
- There is some periodicity in the error autocorrelation plots which may be due to noise injected in the flight test data.

For qNN, the diagram below (Figure 4.28) shows the error plot and the autocorrelation plot:

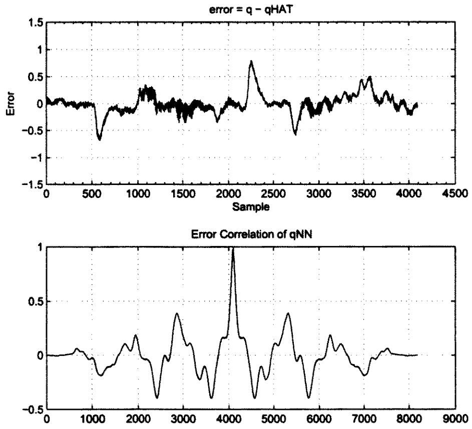


Figure 4.28: Error Plot and Error Autocorrelation Plot for qNN

From the figure, the following remarks are drawn on the quality of thetaNN output:

- The absolute error for the output is bounded by an upper value equal to 0.75.
- The shape of the error autocorrelation plot is similar to the typical error autocorrelation plot (Figure 4.19) which means that errors between the successive samples are random which in turn indicates that the ANN has extracted all the features in the training data.
- Periodicity in the error autocorrelation plot is more apparent than before which may be due to noise injected in the flight test data.

4.8 FDIA Simulation Results

We have developed ANN models for all sensors in the lateral and longitudinal channels with a master ANN for each channel of the Bell-205 helicopter. The overall system has shown its capability to model the sensors with a satisfactory degree of accuracy which in turn helps to detect faults, isolate them and enables the decision logic to accommodate the faults. The faults considered are biases which are modelled by step function with various magnitudes being added to the original signal. It is quite possible to consider other types of faults such as a ramp type. However, as stated before the ANN-based FDI with the structure we developed does not require fault signature and, thus, any type of faults can be considered.

For the lateral channel, we have simulated a single fault (Figure 4.29) and two simultaneous faults (Figure 4.30) both at time (t=2 seconds). The successful detection, isolation, and accommodation is shown in (Figure 4.31). The system continued the operation normally till the end of the simulation.

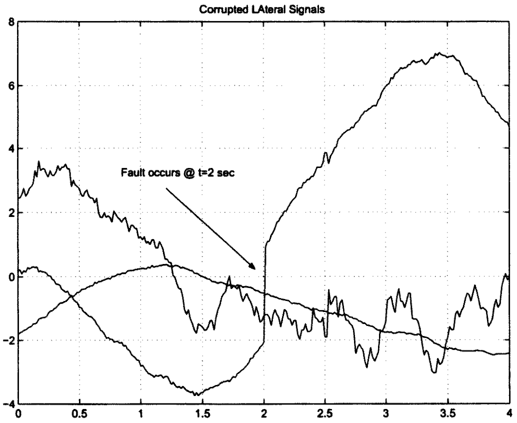


Figure 4.29: Lateral Sensors' Measuremnts with one fault occurs at time t= 2 seconds

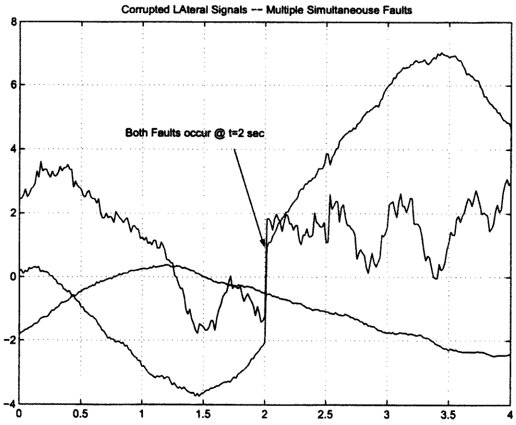


Figure 4.30: Lateral Sensors' Measuremnts with two Simultaneous faults occurring at time t= 2 seconds

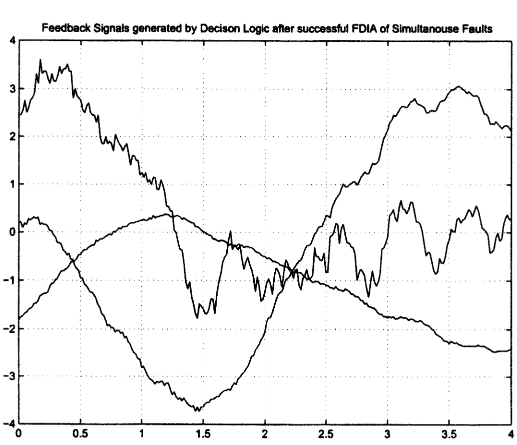


Figure 4.31: Successfull Fault Detection, Isolation, and Accommodation of Multiple Simultaneous Faults

For the longitudinal channel, we have simulated multiple faults at time (t=1 second) as

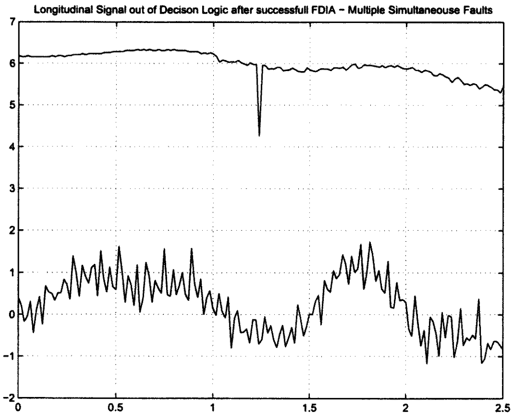


Figure 4.33: Successfull Fault Detection, Isolation, and Accommodation of θ signal

shown in (Figure 4.32) and the system has successfully detected them, isolated them and accommodated them as shown in (Figure 4.33).

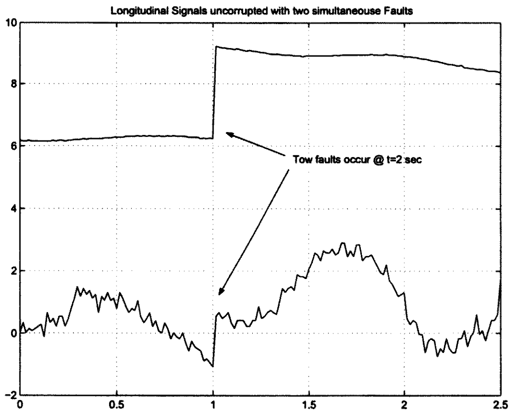


Figure 4.32: Longitudinal Sensors' Measuremnts with two Simultaneous Faults

Prior to leaving this section on simulation results, it is worth mentioning that in order to bring the simulation results to real application, the training has to be revisited in order to cover the typical flight envelope.

4.9 Conclusion

A number of lessons have been gained during the design and testing of the ANN-based FDIA systems for the Bell-205 helicopter. Here below a recording of the most important ones:

- ANN **Design**: In the literature there are some conclusions about the size of the ANN that may be viewed as contradicting with each other. For example, Hornik et. al. in [54], has proved mathematically that an MLP with as few as a single hidden layer is indeed capable of universal approximation of any function in a very precise and satisfactory sense. In our design we found this claim is generally true where a single hidden layer is quite sufficient. In other situations, as indicated above, two hidden layers ANNs give more satisfactory results. This needs to be taken care of when designing ANNs. The need for more than a single hidden layer may arise from poor training data. It is worth mentioning that the theoretical basis for the number of layers is under investigation (see e.g. [40]).

- **Number of Neurons in Hidden Layer**: A similar discussion can also be extended to the number of neurons in the hidden layer. It is indicated that as the number of neurons in the hidden layer increases, the accuracy and convergence speed increase as well. In other references, the advice is to start with a small number of neurons. In [57], comparative analysis indicates a strong relationship between the number of neurons in a hidden layer and the number of training cycles[2] and the training data set size which collectively affect the mean square error between the ANN output and the desired output and the degree of the ANN output quality.

- **MIMO vs. MISO modelling**: In [29], it is been stated that (...models for predic-

[2] In batch training a number of cycles is repeated where in each cycle the whole data set is presented to the ANN and after that the weights and biases are updated. In Matlab, the cycle is called epoch

tion and control will be able to produce better results if constructed for all outputs simultaneously). This means that building a MIMO Model for a group of sensors (such as the ones we did in LANNOF and LONOF) produces better results than those by models which are MISO for individual sensors (such as the one for pitch angle or its rate). By inspecting the mean square errors (MSE) for the MIMO and MISO models even when using the same data set, sometimes the MISO MSE is less accurate than that of MIMO.

Chapter 5

Integration of ANN-Based FDI and H_∞ Controller

The basic theme of this chapter is to test the functionality of the proposed FTFCS scheme using the ANN-based FDI system developed in Chapter 4 with lateral and longitudinal controllers that were designed using the H_∞ optimization method. The controllers are then integrated with the aforementioned FDI system and the final system is then tested in simulation. Due to the problem of acquiring the nonlinear model, the DERA linear model has to be used in the test. A set of data has thus been generated by running the linear model and a scaled down version of the FDI system designed is based on the data.

Designing controllers for Bell-205 helicopter is not an objective by itself in this work. Rather, they are designed to be integrated with the FDIA system to test the functionality of the concept of the FTFCS. Various H_∞ controllers for Bell-205 Helicopter have been designed and tested by the Leicester Engineering department not only in simulation but also in piloted flight tests [19, 23, 25, 76, 77, 81]. Previous design experience accumulated in the department has led to the conclusion that the available models for the Bell-205 helicopter are incapable of predicting cross coupling accurately. This has motivated the decoupled design of the controllers for the lateral and longitudinal dynamics [14, 77]. This approach is followed in our design of the FTFCS. We use the S-over-KS algorithm to design the lateral and longitudinal controllers. In this chapter we first give a brief introduction on H_∞ control theory in general and mixed-sensitivity design in particular (or equivalently called S-over-KS, S-over-T, or S-over-KS-over-T). Afterwards, the design of the lateral and longitudinal controllers for the Bell-205 helicopter is to be presented. The overall system structure is as shown in Figure 5.1 below.

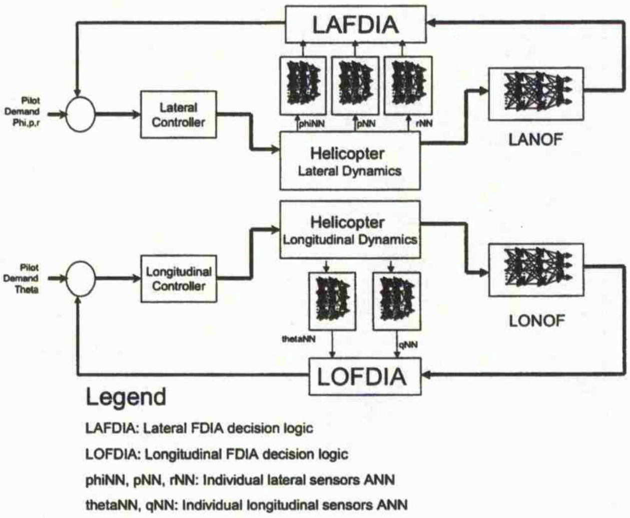

LAFDIA
Pilot Demand Phi,p,r
Lateral Controller
phiNN
pNN
rNN
Helicopter Lateral Dynamics
LANOF
Helicopter Longitudinal Dynamics
Pilot Demand Theta
Longitudinal Controller
LONOF
thetaNN
qNN
LOFDIA
Legend
LAFDIA: Lateral FDIA decision logic
LOFDIA: Longitudinal FDIA decision logic
phiNN, pNN, rNN: Individual lateral sensors ANN
thetaNN, qNN: Individual longitudinal sensors ANN

Figure 5.1: Plant-Controller Architecture with Sensor FDIA in the loop

5.1 H_∞ controller Design

Classical control methods which are mainly for single-input single-output (SISO) system are still popular in the industrial community in spite of the fact that modern techniques have proven successful and have overcome many of the shortcomings of the classical techniques [7]. Major shortcomings of the SISO design approach lie in the ignorance of the coupling in the multivariable system and the inability of dealing with the differences between physical models and mathematical models. Both of these issues are of paramount importance. Control system design relies on mathematical models which are normally nonlinear. The model usually does not represent the complete physical system. Actual systems have various dynamic perturbations as well as dynamics which are difficult, if not totally impossible, to be included in a model. Further, a physical system may undergo ceratin variations due to changes in certain conditions under which the system operates.

In addition, linear control design uses a linear model which is only an approximation of the nonlinear model around a certain operating point. All these problems affect the performance of a controller and if it is not robust enough, the plant may face instability or severe performance degradation. A robust controller is even more important in the proposed FTFCS because of the possible errors in estimation of the (healthy) system outputs generated by the ANNs. H_∞ control algorithms provide a framework for designing robust controllers. Robustness means that a designed closed-loop system should maintain internal stability and adequate performance in the presence of certain uncertainties. The design process is based on the idea of mapping the design objectives into a single matrix transfer function and minimizing its infinity norm. Design requirements could include disturbance rejection, noise attenuation, certain time response requirements (such as rising time), and frequency response requirements.

It is not intended to cover all the details of the theory of H_∞ here but rather to give a brief coverage that is deemed necessary to bring the chapter into context. Details of the subject matter are covered in many references where the theoretical aspects of the subject may be consulted (see e.g. [49, 50, 84] and references therein) while design and analysis with mathematical coverage may be better found in other references (see e.g. [51, 75, 83] and references therein).

Normally the performance and robustness requirements are conflicting and satisfying them all may not be possible. It is the major challenge for the designer to make compromises and to satisfy as many as possible of the requirements. An H_∞ design is posed as a minimization of a cost function that reflects the conflicting requirements in a multi-objective design problem. The cost function is normally written in terms of a linear fractional transformation (LFT) of the interconnected plant (P) and the controller (K) as seen in the diagram below (Figure 5.2).

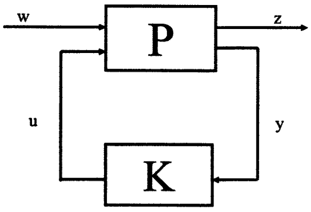


Figure 5.2: The Standard H_∞ Configuration

In the above diagram, w stands for the exogenous input, z stands for the signal to be penalized, u is the control input vector, and y is the output measurements. The interconnected plant matrix P can be written in a partition form as:

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$$

From the above, one can write the transfer function from w to z as:

$$\begin{aligned} T_{zw} &= P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \\ &= \mathcal{F}_l(P, K) \qquad (5.1) \end{aligned}$$

where $\mathcal{F}_l(P, K)$ is the lower LFT. The cost function to be minimized is $\|\mathcal{F}_l(P, K)\|_\infty$ over all the stabilizing controllers which can be re-written as:

$$\min_{K stablizing} \|\mathcal{F}_l(P, K)\|_\infty$$

The relationship between the original plant transfer function $G(s)$ and the interconnected plant P depends on the actual design objectives. This will be seen later on in the S-over-KS procedure. Another major method used in the H_∞ design is the loop shaping design procedure. The H_∞ loop shaping design procedure (LSDP) combines the classical loop shaping procedure for SISO settings and H_∞ robust stabilization technique. The design proceeds in two steps. In the first step, the plant is augmented with pre- and post-compensators to shape the singular values of the open loop system. In the second step, a stabilizing controller is found to maximize the stability margin of the system assumed to suffer uncertainties on normalized co-prime factors of the shaped plant. Controllers designed with this method possess strong robustness properties [6]. The procedure is detailed in many references (see e.g. [75]). As we are to use mixed-sensitivity, the approach is described in more detail in the following subsection.

5.1.1 Mixed-Sensitivity Design

As stated before, there are usually conflicting requirements in control system design. The mixed-sensitivity optimization helps to deal with these requirements. For example, the designer may wish to minimize the effect due to disturbance entering the system as well as the control signal to avoid actuator saturation. Minimization of the combination of the sensitivity transfer function S, which is the transfer function between the disturbance signal (d) and the output (y), and the transfer function KS, which denotes the control signal, will lead to the optimal or suboptimal controller satisfying both requirements. In mathematical terms, it is desirable to minimize the cost function:

$$min \left\| \begin{matrix} S \\ KS \end{matrix} \right\|_\infty$$

To meet actual design specifications, frequency dependent weight matrices are incorporated in the cost function which result in a minimization problem as follows:

$$min \left\| \begin{matrix} W_1 S \\ W_2 KS \end{matrix} \right\|_\infty$$

The weighting functions are the designer knobs via which certain adjustments can be made to attain certain design goals. The guidelines for selection of weighting functions can be found in [55, 58, 59] and the references therein. W_1 is normally chosen to be a low pass filter with a bandwidth equal to that of the disturbance to be attenuated and W_2 is chosen as a high pass filter with a corner frequency equal to the closed loop bandwidth [75]. Selecting W_2 as a high pass filter helps to suppress unmodelled or poorly modelled dynamics in the high frequency range. In the framework of the S/KS approach, the design problem can be formulated as a regulation or tracking problem. In the former, design objectives can include e.g. disturbance attenuation and actuator saturation. In the latter setting, the objectives can be e.g. forcing the output to track certain reference signal.

5.1.2 Bell-205 Helicopter Lateral Controller

The lateral model is previously described in Chapter 2. The input (pilot) commands are the lateral cyclic, δ_{lc}, and the pedal collective, δ_{tr}. The outputs are the roll attitude, ϕ, the roll rate, p, and the yaw rate, r. The closed loop is given by the following diagram (Figure 5.3):

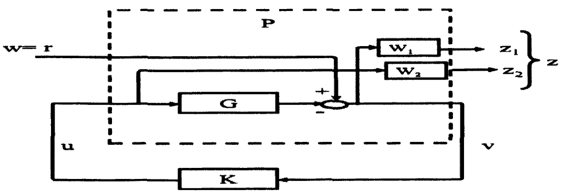


Figure 5.3: Mixed-sensitivity H_∞ Controller

The linear model of the lateral dynamics is given by the following equations:

$$\begin{aligned} \dot{x} &= Ax + Bu \qquad (5.2) \\ y &= Cx + Du \qquad (5.3) \end{aligned}$$

where:

$$A = \left(\begin{array}{cccc} -0.0000 & 1.0000 & -0.0005 & 0 \\ 0 & -2.4732 & -0.0006 & -0.0182 \\ 0 & -0.2716 & -0.6943 & 0.0176 \\ 32.1742 & -4.8115 & -9.1062 & -0.0502 \\ 0.4075 & -32.9361 & -0.0359 & -0.2338 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right.$$

$$\left. \begin{array}{ccc} 0 & 0 & 0 \\ -0.0157 & -20.3127 & 4.6427 \\ 0.0284 & -2.1174 & -13.4637 \\ -0.0494 & -32.4426 & 18.1631 \\ -0.5458 & -21.6488 & 0 \\ 0 & -12.5786 & 0 \\ 0 & 0 & -25.0000 \end{array} \right)$$

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -2.1098 & 0 \\ 0 & 7.0991 \end{pmatrix} C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

The objective of the design is to minimize the disturbance influence and limit control signals which, in mathematical terms, is represented by:

$$\left\| \begin{matrix} \mathrm{W}_1 S \\ W_2 K S \end{matrix} \right\| \leq \gamma$$

As mentioned before, $S = (I - GK)^{-1}$ is the sensitivity function and K is the controller to be designed. W_1 and W_2 are two frequency dependent weighting functions and are in the form of:

$$W_1 = \begin{bmatrix} W_\phi & 0 & 0 \\ 0 & W_p & 0 \\ 0 & 0 & W_r \end{bmatrix}$$

$$W_2 = \begin{bmatrix} W_{FDA} & 0 \\ 0 & W_{FDR} \end{bmatrix}$$

The subscripts ϕ, p, r refer to the output signals while FDA and FDR refer to the control signals [1] . After several trials, the weighting functions are chosen as below:

$$\begin{aligned} W_\phi &= 0.5\frac{s+0.6}{s+0.005} \\ W_r &= 0.5\frac{s+0.6}{s+0.001} \\ W_p &= 0.001 \end{aligned}$$

while

$$\begin{aligned} W_{FDA} &= 0.2\frac{s+0.005}{s+20} \\ W_{FDR} &= 0.1\frac{s+0.005}{s+20} \end{aligned}$$

With those weighting functions, an H_∞ controller is designed which gives the frequency and time responses of the closed-loop system as shown in Figure 5.4 and Figure 5.5.

[1] For definitions of symbols, refer to Appendix-C

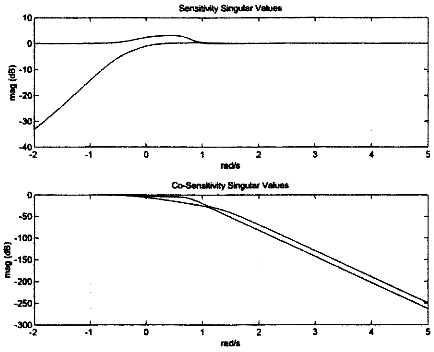


Figure 5.4: Lateral Frequency Response

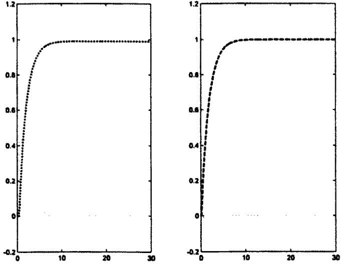


Figure 5.5: Lateral Unit Step Response

The frequency response shows good robustness to unmodelled or poorly modelled dynamics at high frequencies plus good disturbance rejection. The time response shows a well damped case and the rise time is about 3 seconds which is acceptable.

5.1.3 Bell-205 Longitudinal Controller

Similarly, the design for the longitudinal dynamics is carried out. The longitudinal dynamics is a single input and multiple output system (SIMO).The input is the longitudinal cyclic, δ_{ls}. The outputs are the pitch attitude, θ, and pitch rate, q. The linear model is with the following parameters:

$$A = \begin{pmatrix} 0 & 0.9999 & 0 & 0 & 0 \\ 0 & -0.7971 & 0.0018 & -0.0038 & 5.0900 \\ -0.0002 & -2.5715 & -0.0502 & -0.0494 & 3.2565 \\ 0.0173 & 79.7602 & -0.2338 & -0.5458 & -6.9670 \\ 0 & 0 & 0 & 0 & -12.5786 \end{pmatrix}$$

$$\mathrm{B} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -2.6498 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad D = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

The weighting functions selection follows the same guidelines used in the lateral controller case. Again, after successive trials, the weighting functions used are given by the following:

$$W_1 = \begin{bmatrix} W_\theta & 0 \\ 0 & W_q \end{bmatrix}$$

$$\begin{aligned} W_\theta &= 0.5\frac{1}{s+0.001} \\ W_q &= \frac{s}{s+0.001} \end{aligned} \tag{5.4}$$

while

$$W_2 = 40\frac{s+.001}{s+5}$$

The time response is as shown in the diagram below (Figure 5.7) and the frequency response is depicted in Figure 5.6.

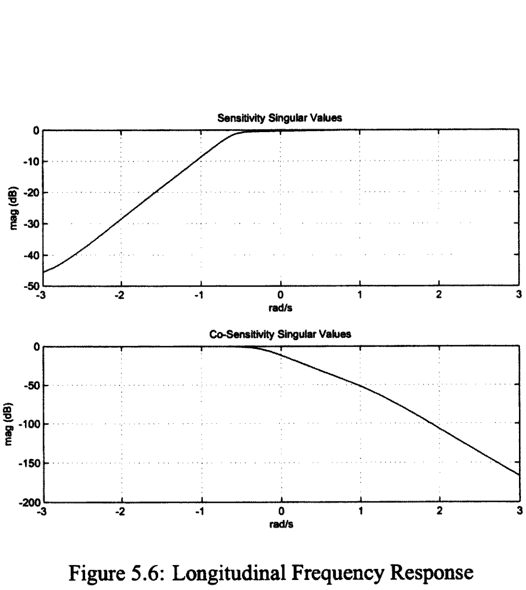


Figure 5.6: Longitudinal Frequency Response

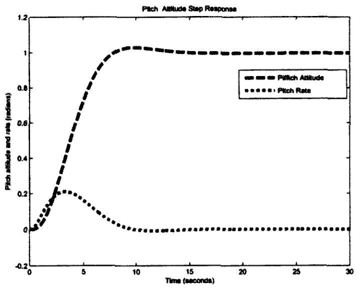


Figure 5.7: Longitudinal Unit Step Response

5.2 Integrating Controllers with the FDI

The aforementioned controllers have been integrated with the respective FDIA subsystems that are discussed in Chapter 4. This results in two decoupled Fault-Tolerant Flight Control Systems (FTFCS). The integrated systems are first described and, in this section, the simulations results are reported afterwards.

5.2.1 Lateral Case

The lateral controller described in the previous section and the ANN-based FDI system described in Section 4.5.1 are integrated as shown in the Simulink diagram below (Figure 5.8) which forms the lateral fault-tolerant flight control system. Partial details of the attained results were reported in [1].

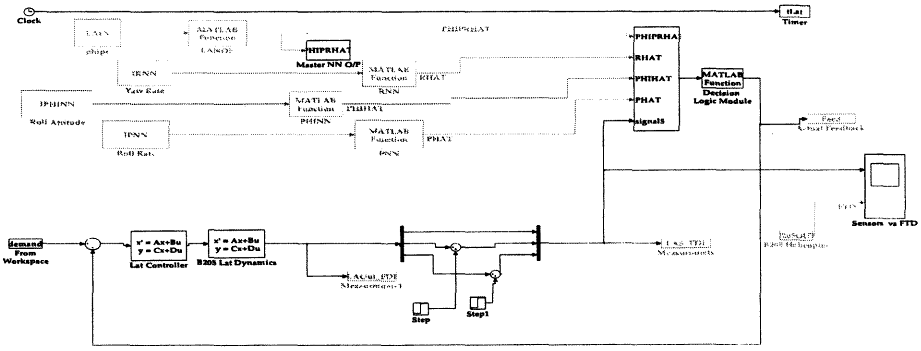


Figure 5.8: Lateral FTFCS for Bell-205 Helicopter

With reference to Figure 5.8, the helicopter's measured outputs (i.e. the feedback signals) are replaced by the estimated outputs from the ANN FDI scheme once a decision is made that a failure has occurred at the corresponding sensor. Such failures could be a sensor stuck at a fixed level or a deviation from healthy output values, or other cases ([2, 3]).

In the simulation, the above system has been tested with two faults. The first one (roll rate) is a step of size=2 that happens at t=0.5 second while the second fault (yaw rate) is a step of size 1.5 that happens at t=1.0 second. Without the FDIA, the outputs are as shown in the right-hand-side part in Figure 5.9. However, with successful implementation of fault detection, isolation, and accommodation, the outputs of the system are shown in the left-hand-side part which are the same as in the fault-free case. We now further discuss a few issues concerning the closed-loop system with the FTFCS.

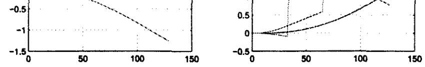


Figure 5.9: Successful FDIA with the H_∞ Controller in the loop

In model-based techniques, it is necessary to anticipate all types of possible faults and include their signatures in the design. This is not the case with the ANN-based technique because it is a different kind of system identification approach. Any discrepancy between the neural model and the physical plant will be observed and appropriate actions are initiated. Of course, not all discrepancies are signs of faults as there may be false alarms or intermittent problems. These will to be taken care of by the fault accommodation module. This is the fundamental feature of an ANN-based FDI that we demonstrated via various simulations in the previous chapter (Chapter 4). However, when an ANN-based FDI (originally designed based on real-time flight test data) is integrated with a linear model it has shown certain discrepancies between the actual outputs and the expected outputs. Tracing these discrepancies back to their possible roots show that they may be attributed to one or more of the following:

- FDI system itself.
- the controller.
- the linear model.

Chapter 4 has proved that ANN-based FDI is accurate enough and the testing was made against Flight Test Data (FTD). The controller was designed according to standard robust control methods and its resulting time and frequency responses are highly acceptable though they may be further improved by manipulating the design knobs (the weighting functions). What is remaining is the last factor; i.e. the linear model. The input that activates the FDI system and the linear model is actually a subset of the FTD. Activating the linear model with a subset of the FTD does not necessarily produce the outputs that are close to the output data collected during flight test. This conclusion is supported by evidence from comparative analysis conducted earlier that shows a certain discrepancy between the linear and nonlinear models on the one hand and the flight test data on the other hand. Discrepancy between the linear models (especially the NASA models) and the flight test data was much more. Details are in [14].

The plots shown in (Figure 5.10) are produced to demonstrate some of the discrepancies between the linear model and the flight test data (FTD). An input signal has been selected from the FTD and used to excite the linear model and the outputs of the linear model are compared against the corresponding signals recorded earlier in the FTD. The first plot in the figure shows a sign convention difference between the two and a scaling as well. The second plot in the same figure shows a scaling difference plus the noise in the FTD.

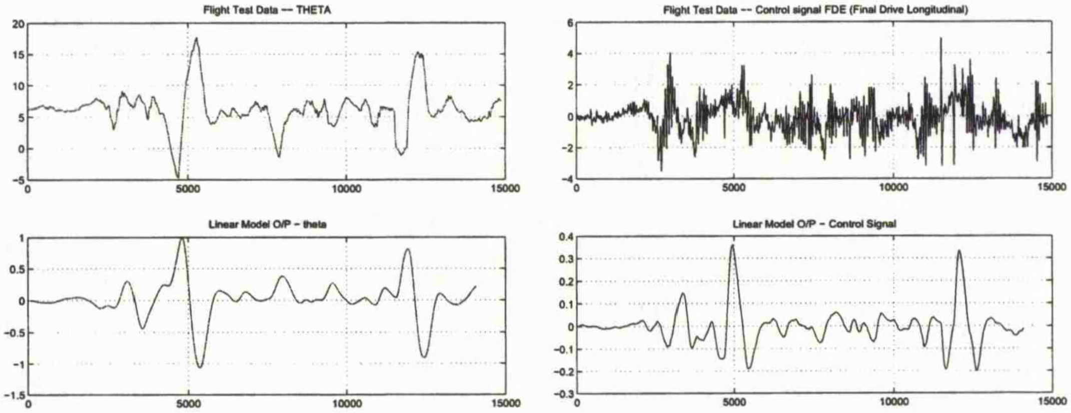
With reference to [23] (particularly Figure 2), the diagram below (Figure 5.10) that has been explained above, and discussion in [78], the following observations are made on the causes of the discrepancies between the linear model and the FTD:

Sign Convention There is a sign convention in the longitudinal axis. This is not documented and must be observed when comparing the outputs of the FTFCS and the FTD.

Zero-Line Flight Test Data sets are not necessarily recorded around the zero line. This does not create any problem as far as the ANN-based FDI training and testing is concerned. However, when the FDI and the model (whether it is linear or nonlinear) are excited with some inputs form FTD, the outputs of both systems will differ greatly due to this undocumented issue.

Potentiometers Prior to flight, the pilot adjusts a set of potentiometers according to his own preferences. This may also occur during flight when for example the pilot may desire to perform some aggressive manoeuvres and for that purpose he adjusts certain potentiometers to respond for very small stick movements. These adjustments are not recorded in the FTD and do create different relationship between the inputs and the outputs. Sometimes, it is a pure gain and in some cases it is a gain and a shift and, moreover, these variations differ from one flight test to another.

The above points have highlighted that during forthcoming flight testing activities, the relevant documentation is extremely important to facilitate future fault-tolerance designs.



(a) Linear Model output (Theta) vs. Flight Test Data

(b) Linear Model Control Signal vs. Flight Test Data

Figure 5.10: Comparison of Linear Model and Flight Test Data

5.2.2 Longitudinal Case

The longitudinal mixed-sensitivity controller and FDI system are integrated as shown in the simulink diagram below (Figure 5.11):

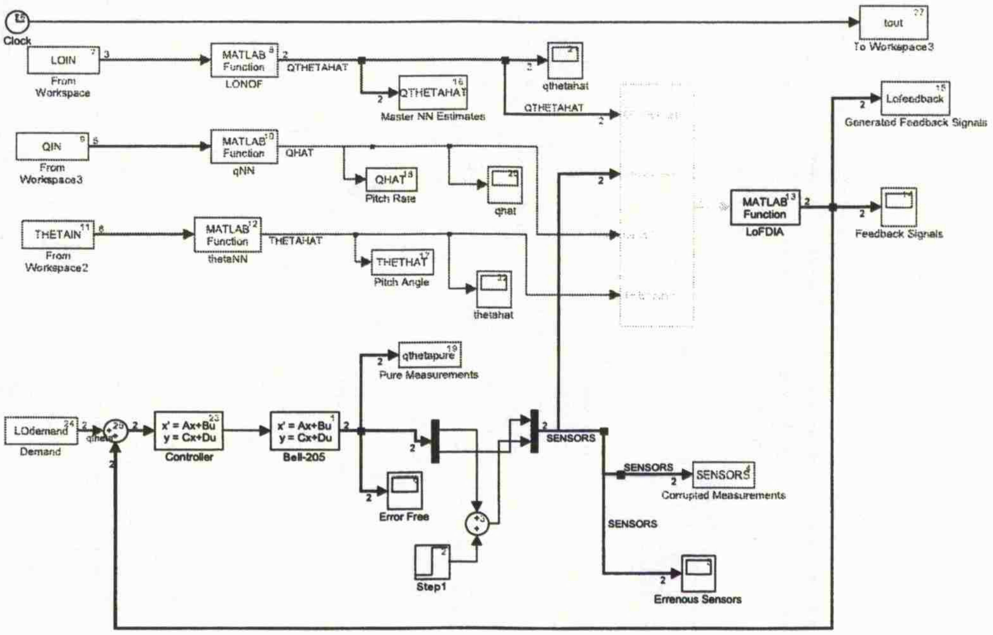


Figure 5.11: Longitudinal FTFCS for Bell-205 Helicopter

The intention was to test the longitudinal FTFCS in the same manner as the lateral FTFCS. However, during the simulation it was noted that significant discrepancies exist between the feedback signals and the expected outputs. Investigation has taken place to identify the source of discrepancy. The subsystems have been tested separately and the results are as follow. The closed loop system with linear model (Figure 5.12) was tested against flight test data where pitch attitude demand (DENET) and pitch rate demand (zero value) were used as input and the outputs were compared against the flight test data (θ and q). The result of activating the linear model with inputs from FTD shows that the linear model output (pitch attitude) differs from the output recorded in the FTD although the linear model output shows that the output is tracking the input demand. From this observation it may be concluded that the discrepancy is mainly due to FTD where sign difference and scaling are the main issues. These discrepancies are clearly spotted from the second and third plots in (Figure 5.13).

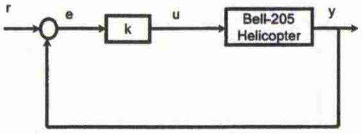


Figure 5.12: Closed Loop System without ANN-based FDIA

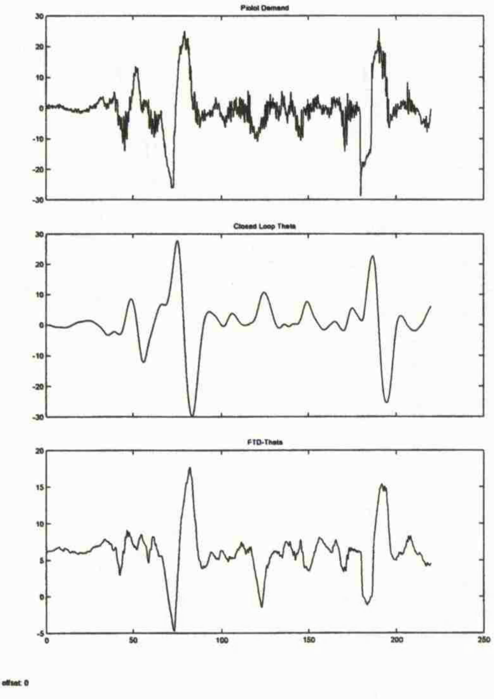


Figure 5.13: Simulation Results of Closed loop without ANN-Based FDIA

The longitudinal ANN-based FDIA was previously tested in detail and results were satisfactory with respect to FTD. As the linear model is in disagreement with the FTD, the ANN-based FDI output will definitely disagree with linear model outputs. As documented in Chapter 4, the FDIA was designed using flight test data which is a true representation of the helicopter. Moreover, the ANN-based FDIA was validated with different data sets and tested with another flight data set and the results were quite acceptable with comments as indicated therein. This confirms that testing the integrated system with linear models in place (plus the comments on FTD mentioned before in last section) is the source of discrepancy. This is understood in the light of the clarifications given in the previous section on lateral FTFCS testing results. The FDIA uses various measurements as input and these measurements are collected from actual flight test data and the only way to get the same or similar values is either from flight test data or from the nonlinear model which is very much closer to flight test data. Thus, one has to replace the linear model with either the nonlinear model or use flight test data. We have used flight test data before to represent the helicopter outputs in testing the FDIA and simulating various faults and the results as shown in Chapter 4. Of course the controller was not used at that stage. This calls for testing the closed loop system with nonlinear model in place. Unfortunately, the nonlinear model is not available for this research.

To conclude this matter, we have tried another approach; i.e. to develop an ANN-based FDIA using linear models inputs and outputs of the system shown above in Figure 5.12. It must be understood that the Linear ANN-based FDI is not as capable as the one based on the flight test data or another one designed using nonlinear model. However, the ultimate goal was to demonstrate the functionality of the FTFCS concept.

5.3 ANN-based FDI using Linear Model Data

The data for this system has been generated by using flight test data input θ_{dem} to generate the outputs $\theta, \dot{q}$ and also the output of the controller was recorded during the process to generate another variable, for training purposes, $Ksignal$ (which is equivalent to longitudinal cyclic FDE).

A two-stage ANN-based FDIA was designed for the longitudinal dynamics. The signals available for training are shown in Table 5.1:

Table 5.1: Input Space for ANN-based FDIA using Linear Model Data

Variable	Meaning
Ksignal	Longitudinal Controller output (dimension=1)
thetaDemand	Pilot Demand on Pitch
qDemand	Pilot Demand on Pitch Rate (it is a zero feedback signal)

The outputs for the MIMO ANN are the pitch attitude (θ) and pitch rate (q). For the MISO ANNs, one of them has pitch attitude as an output and the other has pitch rate as an output. Both of the ANNs use the same input as the master ANN does which are shown in the table above (Table- 5.1). It is obvious that the input variables are limited in number and, thus, the networks are not guaranteed to perform well in all situations.

The results of the training of the MIMO ANN are shown in Figure 5.14 below. From the figures below, one can easily see that the master network is not predicting the pitch attitude accurately but this should not be a problem as the master neural network is used for fault detection. If the fault is not confirmed by the individual isolating networks then the fault alarm will be cleared. This, of course, is not the original intention but for the purpose of testing, it has been made like so.

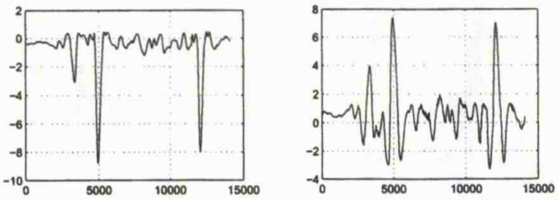


Figure 5.14: Master ANN (LONOF) Outputs After Training vs FTD

The pitch attitude neural network output is shown in Figure 5.15:

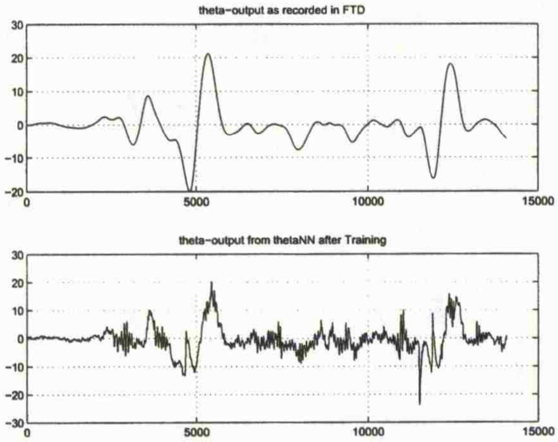


Figure 5.15: Pitch Attitude ANN (thetaANN) Output After Training vs. FTD

and the pitch rate network output is shown in Figure 5.16 below:

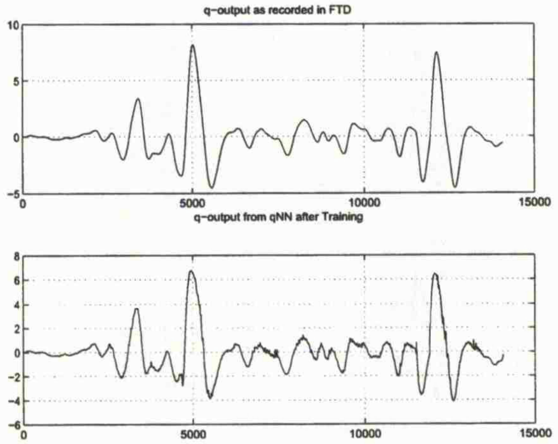


Figure 5.16: Pitch Rate ANN (qNN) Output vs FTD

The isolating networks (thetaANN and qNN) have acceptable outputs which help to overcome the master neural networks capability. These networks have been integrated with the closed loop control system that uses the DERA linear model and the results are presented hereafter.

5.3.1 Limited Version of FTFCS Simulation Results

As mentioned above, the limited version of the ANN-based FDIA have been integrated with the closed loop control system by breaking the feedback loop and connecting the output of the decision logic to the summing junction of the feedback loop. *The main objective is to show that the proposed architecture is functioning as anticipated.* The overall system is similar to the one shown in Figure 5.8. The first simulation result is shown in Figure 5.17 below (Note: the x-axis unit is sample where 1 sec=64 samples). A fault with random signal is injected in the pitch channel. The third pane of the plot shows this faulty signal. The linear model output which is the reference healthy signal (this is shown in the first pane of the figure). The sensor readings after accommodation (second pane of the figure) are compared against the linear model data and as shown in the figure they are almost identical. Though the values are so small to compare but the shape of the two signals and their range confirm the agreement between the two.

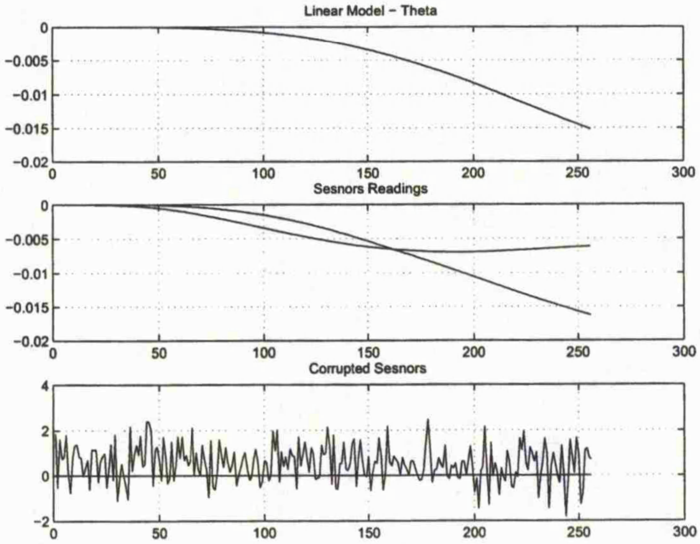
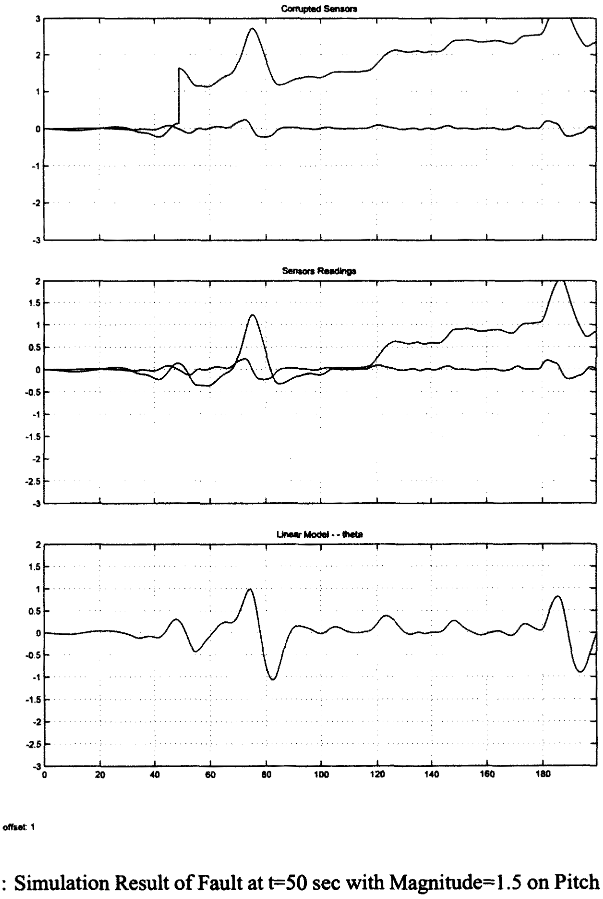


Figure 5.17: Simulation Result of Fault at t=0 sec with Random Magnitude on pitch channel

In the following diagram (Figure 5.18), the simulation is carried out over a larger time span where the fault is simulated at time t=50 seconds. The fault was isolated but due to the accuracy of the isolating network, the sensor reading is different form the linear model data from the time t=80 seconds and onwards.



Time offset: 1

Figure 5.18: Simulation Result of Fault at t=50 sec with Magnitude=1.5 on Pitch Attitude Channel

With the simulation carried out with fault on pitch rate channel, the result is good enough as shown in the diagram below (Figure5.19) where the sensor reading (second pane of the figure) is in agreement with the linear model data (third pane).

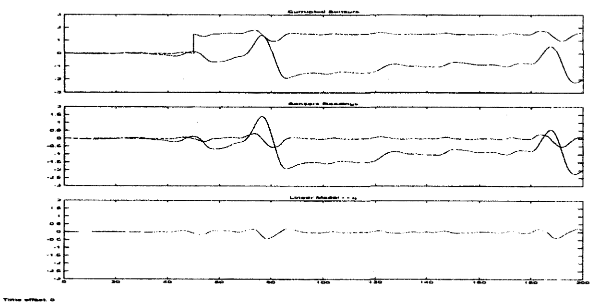


Figure 5.19: Simulation Result of Fault at t=50 sec with Magnitude=1.5 on Pitch Rate channel.

5.4 Performance and Robustness Analysis

As described earlier, the lateral channel FDIA is composed of four ANNs. Each ANN is of multilayer structure, and each layer is with several neurons. Furthermore, every ANN has a Time-Delay Line (TDL) between its inputs and outputs in order to properly match the system dynamics. All the above leads to some computational delays which happen in the implementation of the scheme. The delay is quite natural as ANNs in general are parallel in nature while the computation is carried out on sequential machines. To overcome the delay, it is recommended that computational machines with parallel processors which are suitable for neural networks computations are used. These are available nowadays and their cost is not high relative to their potential benefits in some applications.

For the original closed loop system, the plant (i.e. the Bell-205 Helicopter) is not stable but the closed loop is stabilized by the controller (K) which was designed using the H_∞ mixed-sensitivity approach. After integrating the FDIA with the controller, the overall system is represented by Figure 5.20.

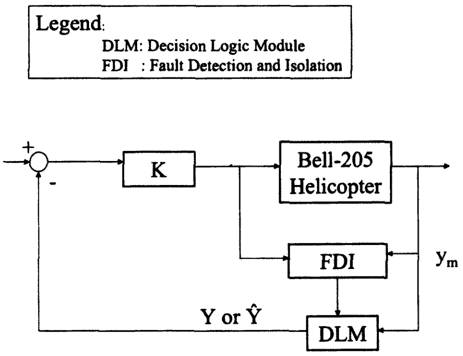

Legend:
DLM: Decision Logic Module
FDI : Fault Detection and Isolation
+
-
K
Bell-205
Helicopter
FDI
DLM
Y or Ŷ
y_m

Figure 5.20: Fault-Tolerant Control System - Fixed Controller Case

In Figure 5.20, the FDIA breaks the feedback loop and receives the plant measured outputs (y_m). The ANNs produce estimates for the three outputs $(\hat{\phi}, \hat{p}, \text{and } \hat{r})$. The ANN-based FDIA checks to find out if a single or multiple fault occurs. If that happens, the FDIA would replace the faulty value(s) by the estimate(s). In that case, a stability problem of the closed-loop system may arise, due to the discrepancy between the estimated outputs and the (healthy) measured outputs. It could be argued that this discrepancy has been taken care of in the design of the controller since the sensitivity function minimization is included which may be interpreted as consideration on the output disturbance. However, since we have the estimation of error bounds for ANNs, we may analyze the stability more explicitly. Given the original closed loop system, the plant being stabilized using the H_∞ mixed-sensitivity controller K, we need to show if the closed loop in Figure 5.21, which is a re-draw of Figure 5.20, is still stable with the same controller.

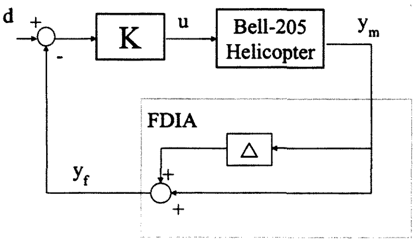


Figure 5.21: Closed-Loop with FDIA Model

In Figure 5.21, the FDIA subsystem is enclosed by a dashed box. The output of the FDIA subsystem is thus represented by:

$$y_f = (I + \Delta)y_m \tag{5.5}$$

where y_f is the FDIA output and y_m is the sensor measurements (the roll attitude ϕ, the roll rate p and the yaw rate r). The transfer function $T_{\Delta io}$, from the output of Δ to its input, is:

$$T_{\Delta_{oi}} = (I + GK)^{-1}GK \tag{5.6}$$

where G represents the Bell-205 helicopter. The closed-loop system can be re-drawn as in Figure 5.22 for robust stability analysis. From the well-known Small Gain Theorem, the closed loop remains stable if the following holds for all possible Δ:

$$\|T_{\Delta_{oi}}\Delta\|_\infty < 1$$

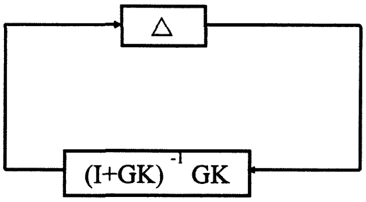


Figure 5.22: Closed-Loop with FDIA Model - SGT view

However, the norm calculation shows that

$$0.99827 \leq \|T_{\Delta_{io}}\|_\infty \leq 0.99927$$

While as the perturbation block Δ has a norm of 1.141, the above reveals clearly that this designed controller does not give robust stability of the closed-loop system. In reality, it means that for some sensor faults the closed-loop system may become unstable if the measured output is replaced by estimated signals from the ANNs. The above consideration is, however, under the assumption that the uncertainty Δ is *unstructured.* In our case, the uncertainty block actually represents the possible error at output channels between healthy sensor measurements and estimations generated by ANNs, and is thus highly *structured.* It is therefore much more appropriate to discuss robust stability of the system using μ-analysis. It turns out that the bound of μ value of the corresponding interconnected system, over the frequency range of $10^{-3}rad/s$ to $10^{3}rad/s$, is less than 0.61 (see Figure 5.23). That result means the robust stability has been indeed attained. The highly structured uncertainty in this system leaves scope for further improvement of controller design. The μ-synthesis would be an obvious choice. The controller design using the μ-synthesis is the subject of next chapter.

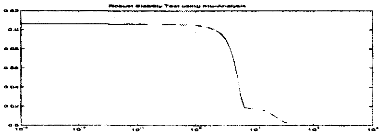


Figure 5.23: μ Analysis Robust Stability Output

Before leaving this chapter, we must point out that the estimation error is high in one of the channels. That is the result of the ANN being trained with the available, limited flight data set. Should a better flight data set be obtained and used for the training, the result could differ significantly.

5.5 Conclusion

We have in this chapter presented the design of robust controllers for the decoupled dynamics and integrated them with the ANN-based FDIA. Due to inconsistency imposed by using Flight Test Data (FTD) with linear models, the lateral and longitudinal systems have shown great discrepancies. Analysis and discussion with participants in flight testing has led to the conclusion that linear model outputs are not in line with the FTD. As FTD is used to activate the predictions generated by the neural networks, the output of the linear model and FDI did not tally with each other. To confirm this analysis, a *limited version* of the ANN-based FDIA was designed and integrated in the closed loop control system that uses the DERA linear model. The outcome has confirmed that the proposed architecture is working. This calls for testing the original system with the DERA Nonlinear Model in place. This will help to fine tune the design parameters of the system and will highlight the enhancements required. In addition to the experience gained during the FTFCS design and simulation, certain remarks have been made about the FTD and the required preparations needed to carry forward further FTFCS research activities.

Chapter 6

Fault-Tolerant Controller using μ Analysis and Synthesis

We have in the previous chapter presented the design of H_∞ controllers for lateral and longitudinal dynamics and shown the simulation results of the integration of those controllers with the respective ANN-based FDIAs. The plan for this chapter is to look into what μ synthesis can contribute to Fault Tolerant Flight Control System (FTFCS). This is the main theme of this chapter. The chapter is organized as follows. First, a brief introduction to the subject of μ analysis and synthesis is given. Afterwards, the design of a μ controller for Bell-205 is presented. Comparison of the controller against the H_∞ controller is presented. Then the μ controller is integrated with the ANN-based FDI. As the goal of this chapter is to explore any additional benefits to be gained by using the μ approach and in view of the previous chapter results, the design and implementation will be limited to the lateral FTFCS.

6.1 Introduction

The starting point in most control system designs is a mathematical model. However, models for plants are never perfect. There are always some dynamics neglected especially in the high frequency range or due to lack of knowledge of the plant at those frequencies.

Sometimes the lack of knowledge of plant nonlinearities or neglecting actuator dynamics lead to mismatches between the plant and the model. The basic theory of H_∞ control is to design a stabilizing controller that is robust in the face of uncertainty in the model. The design of such a controller normally involves lumping together all the sources of the uncertainties into one block that is normally normalized so that the infinity norm of the uncertainty block satisfies the following relationship:

$$||\Delta||_\infty < 1$$

For this normalized uncertainty to hold true, weighting functions are used to quantify the discrepancy over the desired frequency range. In addition, the transfer function between the error signal z and the exogenous inputs entering the plant (disturbances, noise, ...etc) is minimized to satisfy the following relationship:

$$||T_{zw}||_\infty < \gamma$$

where γ is a pre-defined number. In doing that the structure of the uncertainty is lost or assumed to be a full matrix. Additionally, H_∞ controllers for plants that have highly structured uncertainty may result in a very conservative design. The reason for this conservativeness is due to the fact that not all the uncertainties will occur at the same time and, thus, taking them all into consideration amounts to assuming that they will all take place at the same time. If this assumption does not happen in reality, the designed controller will be conservative. The occurrence of such uncertainties is quite probabilistic.

On the other hand, μ has a proven record in improving the robust stability, nominal performance, and robust performance significantly [42]; i.e. with less conservativeness. In our case, the problem setup includes an FDIA system in the feedback loop which imposes an additional structured uncertainty due to time-delay and the inevitable discrepancy between the actual sensor readings and the estimations generated by the FDIA. This has motivated the use of μ synthesis for the design of controllers.

6.2 μ Synthesis

μ (which stands for structured singular value) was initially used for robustness analysis. Analytical solution for a μ controller does not yet exist. However, numerical procedures such as the D-K iteration have been used to synthesize an optimal controller. Here we will provide the definition of μ and describe how to use it in designing an optimal controller [69].

Consider a plant, P, uncertainty block, Δ, and controller, K, in the general control configuration as depicted in the diagram below (Figure 6.1) which is redrawn in what is known as the M-Δ configuration as given in the diagram (Figure 6.2). From the diagram and using Linear Fractional Transformation (LFT),the interconnection matrix, M, is defined as:

$$M = F_l(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}$$

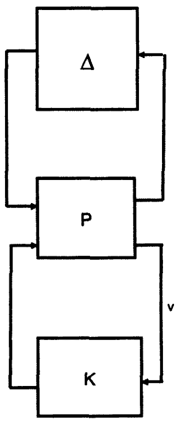


Figure 6.1: General Control Configuration

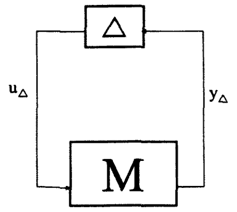


Figure 6.2: M-Δ Configuration

The definition of μ is given by the mathematical formula:

$$\mu(M)^{-1} = \min\{\bar{\sigma}(\Delta) : \Delta \in \Delta, det(I - M\Delta) = 0\} \quad \text{for structured } \Delta \qquad (6.1)$$

which is read as the minimum perturbation that makes the determinant of $(I - M\Delta) = 0$; i.e. makes it singular (thus, unstable). For unstructured (full matrix) perturbation, the smallest perturbation Δ has maximum singular value equal to $1/\bar{\sigma}(M)$. From the properties of μ (see e.g. [75]), rescaling μ can be made without sacrificing stability. The rescaling can be done by pre and post multiplying M with a stable matrix D (that commutes with Δ) and its inverse interchangeably with Δ respectively. Then it is required to find an optimal controller by solving the following problem:

$$\min_K(\min_D \| DMD^{-1}\|_\infty)$$

The system interconnection is first formulated. The interconnection takes into consideration all of the uncertainties in the plant and the interconnection can either be manually done manually, by writing input/output equations, or using the **Matlab** *sysic* command, or using the **Matlab** *linmod* command with a Simulink model. All these approaches lead to the same output as explained in many references (see e.g. [51, 75, 83]). The design of the controller is then carried out by the D-K iteration procedure which involves four steps as follows [51, 75]:

Initialize D Start with an initial value for D which is normally taken as the identity matrix.

K-Step During this step, an H_∞ controller is designed for the interconnected plant which is represented by:

$$F_l(\tilde{P}, K) \quad \text{where} \quad \tilde{P} = DMD^{-1}$$

D-Step During this step fix K and solve the convex optimization problem to find $D(jw)$: $D(jw) = arg\,\inf_D \bar{\sigma}[DF_l(P, K)D^{-1}(jw)]$ over an intended frequency range

Curve Fitting Curve fit $D(jw)$ in order to obtain stable $D(jw)$ and $D^{-1}(jw)$.

The above steps are repeated till the procedure converges to an optimal controller or a given number of iterations is reached. It should be noted that the D-K iteration may not converge in some cases.

6.3 μ Controller Design for Bell-205 Helicopter

As the uncertainty from the discrepancy shown in the ANN outputs is highly structured, the μ tool is a natural choice. We will first present the uncertainties and their modelling and then provide the details of the weighting functions used. The overall design is represented in the following diagram (Figure 6.3)

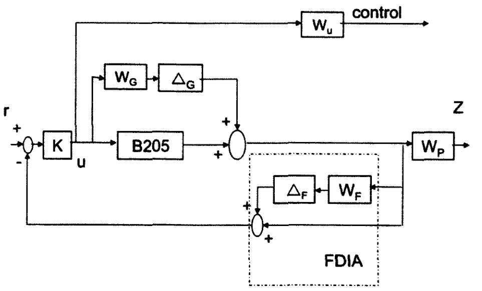


Figure 6.3: Design Problem Formulation Diagram

6.3.1 Uncertainty Modelling

Modelling uncertainties in the controller design framework is very important. Of course including many sources of uncertainties more than necessary, may lead to a conservative design. Here we have considered the most important sources of uncertainties that may, if ignored, result in a controller that may not only lack robustness but also be unstable once put in real operation.

The sources of the uncertainties in the closed loop with FDIA in the feedback loop come from the following:

- Presence of FDIA in the loop. As the FDIA is placed in the closed loop, two implications can be drawn. First, there is some computational delay. Secondly, there is always a difference between the actual signal and the one estimated by the neural model. These differences may have maximum values (upper bounds) and as per the simulations for the lateral dynamics, this is found to be the case for (ϕ, p, r):

$$\begin{pmatrix} 1.141 & 0.0929 & 0.9879 \end{pmatrix}$$

The time delay is modelled using a second order Pade' approximation that is mathematically represented as:

$$\tau = \frac{s^2 - 12s + 48}{s^2 + 12s + 48} \tag{6.2}$$

This may suggest that it can be modelled using parametric uncertainty. Note that the above parameters are the result of the current neural networks using the flight test data. The accuracy of the neural networks can, thus, be improved once the sets of neural networks are trained with flight test data sets collected for this purpose. Especially, the first parameter (1.141) and third parameter (0.9879) are relatively high. Also, the numbers indicate the peak (max) discrepancy that has occurred during the validation of the FDIA. This peak may have been much lower if the training data set was rich enough. Moreover, if the frequency of peak value occurrence is low, then the violation of lowering the limits on those two channels may not be harmful. This matter needs to be further checked once a set of rich training data files are obtained for re-training of the neural networks.

- The model used for the controller design is DERA 6DOF linearized model at the operating condition of 20 knots. As per the comparative study [14], a certain amount of deviation exists between the nonlinear model and the flight test data on the one hand and the linearized DERA model (6DOF and 12DOF) on the other hand. There are cases where the 12DOF model overcomes the deviations as it includes some detailed rotor modelling. The comparison takes place in the time-domain as well as in the frequency domain. As our design is based on frequency domain, the following comments are taken from the above mentioned reference and will be used in the selection of the weighting functions:

  - Yaw: There is an offset that reaches 8dB at 10 rad/s which raises the uncertainty up to 50% at frequencies higher than 2 rad/s. The model is in error for lower frequency.

- Roll: There is an offset across all frequencies (3 dB between 3 and 7 rad/s). The peak uncertainty is 30%.
- Pitch: There is an offset from 3dB at 1 rad/s to approximately +10dB at 10 rad/s. This raises the uncertainty level to 30% at low frequencies and to 50% at higher frequencies.

Due to the existence of the above uncertainties in the various axes, the 6DOF model is considered to be of low/medium fidelity. From the above, we have three perturbation blocks given by:

$$\Delta = \begin{pmatrix} \Delta_P & 0 & 0 \\ 0 & \Delta_G & 0 \\ 0 & 0 & \Delta_F \end{pmatrix}$$

where Δ_G is an unstructured uncertainty block and has the same dimensions as the plant (i.e. 3X2), Δ_F is a structured uncertainty and has a dimension of 3X3 and Δ_P is a block structured uncertainty which is represented by:

$$\Delta_P = \begin{bmatrix} \Delta_U & 0 \\ 0 & \Delta_p \end{bmatrix}$$

where Δ_U is fictitious unstructured uncertainty to penalize the control signal while Δ_p is another fictitious uncertainty to specify the performance requirements.

6.3.2 Weighting function Selection

Although, the information about model uncertainty looks complete, it really does not tell us a lot of the details required. For example, it does not tell us how the uncertainty is related to the input channels and, also, it does not compare the uncertainty magnitudes with the respect to the open loop. The latter is important to convey to the designer whether for example 3dB is a large or small quantity. To conclude, the model uncertainty will be treated as an unstructured uncertainty and the information above will be used to aid in constructing the weighting functions. Accordingly, and in line with the guidelines given in [75], the following weight functions have been selected.

Weights Selection for Unmodelled Dynamics

The following first order transfer function is used [75]:

$$w(s) = \frac{\tau s + r_0}{\frac{\tau}{r_\infty} s + 1} \tag{6.3}$$

where r_0 is the relative uncertainty at steady state, $\frac{1}{\tau}$ is the frequency at which relative uncertainty reaches 100% and r_∞ is the magnitude of the weight at high frequency. Accordingly, the respective weights have been chosen. The weighting function for the unmodelled dynamics W_G is given by:

$$W_G = \begin{pmatrix} w_\phi & 0 & 0 \\ 0 & w_p & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.4}$$

where:

$$w_\phi = \frac{0.33s + 0.3}{0.165s + 1} \tag{6.5}$$

and,

$$w_p = \frac{0.1s + 0.5}{0.05s + 1} \tag{6.6}$$

From our experience with the H_∞ design we have selected the weighting functions W_U to be equal to:

$$W_U = \begin{pmatrix} 0.2\frac{s+0.005}{s+20} & 0 \\ 0 & 0.1\frac{s+0.005}{s+20} \end{pmatrix} \tag{6.7}$$

The other performance weighing function W_p has been selected as follows:

$$W_p = \begin{pmatrix} 0.5\frac{s+1}{s+0.001} & 0 & 0 \\ 0 & \frac{0.5s+0.3}{s+0.001} & 0 \\ 0 & 0 & 10^{-4}\frac{s+0.1}{s+0.001} \end{pmatrix} \tag{6.8}$$

The weighing function to account for the presence of the FDIA, W_F, has been selected as follows:

$$W_F = \begin{pmatrix} 1.141\frac{s^2-12s+48}{s^2+12s+48} & 0 & 0 \\ 0 & 0.0929\frac{s^2-12s+48}{s^2+12s+48} & 0 \\ 0 & 0 & 0.9879\frac{s^2-12s+48}{s^2+12s+48} \end{pmatrix} \tag{6.9}$$

The overall block diagram is shown in Figure 6.4:

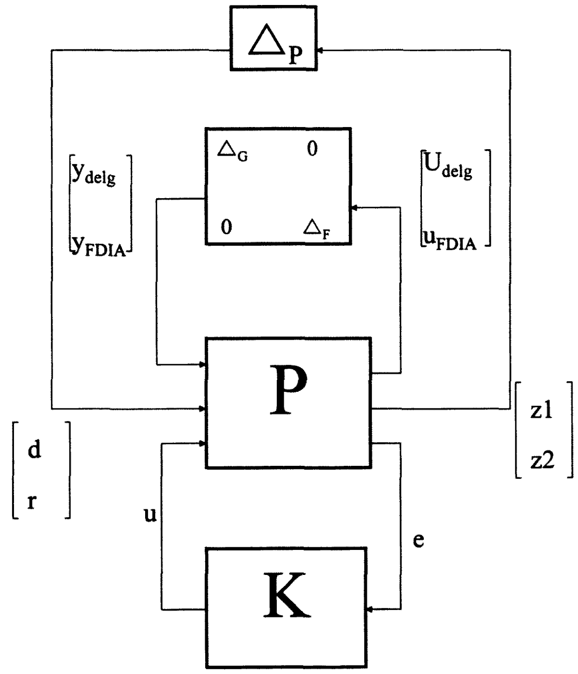


Figure 6.4: General Block Diagram

6.3.3 Performance and Stability Analysis

The controller is designed using the D-K iteration and the following diagram (Figure 6.5) shows the μ values for the designed closed loop system:

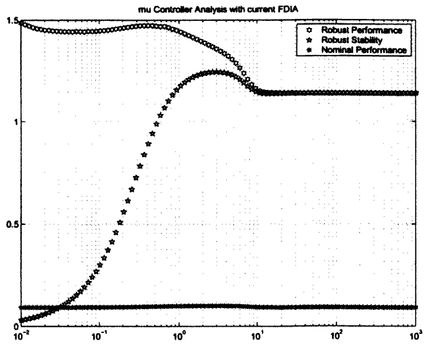


Figure 6.5: μ Controller with all uncertainties Performance and Robustness Analysis Results

It can be seen that the controller attains the nominal performance but fails to satisfy the robust performance and robust stability requirements. As a matter of fact, our mixed-sensitivity H_∞ controller demonstrates lower capability as seen in Figure 6.6.

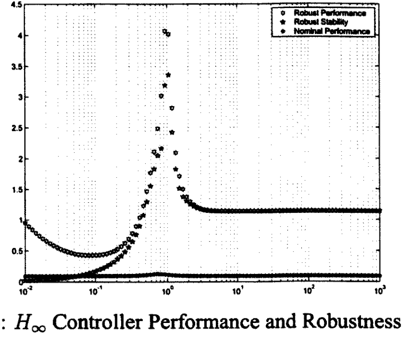


Figure 6.6: H_∞ Controller Performance and Robustness Analysis

From the two diagrams Figure 6.5 and Figure 6.6 we summarize the stability and performance values in the following table:

Table 6.1: H_∞ and μ Controllers Comparison

	H_∞	μ
NP	0.1269	0.0934
RP	3.3613	1.141
RS	4.0693	1.1412

where NP stands for nominal performance, RP stands for robust performance, and RS stands for robust stability. It has to be mentioned that the comparison is made with the same generalized interconnection plant constructed for the design of the μ controller. It is clear that a controller such as the one designed in this chapter using μ-synthesis outperforms the H_∞ controller designed in Chapter 5 as the latter was designed without taking into consideration all the uncertainties which include the presence of the FDIA itself. It is to be noted that the μ controller requires further improvements by conducting more iterations and manipulating the design knobs (the weighting functions). Another avenue for the improvement of the overall performance of the system is to consider the improvement of the FDIA itself as discussed below.

Although the design of the H_∞ controller can be further improved, μ Synthesis provides us with greater flexibility to fine tune the controller without actually manipulating the weighting functions. The designed controller was made for a plant that is augmented with uncertainties pertaining to model uncertainty and FDIA accuracy and delay. Assuming the model uncertainty is accurate enough and must be left intact, the FDIA uncertainty was derived under certain assumptions. This leaves us room to improve the robustness of the controller by improving certain aspects of the FDIA. For example, if the uncertainty in the FDIA is lowered by 50% which is quite possible if the ANN-based FDI is trained with richer data sets, the robustness and performance of the controller is given by Figure 6.7:

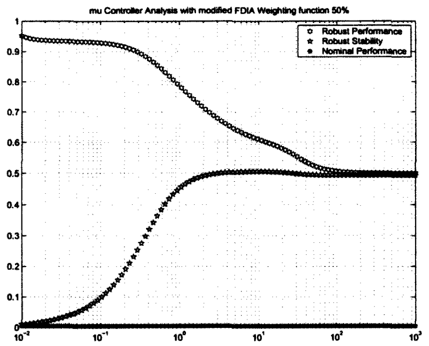


Figure 6.7: Nominal and Robust Performance and Stability with 50% Improvement in FDI)

As we can see, the controller attains nominal performance and robust stability but lacks robust performance. If we leave this discrepancy and consider it unavoidable but consider removing the delay, the controller can attain only nominal performance (same as H_∞ controller) as depicted in the following diagram (Figure 6.8):

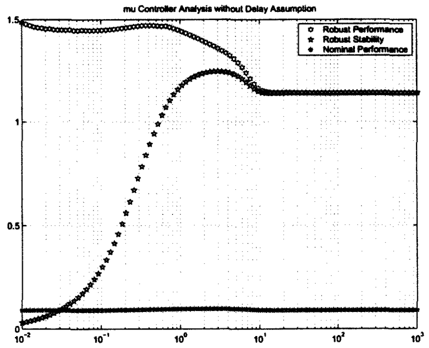


Figure 6.8: Nominal and Robust Performance and Stability with No Delay

If we leave the delay and only improve the discrepancy in the roll attitude channel by retraining the respective ANN, the controller performance is depicted in the following diagram (Figure 6.9) where the robust nominal performance and robust stability is attained with robust performance improved:

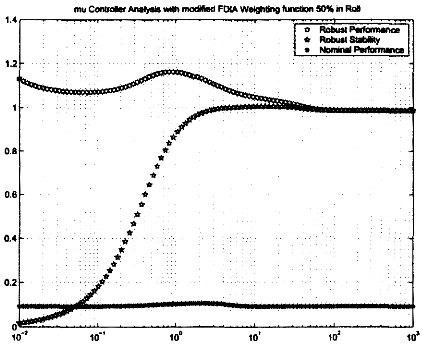


Figure 6.9: Nominal and Robust Performance and Stability with 50% improvement in Roll Channel

To summarize, one can deduce easily that μ synthesis has an advantage over H_∞ where more possibilities are identified to fine tune the controller without really sacrificing the design assumptions.

6.4 Simulation Results

The overall system in Simulink is depicted in Figure 6.10. In the simulation, the same comments in the previous chapter about the discrepancy between the linear model on the one hand and the ANN-based FDI and FTD on the other hand are also valid here. For that reason, a set of an ANN-based FDI using data generated from linear model only has been developed for the purpose of checking the functionality of the system. The neural networks performance is depicted in the following diagrams:

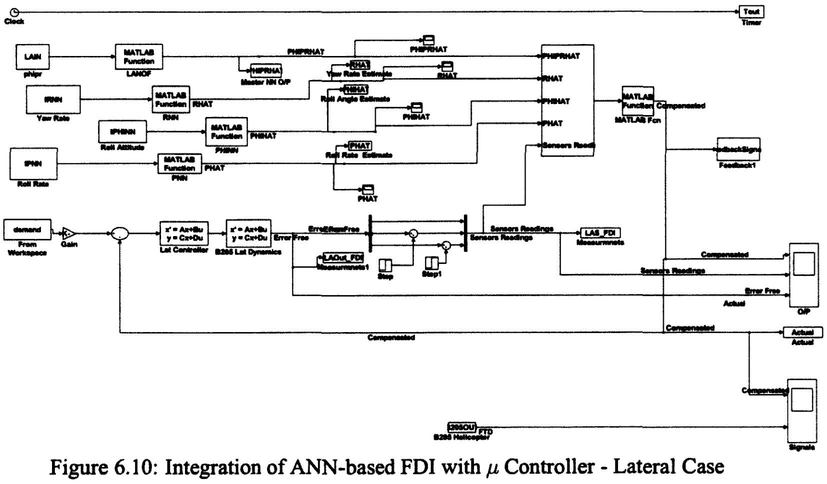


Figure 6.10: Integration of ANN-based FDI with μ Controller - Lateral Case

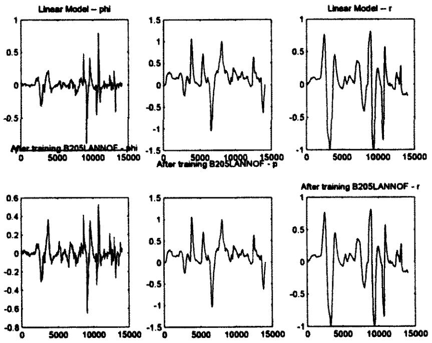


Figure 6.11: Master ANN (LANOF) Performance vs Linear Model

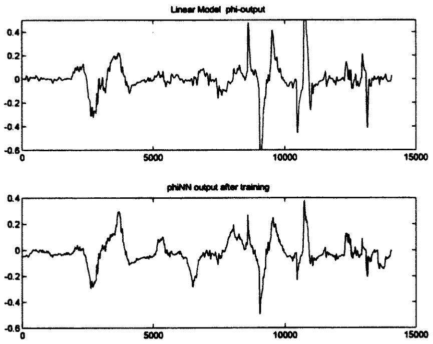


Figure 6.12: Roll Attitude ANN (phiNN) performance vs Linear Model

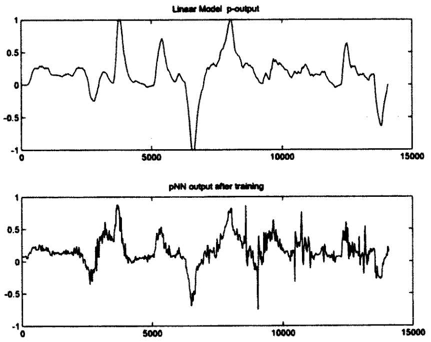


Figure 6.13: Roll Rate ANN (pNN) performance vs Linear Model

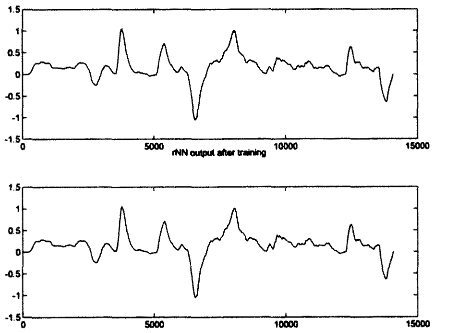


Figure 6.14: Yaw Rate ANN (rNN) performance vs Linear Model

The resultant system (μ-based FTFCS) has been tested and the simulations results are given in Figure 6.15, Figure 6.16 and Figure 6.17 :

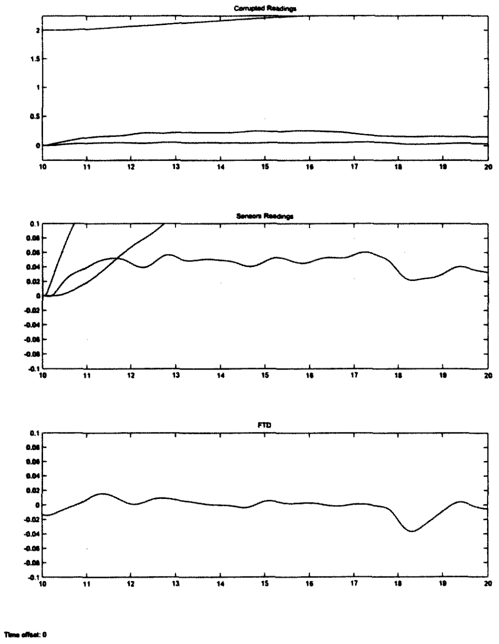


Figure 6.15: Simulation Results of faults occurred at t=9.985 sec with Magnitude=2.0

In the above diagram (Figure 6.15), a fault of magnitude equal to 2.0 at time=9.985 (which is one sample less than t=10 sec) on the roll attitude channel has been simulated and the fault has been cleared and the sensor reading is exactly the same as the FTD used in the linear model.

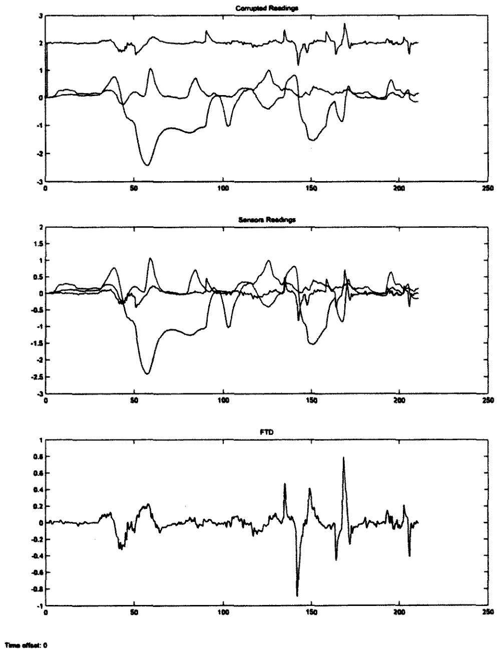


Figure 6.16: Simulation Results of faults occurred at t=1 sec with Magnitude=2.0

In the above diagram (Figure 6.16), a fault of magnitude equals 2.0 at time t=1.0 on roll attitude channel has been simulated and the and the fault has been cleared and the sensor reading is in general similar to the FTD used in the linear model. There are some spikes that make the estimated neural network estimate different from the original linear model FTD reading.

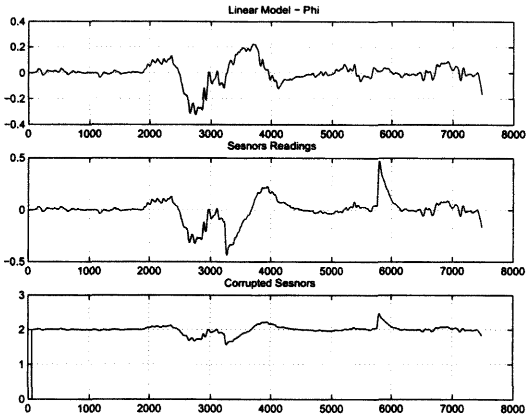


Figure 6.17: Simulation Results of faults occurred at t=10 sec with Magnitude=2.0

Again, the clearance of the fault in the above simulation (Figure 6.17) is quite acceptable with the same comment on the accuracy of the ANN estimate as before. Though, the original healthy signal is small in magnitude ($-0.5 < \phi < 0.5$), the signal has been constructed to recover the original signal.

6.5 Conclusion

In this chapter, we have presented the design of a μ controller for the Bell-205 helicopter (lateral case). As well we have explored the benefit of μ over H_∞ where the controller could be improved without relaxing the uncertainty assumptions. The μ controller with more uncertainty assumptions is better than the H_∞ controller. Improving the FDIA design, which is possible, the revised controller can outperform the H_∞ controller significantly.

The μ controller has been integrated with the ANN-based FDI and the result has proved the functionality of the integrated system though the accuracy has been affected by the fact that the linear model rather than nonlinear model was used in the simulation.

This chapter, and Chapter 5 as well, are mainly on FTFCS using fixed structure controllers. The next chapter, will be on an adaptive approach where the FDIA system will be used to alter the structure of the controller to accommodate the faults.

Chapter 7

ARE-Based Adaptive FTFCS

The aim of this chapter is to explore the theoretical foundation of the application of an adaptive approach to controller reconfiguration in case of failures. The approach is based on updating Algebraic Riccati Equations (ARE) solutions where the impact of the faults on ARE are explored in some detail. The chapter is organized as follows. First, AREs are presented and their link to the synthesis of optimal control systems, in general, and H_∞ in particular is described. Major methods used in solving ARE are outlined in the subsequent sections. Afterwards, a general faulty scenario is considered and the overall controller formulae are derived. Although the approach is applied for sensor faults, similar approaches may be developed for actuator faults or airframe damages. The AREs for the sensor faulty situation are derived and their solutions are outlined. This gives the structure and the governing equations for controller reconfiguration. In line with the research project objectives, the approach is applied to the Bell-205 helicopter with the ultimate goal of designing an adaptive FTFCS for it. At the end of the chapter, some concluding remarks are drawn.

7.1 ARE and H_∞ control

Given a general interconnected plant P as shown in Figure 7.1 (previously seen in Chapter 5) where w stands for exogenous input, z stands for the signal to be penalized, u is the control input, and y is the measurement output.

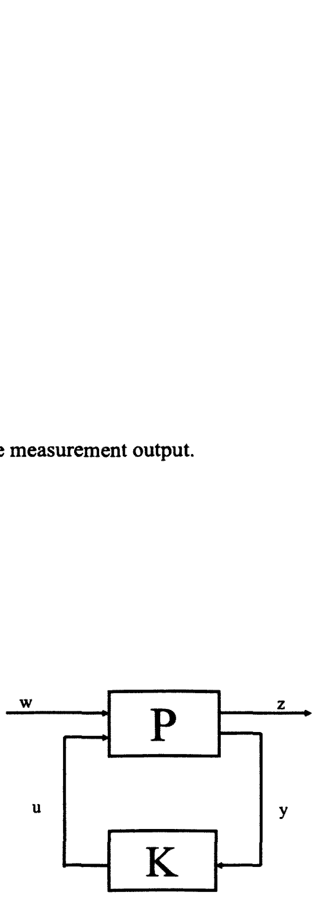


Figure 7.1: The Standard H_∞ Configuration

In the framework of H_∞ control, e.g. S-over-KS design (as discussed in Chapter 5), the controller can be synthesized with the ultimate goal of minimizing the transfer function from w to z (given by the LFT below), or in the suboptimal case, finding a stabilizing controller K such that $\|T_{wz}\|_\infty < \gamma$ where γ is a small positive number:

$$\begin{aligned} T_{wz} &= P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \\ &= \mathcal{F}_l(P, K) \end{aligned} \tag{7.1}$$

where P_{ij} are the partitions of the general interconnected plant P. In the suboptimal case which we consider here, the controller K is given by the following law which involves solutions to two AREs:

$$K = \left[\begin{array}{c|cc} \mathrm{A}_\infty & \text{-}\mathrm{Z}_\infty L_\infty & \mathrm{Z}_\infty B_2 \\ \hline \mathrm{F}_\infty & 0 & \mathrm{I} \\ \text{-}\mathrm{C}_2 & \mathrm{I} & 0 \end{array}\right] \tag{7.2}$$

where:

$$
\begin{aligned}
&Z_\infty = (I - \gamma^2 Y_\infty X_\infty)^{-1},\\
&L_\infty = -Y_\infty C_2^T,\\
&F_\infty = -B_2^T X_\infty,\\
&A_\infty = A + \gamma^{-2} B_1 B_1^T X_\infty + B_2 F_\infty + Z_\infty L_\infty C_2,\\
&X_\infty \geq 0 \qquad \text{is a solution for the following ARE:}\\
&A^T X_\infty + X_\infty A + C_1^T C_1 + X_\infty(\gamma^{-2} B_1 B_1^T - B_2 B_2^T) X_\infty = 0,\\
&Y_\infty \geq 0 \qquad \text{is a solution for the following ARE:}\\
&A Y_\infty + Y_\infty A^T + B_1 B_1^T + Y_\infty(\gamma^{-2} C_1^T C_1 - C_2^T C_2) Y_\infty = 0.
\end{aligned}
$$

As we observe from the above, in the H_∞ control system design, one has to deal with AREs. Our focus is on H_∞ controller design especially for systems where faults are likely. In the following subsequent sections, we will discuss the details of solving AREs.

7.2 ARE Solutions

AREs as the one shown above, may have finite or infinite solutions which can be hermitian, non-hermitian, sign-definite or indefinite. The solution procedures can be based on eigenvalue and invariant subspaces or generalized eigenvalues and deflated subspaces [34, 39, 45, 62]. In the control engineering community, symmetric solutions are of prime interest. Such a solution X with reference to ARE (eq-7.3) below is called the stabilizing solution if $A - RX$ is stable. Major solution procedures are listed below.

7.2.1 Invariant Subspace Method

This method is used to handle AREs of the form:

$$XA + A^T X - XRX + Q = 0 \tag{7.3}$$

The solution is obtained by forming the Hamiltonian matrix:

$$H = \begin{pmatrix} A & -R \\ -Q & -A^* \end{pmatrix}$$

where Q and R are Hermittian matrices (i.e. $Q = Q^*$ where the superscript $*$ denotes complex conjugate transpose). Our focus is on real coefficient matrices and, thus, $Q = Q^T$. The stabilizing solution of the ARE is found by constructing bases for the stable invariant subspace of H. Note that the subspace U is called A-invariant if $AU = U$. The invariant subspaces can be formulated by using the eigenvalues and generalized eigenvectors of H [85]. Assuming that λ_i are eigenvalues with multiplicity k and letting v_i be the corresponding eigenvectors which can be interrelated by:

$$\begin{gathered} (H - \lambda_i I)v_i = 0 \\ (\text{H-} \lambda_i I)v_{i+1} = v_i \\ . \\ . \\ . \\ (\text{H-} \lambda_i I)v_{i+k-1} = v_{i+k-2} \end{gathered}$$

the matrix H is symmetric and has the property:

$$JH = -H^* J$$

where J is defined as:

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

Having formed the Hamiltonian matrix, the solution, X, can be found according to the following steps:

- Compute the matrix of eigenvectors, T:
$$\begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix}^{-1} \begin{pmatrix} A & -R \\ -Q & -A^T \end{pmatrix} \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} = \begin{pmatrix} -\Lambda & 0 \\ 0 & \Lambda \end{pmatrix}$$
where $-\Lambda$ corresponds to left-half plane eigenvalues of H
- The ARE solution, X, is computed by solving a system of linear equations: $X = T_{21}T_{11}^{-1}$

7.2.2 Deflating Subspace Method

In this method, the counterpart to the Hamiltonian matrix is the symplectic matrix which uses the direct inverse of the A matrix. The method is good for AREs with extra matrices such as [39]:

$$\breve{A}^T XE + E^T X\breve{A} - E^T XBR^{-1}B^T E + Q - SR^{-1}S^T = 0$$

Here $\breve{A} = A - BR^{-1}S^T$. As observed, there are some extra matrices which makes the approach of value when handling the AREs corresponding to some faulty scenarios.

The solution is obtained by following the steps below:

- Compute a basis for n-dimensional deflating subspace
- Form the matrix pencil:
$$\begin{pmatrix} A & 0 & B \\ -Q & -A^T & -S \\ S^T & B^T & R \end{pmatrix} - \lambda \begin{pmatrix} E & 0 & 0 \\ 0 & E^T & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- compute the solution by solving a system of linear equations.

7.2.3 Sign Matrix Method

Given a matrix M the sign of this matrix (sgn) is given by:

$$sgn(M) = T\begin{pmatrix} sgn(Re\lambda_1) & ... & 0 \\ 0 & ... & sgn(Re\lambda_q) \end{pmatrix}T$$

where M has a Jordan decomposition $M = T(D+N)T^{-1}$ and $D = diag(\lambda_1, ..., \lambda_q)$ and N is nilpotent and commutes with D.

Now if we consider the ARE:

$$A^TX + XA - XGX + Q = 0$$

where $G = BR^{-1}B^T$ and the Hamiltonian matrix M is given by:

$$M = \begin{pmatrix} A & -G \\ -Q & -A^T \end{pmatrix} \in R^{2n \times 2n}$$

To solve the above ARE using sign-matrix method proceeds as follows:

Let

$$\begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix} = sgn(M)$$

Then, the solution X for the ARE is given by solving the following [39]:

$$\begin{pmatrix} Z_{12} \\ Z_{22}+I \end{pmatrix} X = -\begin{pmatrix} Z_{11}+I \\ Z_{21} \end{pmatrix}$$

7.2.4 Iterative Method

The solution of an ARE can be obtained analytically provided certain conditions are met. They may be solved by iterative, numerical approaches for instance the Newton's method. Newton's method is capable of producing highly accurate results but at the expense of computational cost. On the other hand, Schur's method (used in the invariant subspace method) is one of the most reliable methods. Actually, the best way to solve an ARE is by implementing the generalized Schur method followed by Newton's method. The sign-matrix method may have numerical instability. Details on comparisons of various methods can be found in [33].

7.3 Controller Reconfiguration in FTFCS

The main theme of robust control is to design controllers that are robust enough in the face of uncertainties which could be due to modelling errors or lack of knowledge on plant dynamics especially at high frequencies. However, a robust controller designed via robust design approaches, may still not be able handle plant faults. For example, robust controllers may not handle large sensor faults, actuator faults, or most critical faults pertaining to plant components such as partial damage of an aircraft's airframe. The failure pertaining to a plant's components (e.g. airframe partial damage) affects the model matrices especially the stability derivative matrix, A. Despite this fact, robust control design tools are unable to treat these faults as model uncertainties and further to accommodate them. Thus, it is important to consider other approaches to handle failures when these faults are in the inputs, components, or outputs of the plant. This is the main theme of fault tolerance control (FCT).

Once a fault is persistent and the FDI system marks that sensor as faulty, the controller must be reconfigured to cope with the situation. There may be two approaches to take, namely:

On-Line Design and Switching In this framework, every time a fault occurs, a new controller is designed on-line which involves solving the two AREs mentioned earlier, if the H_∞ design method is used, which at the end leads to the controller K_f. The design should take minimal time. The switching between the current controller and the newly designed one should be smooth and take a small amount of time as well.

Adaptive Compensation In this alternative framework, the original controller is left in action and is compensated for the fault by solving the AREs in terms taking into consideration the difference Δ, between the normal and faulty signal, and generating a compensating signal corresponding to the change.

7.3.1 On-Line Design and Switching

The Linear Time Invariant Finite Dimensional (LTIFD) system on which the controller is acting upon is given by the general equations:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathrm{y} &= \mathrm{Cx} + \mathrm{Du} \end{aligned}$$

In the above state space representation, all matrices are assumed constant and, further without loss of generality, $D = 0$. When a sensor fault occurs, the output changes from $y \rightarrowtail y_f$ where $y_f = y + \Delta y$ and the output matrix may be considered to be perturbed by an amount ΔC. Thus, the new output matrix corresponding to the faulty case is given by:

$$C_f = (I + \Delta)C$$

In this chapter, we consider the H_∞ S/KS (sub)optimization design as an illustrative case. The configuration of this design is depicted in the following diagram (Figure 7.2).

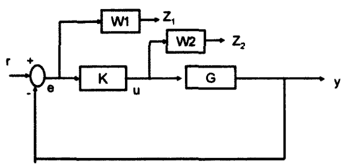


Figure 7.2: H_∞ S/KS Mixed-sensitivity Problem Configuration

The objective to be achieved is:

$$\left\| \begin{matrix} W_1 S \\ W_2 K S \end{matrix} \right\|_\infty \leq \gamma$$

where $S = (I + GK)^{-1}$. In order to show the idea more clearly, we assume without loss of generality that $W_1 = I$ and $W_2 = I$. If the above plant is connected with the controller as in Figure 7.3, the general interconnected plant is given by:

$$P = \left[\begin{array}{c|cc} \mathrm{A} & \mathrm{B}_1 & \mathrm{B}_2 \\ \hline \mathrm{C}_1 & \mathrm{D}_{11} & \mathrm{D}_{12} \\ \mathrm{C}_2 & \mathrm{D}_{21} & \mathrm{D}_{22} \end{array} \right]$$

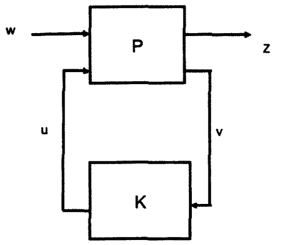


Figure 7.3: General Control Configuration

where:

$B_1 = 0 \qquad B_2 = -B \qquad C_1 = \begin{bmatrix} C \\ 0 \end{bmatrix}$

$C_2 = C \qquad D_{11} = \begin{bmatrix} I \\ 0 \end{bmatrix} \qquad D_{12} = \begin{bmatrix} 0 \\ I \end{bmatrix} \qquad D_{21} = I \qquad D_{22} = 0$

As we have seen in the previous section, the H_∞ controller is obtained by solving a set of two Algebraic Riccati Equations and those AREs are dependent on the plant matrices and among them the output matrix C which now has in the faulty case been replaced by C_f. Definitely, the solution obtained originally does not hold for the new matrices and, thus, the controller may not be able to control the plant properly. Indeed, in some fault situations, a plant may face severe performance degradation and stability problems if no action is taken on time. Thus, it is desirable to get a new controller that once the fault occurs, it takes over instantaneously. As there are various scenarios for the faulty situations, various controllers may need to be designed correspondingly with some switching mechanism. This requires great amounts of knowledge about the faults which then needs to be translated into some mathematical formulation. This on-line approach requires the solution of the new AREs that correspond to the faulty situation. Given the general linear model, the H_∞ controller is obtained by solving the two AREs given by:

$$\begin{aligned} A^T X_\infty + X_\infty A + C_1^T C_1 + X_\infty(\gamma^{-2} B_1 B_1^T - B_2 B_2^T) X_\infty = 0 \\ \mathrm{AY}_\infty + Y_\infty A^T + B_1 B_1^T + Y_\infty(\gamma^{-2} C_1^T C_1 - C_2^T C_2) Y_\infty = 0 \end{aligned}$$

In the S/KS case, the two AREs are:

$$A^T X_\infty + X_\infty A + (1-\gamma^{-2})^{-1} C^T C - X_\infty B B^T X_\infty = 0 \tag{7.4a}$$

$$A Y_\infty + Y_\infty A^T - Y_\infty C^T C Y_\infty = 0 \tag{7.4b}$$

where in case of sensor failure, the equations become:

$$A^T X_\infty + X_\infty A + (1-\gamma^{-2})^{-1} C_f^T C_f - X_\infty B B^T X_\infty = 0 \tag{7.5a}$$

$$A Y_\infty + Y_\infty A^T - Y_\infty C_f^T C_f Y_\infty = 0 \tag{7.5b}$$

With the solution to the above new AREs, a new controller can be formulated to replace the original controller. Should another fault occur, another controller is formulated in view of the fault signature (single or multiple). As such, this approach involves on-line design of the controller and switching between the current controller and the new controller

7.3.2 An Adaptive Compensation

In the previous section we have derived the H_∞ controller that replaces the normal controller. Of course, for the new controller to take over, a switching mechanism is required to smoothly handle the situation. The approach reported in this section is based on exploring the relationship between the formulas which construct the controllers in the normal and the faulty cases.

As discussed before, the controller in the nominal case is

$$K_{normal} = \mathfrak{F}(A, B, C, D, X_\infty, Y_\infty)$$

where X_∞ and Y_∞ are the stabilizing solutions to the two algebraic Riccati equations (AREs), eq- 7.5a and eq- 7.5b, respectively.

On the other hand, in the case of sensor failure, the controller is

$$K_f = \mathfrak{F}(A, B, C_f, D, \tilde{X}_\infty, \tilde{Y}_\infty)$$

where $C_f = (I + \Delta)C$, and $\tilde{X}_\infty$ and $\tilde{Y}_\infty$ are the solutions to the following AREs,

$$A^T\tilde{X}_\infty + \tilde{X}_\infty A + (1 - \gamma^{-2})^{-1}C^T(I + \Delta)^T(I + \Delta)C - \tilde{X}_\infty BB^T\tilde{X}_\infty = 0 \quad (7.6a)$$

$$A\tilde{Y}_\infty + \tilde{Y}_\infty A^T - \tilde{Y}_\infty C^T(I + \Delta)^T(I + \Delta)C\tilde{Y}_\infty = 0 \quad (7.6b)$$

These two new AREs can be considered as slightly perturbed (in the output matrix C) from the original equations. Hence we may reasonably assume that the solutions are closely related by "small" perturbations such as in the forms

$$\tilde{X}_\infty = X_\infty + \Delta_{X_\infty} \quad (7.7)$$

$$\tilde{Y}_\infty = Y_\infty + \Delta_{Y_\infty} \quad (7.8)$$

By substituting $\tilde{X}_\infty$ and $\tilde{Y}_\infty$ into the new AREs we get

$$\begin{aligned} &A^T(X_\infty + \Delta_{X_\infty}) + (X_\infty + \Delta_{X_\infty})A + (1 - \gamma^{-2})^{-1}C^T(I + \Delta)^2C \\ &\quad - (X_\infty + \Delta_{X_\infty})BB^T(X_\infty + \Delta_{X_\infty}) = 0 \end{aligned} \quad (7.9)$$

and

$$\begin{aligned} &A(Y_\infty + \Delta_{Y_\infty}) + (Y_\infty + \Delta_{Y_\infty})A^T \\ &\quad - (Y_\infty + \Delta_{Y_\infty})C^T(I + \Delta)^2C(Y_\infty + \Delta_{Y_\infty}) = 0 \end{aligned} \quad (7.10)$$

We will focus, for the time being, on the first equation. By expanding it, we get

$$
\begin{aligned}
&A^T X_\infty + A^T \Delta_{X_\infty} + X_\infty A + \Delta_{X_\infty} A + (1-\gamma^{-2})^{-1} C^T C \\
&\qquad (1-\gamma^{-2})^{-1} C^T \Delta^2 C + 2(1(1-\gamma^{-2})^{-1}) C^C + (1-\gamma^{-2})^{-1} \Delta^2 C^T C \\
&\qquad - X_\infty B B^T X_\infty - X_\infty B B^T \Delta_{X_\infty} - \Delta_{X_\infty} B B^T X_\infty - \gamma - 2 \Delta_{X_\infty} B B^T \Delta_{X_\infty} = 0
\end{aligned}
\tag{7.11}
$$

After re-arranging the terms, we get the following equation:

$$
\begin{aligned}
&A^T X_\infty + X_\infty A + (1-\gamma^{-2})^{-1}) C^T C - X_\infty B B^T X_\infty \\
&\qquad + A^T \Delta_{X_\infty} + \Delta_{X_\infty} A + ((1-\gamma^{-2})^{-1} C^T \Delta^2 C + 2(1(1-\gamma^{-2})^{-1}) C^C + (1-\gamma^{-2})^{-1} \Delta^2 C^T C) \\
&\qquad - \Delta_{X_\infty} B B^T X_\infty - X_\infty B B^T \Delta_{X_\infty} - \Delta_{X_\infty} B B^T \Delta_{X_\infty} = 0
\end{aligned}
\tag{7.12}
$$

Clearly, in the above equation the first four terms yield zero and the rest form another ARE:

$$
\begin{aligned}
&(A - B B^T X_\infty)^T \Delta_{X\infty} + \Delta_{X_\infty} (A - B B^T X_\infty) \\
&- \Delta_{X_\infty} B B^T \Delta_{X_\infty} + ((1-\gamma^{-2})^{-1} C^T \Delta^2 C + 2(1(1-\gamma^{-2})^{-1}) C^C + (1-\gamma^{-2})^{-1} \Delta^2 C^T C) = 0
\end{aligned}
\tag{7.13}
$$

Dually, the second ARE concerning $\tilde{Y}_\infty$ can be processed in a similar manner and will lead to the following ARE of Δ_{Y_∞}:

$$
\begin{aligned}
&(A - Y_\infty C^T (I+\Delta)^2 C) \Delta_{Y_\infty} + \Delta_{Y_\infty} (A - Y_\infty C^T (I+\Delta)^2 C)^T \\
&- \Delta_{Y_\infty} (C^T (I+\Delta)^2 C) \Delta_{Y_\infty} = 0
\end{aligned}
\tag{7.14}
$$

Hence, we have the following lemma.

Lemma 7.1 *In the case of the sensor failures considered, the ARE solutions $\tilde{X}_\infty$ and $\tilde{Y}_\infty$ may be calculated from*

$$
\tilde{X}_\infty = X_\infty + \Delta_{X_\infty} \tag{7.15}
$$

$$
\tilde{Y}_\infty = Y_\infty + \Delta_{Y_\infty} \tag{7.16}
$$

where Δ_{X_∞} and Δ_{Y_∞} are stabilizing, but not necessarily positive definite, solutions to AREs (eq- 7.13 and eq- 7.14)

Notice that the Riccati equation concerning Y_∞ ($\tilde{Y}_\infty$) can be transformed into a Lyapunov equation, if Y_∞ ($\tilde{Y}_\infty$) is of full rank. That may give numerical advantages in calculation in some cases. However, if the A matrix has stable eigenvalues, the solution to the ARE equation will be rank deficient. A related result is presented in Appendix- D.

Now let us see how the ARE solutions in such a form can be used in the construction of the controller in the sensor failure situation. Recall that the central controller in the present design problem has the following state space realization (notation in the normal case),

$$K_{normal} = \left[\begin{array}{c|c} \mathrm{A - BB}^T X_\infty - (1-\gamma^{-2})^{-1} Z_\infty Y_\infty C^T C & \mathrm{Z}_\infty Y_\infty C^T \\ \hline \mathrm{B}^T X_\infty & 0 \end{array}\right] \qquad (7.17)$$

where

$$Z_\infty = (I - \gamma^{-2} Y_\infty X_\infty)^{-1}.$$

Define

$$\alpha = (1-\gamma^{-2})^{-1/2}$$

and

$$\check{C} = (1-\gamma^{-2})^{-1/2} C = \alpha C$$

and substitute this into Equation 7.17 we get,

$$K_{normal} \;=\; \alpha^{-1} \left[\begin{array}{c|c} \mathrm{A - BB}^T X_\infty - Z_\infty Y_\infty \check{C}^T \check{C} & \mathrm{Z}_\infty Y_\infty \check{C}^T \\ \hline \mathrm{B}^T X_\infty & 0 \end{array}\right] (7.18)$$

Such a controller can be decomposed into three, cascaded parts, by state similarity transformations and system state space model manipulations,

$$K_{normal} = \alpha^{-1}(I+K_1)^{-1} K_2 (I+K_3)^{-1} \qquad (7.19)$$

where,

$$K_1 = \left[\begin{array}{c|c} \text{A} & \text{B} \\ \hline \text{B}^T X_\infty & 0 \end{array}\right] \tag{7.20}$$

$$K_2 = \left[\begin{array}{c|c} \text{A} & \text{Z}_\infty Y_\infty \check{C}^T \\ \hline \text{B}^T X_\infty & 0 \end{array}\right] \tag{7.21}$$

$$K_3 = \left[\begin{array}{c|c} \text{A-BB}^T X_\infty & \text{Z}_\infty Y_\infty \check{C}^T \\ \hline \check{C} & 0 \end{array}\right] \tag{7.22}$$

Similarly, the central controller for the faulty case has the same structure but with:

$$\begin{aligned} \check{C} &\rightarrowtail C_f = (I + \Delta)\check{C} \\ \text{X}_\infty &\rightarrowtail \tilde{X}_\infty = X_\infty + \Delta_{X\infty} \\ \text{Y}_\infty &\rightarrowtail \tilde{Y}_\infty = Y_\infty + \Delta_{Y\infty} \end{aligned}$$

The overall structure of the controller for the faulty case is, thus, depicted in the following diagram (Figure 7.4):

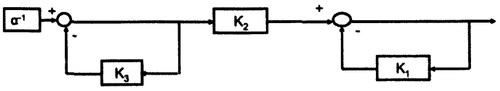


Figure 7.4: Adaptive FTFCS Controller Overall Structure

Using the relationships derived for K_f and K_{normal}, the adaptive compensator for the Bell-205 helicopter can be constructed in the form of the three controllers (K_1, K_2, K_3) with direct substitution of equations (7.20- 7.22) in the above figure. Explicitly, the three components of the controller in the sensor failure case can be depicted as in the following diagrams, Figure(7.5), Figure(7.6) and Figure(7.7), respectively.

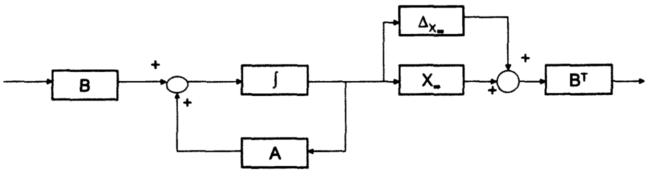


Figure 7.5: Adaptive FTFCS - First Controller (K_1)

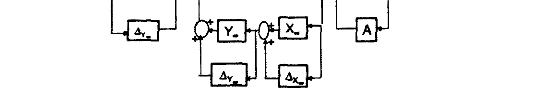


Figure 7.6: Adaptive FTFCS - Second Controller (K_2)

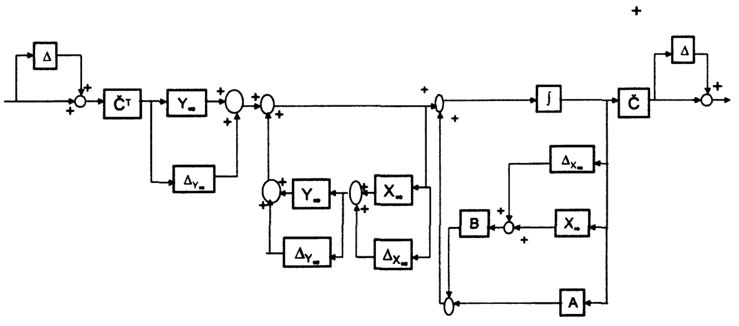


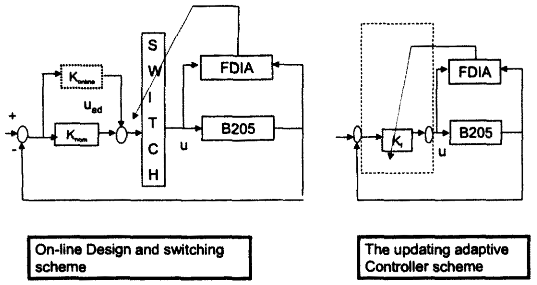
Figure 7.7: Adaptive FTFCS - Third Controller (K_3)

In the above three diagrams, it has to be noted that $\check{C} = \alpha C$. Such a realization of the controller can be updated on line whenever Δ is obtained and Δ_{X_∞} and Δ_{Y_∞} are calculated. A prominent feature is that it may well provide a less bumpy transfer, since only scattered parts are affected and those changes are expected to be small.

Of course, when there is no fault, the controller K_f is simply equivalent to K_{nromal}, because $\Delta = 0$, $\Delta_{X_\infty} = 0$ and $\Delta_{Y_\infty} = 0$. It should be mentioned that the description of a controller as in eq- 7.17 is by no means unique. Various structures are available. The key idea here is to make the updates small in number and to keep the unchanged parts in the controller as many as possible.

7.3.3 Adaptive Scheme Selection

The on-line design and switching scheme and the adaptive scheme are depicted in the following diagram (Figure 7.8).



On-line Design and switching scheme

The updating adaptive Controller scheme

Figure 7.8: Adaptive Schemes

It is recommended that the adaptive compensation approach be used for the following reasons:

- The stringent requirements in regard to minimal on-line design time and smooth switching among controllers may become difficult to adhere to and more difficult to guarantee.
- The original controller which accounts for the healthy situation is intact. The additional part may not have the same negative impact as switching among controllers. It is envisaged that additional compensational signal may not need to be generated for all faulty situations pending on the control logic module. However, this needs to be investigated.
- The approach has potential for practical use. Further investigation may prove that it is not only theoretically appealing but also may lead to some advancement in the design of fault-tolerant flight control system.
- In the adaptive framework, the false alarms and intermittent problems may be easily dealt with without major interruption or performance degradation in contrast to the

case of switching between controllers. The reason is clearly due to the fact that the nominal controller is left intact and the compensating signal may be suppressed smoothly in subsequent cycles .

7.4 Conclusion

We have explored the feasibility of controller reconfiguration based on ARE and developed the mathematical architecture and from that we have extracted the structure of the adaptive updating controller. What is required further is to test the scheme and run various desktop and if possible in-flight simulations to fine tune the system. Additionally, performance and stability is another future opportunity. The pros and cons of the scheme has already compared against the on-line design and switching scheme. Based on the merits of the second approach, a new conceptual reconfigurable controller for Bell-205 helicopter has been developed.

Chapter 8

Conclusion

8.1 Achievements

The theme of the thesis is to investigate the development and the application of the FTFCS concept, as articulated in Chapter 1, for helicopters. To the best of my knowledge this has never been reported in the literature. This may be due to the limited architectural features of helicopters in general. Based on the analysis conducted at the earliest stage of the project, fault-tolerance to sensors failures has been considered with application to the Bell-205 helicopter in this research.

The following achievements have been attained:

ANN-based FDIA for Bell-205 Helicopter: This is a major achievement where two ANN-based FDIA systems have been designed using real flight test data where one of them is for the lateral dynamics and the other one for the longitudinal dynamics. Both systems have been tested in simulation and showed satisfactory performance in terms of the training goals and testing against other flight data. Major features of these two systems are:

- Input space independence among the neural networks.

- All the sensors feedback loops are considered which makes the system of large scale.

H_∞ **Controllers:** Designing controllers was not an objective by itself in the project. However, longitudinal and lateral controllers have been designed for the purpose of being integrated with respective FDIA systems. The controllers have proved on desktop simulations very much satisfactorily in terms of frequency and time domain responses.

μ **Controller:** In the previous designs of H_∞ controllers, the controllers have been designed separately without taking into account the presence of the FDI in the feedback loops. As we see in Chapter 1, the most appropriate design is the one that tackles them together. In Chapter 6 we have designed a controller for lateral dynamics using μ analysis and synthesis where the presence of the FDIA in the feedback loop, in addition to other uncertainties, is included to formulate the structured uncertainty block. The μ tool has helped to identify possible improvements.

ARE-Based Adaptive FTFCS: We have considered in great detail the study of the impact of sensor faults on the Algebraic Riccati Equations (AREs) that are usually solved when designing robust controllers. We have derived in detail the formulae of a new adaptive controller in Chapter 7. The proposed scheme has been compared against on-line reconfiguration and based on the merits of the adaptive scheme, it is recommended as a viable solution. The study forms a solid ground for further research.

8.2 Recommendations For Future Research

The field of FTFCS is expanding rapidly and spreading in many directions. On the other hand most of the effort is on transport or fighter aircrafts but with less degree of concentration on rotory-wing aircraft and very little focus on helicopters. There is almost no reported work on FTFCS development for helicopters. At the same time, there is some effort to develop specifications for FTFCS to cope with military specifications MIL-F-9490D [22]. Thus, it is recommended, in view of intense involvement in helicopter flight control system in Leicester university, that the following activities are to be continued to improve the **implementation of the concept of FTFCS for helicopters**:

Further Verification: The ANN-based systems have shown satisfactory results on desktop simulation. It is urged that additional effort is paid to collect much richer flight test data for further training and validation of the various neural networks. So far, testing and validation are carried out against a limited flight test data. To prepare the system for real flight test, the ANN-based FDIA needs to be tuned by comparing its outputs against the DERA nonlinear model in various operating conditions. As we have seen, integrating ANN-based FDIA with the linear model did not produce acceptable results though the concept has been proved to be functioning with the limited version ANN-based FDI developed for that purpose. Having done that, the next step is to carry out flight test of the ANN-based FDIA by itself and based on these flight tests some modifications and improvements can be made and incorporated into the system. And the integrated system (the FDIA with the Controllers) can be tested again.

Adaptive Scheme: The work we have carried out in Chapter 5 and 6 is based on fixed controllers integrated with the ANN-based FDIA. The theoretical study we have done in Chapter 7 has provided details on developing a new controller that can be implemented for Bell-205 helicopter. It is recommended that the adaptive approach detailed therein, to be fully tested to explore its features.

Fuzzy Logic: The accommodation in our design is coded directly using Matlab script. On the other hand, fuzzy logic with its structural logic may be a better platform to code the faults' accommodation process [68]. This preference is based on the fact that fuzzy logic machinery is greatly geared to capture human knowledge usually expressed in natural language. Thus, it is recommended that this part be considered where instead of using threshold values to take remedial actions, more complex knowledge extracted from the pilot experience be coded for more appropriate decision logic.

Comprehensive Neural Model for Bell-205 helicopter: In line with this research, it is recommended to carry on detailed modelling of Bell-205 helicopter as indicated in Chapter 3 (Figure 3.6) where each subsystem of the helicopter is modelled by itself and the resultant subsystems are integrated together to produce an ANN-based model equivalent to the DERA nonlinear model. The neural models that have been developed for the FDI systems can be used for this purpose.

Other ANN Architectures: ANN has a proven record supported with formal mathematical proofs in the modelling of nonlinear systems. We have used one type of neural networks; i.e. the de facto standard in control community, the MLP. Other researchers claim that other architectures such as radial basis function (RBF) are very powerful. It is recommended that some comparative research be done in this regard.

Appendix A

ANN Details

A.1 Introduction

The aim of this appendix is to give sufficient details on the neural networks used in the design of the ANN-based FDIA system. There are two MIMO ANNs one is used in lateral dynamics and the other in longitudinal dynamics FDIA subsystems. Each one of these ANNs is used to perform the fault detection function by checking the health of the measurements of the feedback loop sensors. Also, there are five MISO ANNs where three of them are used in lateral dynamics and the other two in the longitudinal dynamics case. Each of these MISO ANN is used for fault isolation. The details of all of them as generated by Matlab are given next.

A.2 Lateral Dynamics FDIA Neural Netorks Details

Network Name: B205LANNOF

```
B205LANNOF =
Neural Network object:

architecture:

numInputs: 1
numLayers: 4
biasConnect: [1; 1; 1; 1]
inputConnect: [1; 0; 0; 0]
layerConnect: [4x4 boolean]
outputConnect: [0 0 0 1]
targetConnect: [0 0 0 1]

numOutputs: 1 (read-only)
numTargets: 1 (read-only)
numInputDelays: 3 (read-only)
numLayerDelays: 0 (read-only)

subobject structures:

inputs: 1x1 cell of inputs
layers: 4x1 cell of layers
outputs: 1x4 cell containing 1 output
targets: 1x4 cell containing 1 target
biases: 4x1 cell containing 4 biases
inputWeights: 4x1 cell containing 1 input weight
layerWeights: 4x4 cell containing 3 layer weights

functions:

adaptFcn: 'trains'
```

initFcn: 'initlay'
performFcn: 'mse'
trainFcn: 'traingdx'

parameters:

adaptParam: .passes
initParam: (none)
performParam: (none)
trainParam: .epochs, .goal, .lr, .lr$_dec$,
$.lr_inc, .max_fail, .max_perf_inc, .mc,$
$.min_grad, .show, .time$

$weightandbiasvalues$:

IW : $4x1cellcontaining1inputweightmatrix$
LW : $4x4cellcontaining3layerweightmatrices$
b : $4x1cellcontaining4biasvectors$

$other$:

$userdata$: $(userstuff)$

$NeuralNetworkName = B205PHINN$

$B205PHINN =$
$NeuralNetworkobject$:

$architecture :$

$numInputs : 1$

$numLayers : 4$

$biasConnect : [1; 1; 1; 1]$

$inputConnect : [1; 0; 0; 0]$

$layerConnect : [4x4boolean]$

$outputConnect : [0001]$

$targetConnect : [0001]$

$numOutputs : 1(read - only)$

$numTargets : 1(read - only)$

$numInputDelays : 3(read - only)$

$numLayerDelays : 0(read - only)$

$subobjectstructures :$

$inputs : 1x1cellofinputs$

$layers : 4x1cellof layers$

$outputs : 1x4cellcontaining1output$

$targets : 1x4cellcontaining1target$

$biases : 4x1cellcontaining4biases$

$inputWeights : 4x1cellcontaining1inputweight$

$layerWeights : 4x4cellcontaining3layerweights$

$functions :$

$adaptFcn :' adaptwb'$

$initFcn :' initlay'$

$performFcn :' mse'$

$trainFcn :' traingdx'$

$parameters :$

$adaptParam : .passes$
$initParam : (none)$
$performParam : (none)$
$trainParam : .epochs, .goal, .lr, .lr_dec,$
$.lr_inc, .max_fail, .max_per f_inc, .mc,$
$.min_grad, .show, .time$

$weightandbiasvalues :$

$IW : 4x1cellcontaining1inputweightmatrix$
$LW : 4x4cellcontaining3layerweightmatrices$
$b : 4x1cellcontaining4biasvectors$

$other :$

userdata: (user stuff)

Name of the Network = B205PNN

Neural Network object:

architecture:

numInputs: 1
numLayers: 3
biasConnect: [1; 1; 1]

```
inputConnect: [1; 0; 0]
layerConnect: [0 0 0; 1 0 0; 0 1 0]
outputConnect: [0 0 1]
targetConnect: [0 0 1]

numOutputs: 1 (read-only)
numTargets: 1 (read-only)
numInputDelays: 2 (read-only)
numLayerDelays: 0 (read-only)

subobject structures:

inputs: 1x1 cell of inputs
layers: 3x1 cell of layers
outputs: 1x3 cell containing 1 output
targets: 1x3 cell containing 1 target
biases: 3x1 cell containing 3 biases
inputWeights: 3x1 cell containing 1 input weight
layerWeights: 3x3 cell containing 2 layer weights

functions:

adaptFcn: 'adaptwb'
initFcn: 'initlay'
performFcn: 'mse'
trainFcn: 'traingdx'

parameters:

adaptParam: .passes
```

initParam: (none)
performParam: (none)
trainParam: .epochs, .goal, .lr, .lr$_dec$,
$.lr_inc$, $.max_fail$, $.max_perf_inc$, $.mc$,
$.min_grad$, $.show$, $.time$

$weightandbiasvalues:$
$IW: 3x1cellcontaining1inputweightmatrix$
$LW: 3x3cellcontaining2layerweightmatrices$
$b: 3x1cellcontaining3biasvectors$

$other:$

$userdata: (userstuff)$

$NameoftheNetwork = B205RNN$

$NeuralNetworkobject:$

$architecture:$

$numInputs: 1$
$numLayers: 2$
$biasConnect: [1; 1]$
$inputConnect: [1; 0]$
$layerConnect: [00; 10]$
$outputConnect: [01]$
$targetConnect: [01]$

$numOutputs : 1(read - only)$

$numTargets : 1(read - only)$

$numInputDelays : 3(read - only)$

$numLayerDelays : 0(read - only)$

$subobjectstructures :$

$inputs : 1x1cellofinputs$

$layers : 2x1celloflayers$

$outputs : 1x2cellcontaining1output$

$targets : 1x2cellcontaining1target$

$biases : 2x1cellcontaining2biases$

$inputWeights : 2x1cellcontaining1inputweight$

$layerWeights : 2x2cellcontaining1layerweight$

$functions :$

$adaptFcn :' adaptwb'$

$initFcn :' initlay'$

$performFcn :' msereg'$

$trainFcn :' trainlm'$

$parameters :$

adaptParam: .passes

initParam: (none)

performParam: .ratio

trainParam: .epochs, .goal, .max$_{f}ail, .mem_{r}educ$,

$.min_{g}rad, .mu, .mu_{d}ec, .mu_{i}nc$,

$.mu_{m}ax, .show, .time, .lr$,

$.mc, .lr_inc, .lr_dec$

$weightandbiasvalues:$

$IW: 2x1cellcontaining1inputweightmatrix$
$LW: 2x2cellcontaining1layerweightmatrix$
$b: 2x1cellcontaining2biasvectors$

$other:$
$userdata: (userstuff)$

A.3 Longitudinal Dynamics FDI Neural Networks Details

Name of the Network = B205LONNOF

Neural Network object:

architecture:

numInputs: 1
numLayers: 3
biasConnect: [1; 1; 1]
inputConnect: [1; 0; 0]
layerConnect: [0 0 0; 1 0 0; 0 1 0]
outputConnect: [0 0 1]
targetConnect: [0 0 1]

numOutputs: 1 (read-only)

numTargets: 1 (read-only)

numInputDelays: 4 (read-only)

numLayerDelays: 0 (read-only)

subobject structures:

inputs: 1x1 cell of inputs

layers: 3x1 cell of layers

outputs: 1x3 cell containing 1 output

targets: 1x3 cell containing 1 target

biases: 3x1 cell containing 3 biases

inputWeights: 3x1 cell containing 1 input weight

layerWeights: 3x3 cell containing 2 layer weights

functions:

adaptFcn: 'adaptwb'

initFcn: 'initlay'

performFcn: 'mse'

trainFcn: 'traingdx'

parameters:

adaptParam: .passes

initParam: (none)

performParam: (none)

trainParam: .epochs, .goal, .lr, .lr$_{d}ec$,

$.lr_inc, .max_fail, .max_perf_inc, .mc,$

$.min_grad, .show, .time$

$weightandbiasvalues :$

$IW : 3x1cellcontaining1inputweightmatrix$
$LW : 3x3cellcontaining2layerweightmatrices$
$b : 3x1cellcontaining3biasvectors$

$other :$
$userdata : (userstuff)$

$NameoftheNetwork = B205THETANN$

$NeuralNetworkobject :$

$architecture :$

$numInputs : 1$
$numLayers : 3$
$biasConnect : [1; 1; 1]$
$inputConnect : [1; 0; 0]$
$layerConnect : [000; 100; 010]$
$outputConnect : [001]$
$targetConnect : [001]$

$numOutputs : 1(read - only)$
$numTargets : 1(read - only)$
$numInputDelays : 4(read - only)$
$numLayerDelays : 0(read - only)$

$subobjectstructures :$

$inputs : 1x1cellofinputs$

$layers : 3x1cellof layers$

$outputs : 1x3cellcontaining1output$

$targets : 1x3cellcontaining1target$

$biases : 3x1cellcontaining3biases$

$inputWeights : 3x1cellcontaining1inputweight$

$layerWeights : 3x3cellcontaining2layerweights$

$functions :$

$adaptFcn :' adaptwb'$

$initFcn :' initlay'$

$performFcn :' mse'$

$trainFcn :' traingdx'$

$parameters :$

$adaptParam : .passes$

$initParam : (none)$

$performParam : (none)$

$trainParam : .epochs, .goal, .lr, .lr_{d}ec,$

$.lr_{i}nc, .max_{f}ail, .max_{p}erf_{i}nc, .mc,$

$.min_{g}rad, .show, .time$

$weightandbiasvalues :$

$IW : 3x1cellcontaining1inputweightmatrix$

$LW : 3x3cellcontaining2layerweightmatrices$

$b : 3x1cellcontaining3biasvectors$

other :

```
userdata: (user stuff)


Name of Network = B205QNN

Neural Network object:

architecture:

numInputs: 1
numLayers: 3
biasConnect: [1; 1; 1]
inputConnect: [1; 0; 0]
layerConnect: [0 0 0; 1 0 0; 0 1 0]
outputConnect: [0 0 1]
targetConnect: [0 0 1]

numOutputs: 1 (read-only)
numTargets: 1 (read-only)
numInputDelays: 5 (read-only)
numLayerDelays: 0 (read-only)

subobject structures:

inputs: 1x1 cell of inputs
layers: 3x1 cell of layers
```

outputs: 1x3 cell containing 1 output

targets: 1x3 cell containing 1 target

biases: 3x1 cell containing 3 biases

inputWeights: 3x1 cell containing 1 input weight

layerWeights: 3x3 cell containing 2 layer weights

functions:

adaptFcn: 'adaptwb'
initFcn: 'initlay'
performFcn: 'mse'
trainFcn: 'traingdx'

parameters:

adaptParam: .passes
initParam: (none)
performParam: (none)
trainParam: .epochs, .goal, .lr, .lr$_dec,$
$.lr_inc, .max_fail, .max_perf_inc, .mc,$
$.min_grad, .show, .time$

$weightandbiasvalues:$

$IW: 3x1cellcontaining1inputweightmatrix$
$LW: 3x3cellcontaining2layerweightmatrices$
$b: 3x1cellcontaining3biasvectors$

$other:$

userdata: (user stuff)

Appendix B

Simulation Environment

B.1 Computing Facility

The development and simulation of the ANN-based FDIA and the various controller were developed on a typical desktop machine with the following characteristics for both machines used:

First Machine Intel Pentium III	Second Machine Pentium IV Machine
RAM is 512 MB	RAM is 256 MB

The most critical part is the ANN training as its time and resource consuming especially with time-delay neural networks. Though, it was expected that the activity could have been performed with even less powerful machines. Of course, if the target machine on which the code will be run is a kind of parallel architecture such as the neural chips nowadays available in the market, the execution time will be enhanced. This stage of sophistication may need to be dealt with in case it is decided to deploy an ANN-based solution in a real flight testing mission. ANNs as it is well know are inherently parallel in architecture and more suitable to be used on parallel machines.

B.2 Testing and Validation Files

As indicated earlier in the thesis, in order to build a neural model certain precautions have to be observed. First, the neural networks must be trained with representative data. From our analysis, we gathered that the flight test data was short data sets and collected on different piloted simulations for the purpose of controller assessment analysis. From the analysis and in order to have a coherent design of the ANN-based FDIA and the controllers, we have selected data sets that are closely around the operating point $20 Knots$ as we have initially decided to pick the DERA model around the same operating point. Another important point is that the validation data set should be different from the training data set in order to ensure that the neural network is capable to generalize. These constraints have been considered during our work. Below is a summary of the data sets used for each activity:

Table B.1: Training and Validation Flight Test Data Sets

ANN Name	Training FTD	# of Records	Validation FTD	# of Records	Remarks
LANOF	Y099021.e15	4095	Y99021.e15	4087	The data file has 8192 records and is split into two haves one for training and one for validation
PhiNN	Y01339.e14	5006	Y01347.e15	30413	
pNN	Y01339.e14	5006	Y01339.e12	5006	
rNN	Y01339.e14	5006	Y01339.e12	14867	
LONOF	Y01339.e14	5006	Y99021.e15	4087	
thetaNN	Y01339.e14	5006	Y99021.e15	4087	
qNN	Y01339.e14	5006	Y99021.e15	4087	

After designing the controllers, they have been integrated as shown in Chapter 5. Again,

testing has taken place for the lateral and longitudinal FTFCS. The following data sets have been used:
For Lateral FTFCS: Y01347.e23 (2571 records).
For Longitudinal FTFCS: Y01347.e23 (2571 records).

As we note the data set used during FTFCS testing was different from those used during training or validation.

Appendix C

Flight Test Data (FTD) Details

Introduction

Leicester University has conducted a lot of research on the Bell-205 helicopter. As a result many flight test data sets were collected during the flight test of various controllers that were built in the frameworks of the PhD research projects carried over in [76, 77] and other departmental activities on the said vehicle. The data sets were recorded by the in-flight computer which is limited in storage. Additionally, the test flight data were collected for controller performance analysis. These two facts are very much important to highlight when one intends to use them for ANN training. During the construction and training of ANN for FDIA system, it happened that a certain architecture does not converge. Thus, another architecture and/or input space has been tried. This should not be interpreted as the architecture that failed to converge is not suitable. It could very well converge with another flight data set. This is the first implication of the flight data on the design of ANN-based FDIA. The other implication is less severe. That is to say, with better flight data the performance of the ANN during training and afterward would have been better. It could be possible that with better flight data, the training itself should not take that much of time. Not only this but also the resultant ANN would have been much better in terms of generalization. These comments are necessary to mention should there be any future work to be carried out in this direction. The flight data to be collected must be sufficient enough and cover the typical flight envelope in order to train the ANN properly to capture the nonlinear features of the nonlinear dynamics. The table below highlights the description of each variable as has been conveyed to us by NRC Canada.

Variable	Description
FSW	Function switches
SACLA	Pilot input from side arm controller latitude input
SACLE	Pilot input from side arm controller long input
SACR	Pilot input from side arm controller pedal input
SACP	Pilot input from side arm controller collective input
PHI	Euler angle - Deg
THETA	Euler angle - Deg
P	Angular velocity - Deg/sec
Q	Angular velocity - Deg/sec
R	Angular velocity - Deg/sec
P-MIX	Mixed rates to provide lead on rate measurement.
Q-MIX	Mixed rates to provide lead on rate measurement.
R-MIX	Mixed rates to provide lead on rate measurement.
FDA	final drives to actuators in inches of Safety Piolt - latitude
FDE	final drives to actuators in inches of Safety Piolt - longitude
FDR	final drives to actuators in inches of Safety Piolt - pedal
FDP	final drives to actuators - collective
TAS	True Air Speed
CPA	Cockpit pot A used by pilot to linearly vary any parameter he wants like control system gain or stick sensitivity
CPB	Cockpit pot B - ditto
CPC	Cockpit pot C - ditto
CPE	Cockpit pot E - ditto
CPF	Cockpit potF-ditto
DIFF-THE	Safety system stuff
DIFF-PHI	Safety system stuff
TRIP-SIG	Safety system stuff
CONS-ERR	Safety system stuff

Table C.1: FTD Variables - Continued

DANET	Evaluation Pilot input that has been filtered, scaled and dead banded and used as direct input to the control position. It is created from either DELA or SACLA depending on what input device is selected- latitude demand
DENET	Evaluation Pilot input that has been filtered, scaled and dead banded and used as direct input to the control position. It is created from either DELA or SACLA depending on what input device is selected- long demand
DRNET	Evaluation Pilot input that has been filtered, scaled and dead banded and used as direct input to the control position. It is created from either DELA or SACLA depending on what input device is selected- pedal demand
DPNET	Evaluation Pilot input that has been filtered, scaled and dead banded and used as direct input to the control position. It is created from either DELA or SACLA depending on what input device is selected- collective demand
CPD	Cockpit pot A used by pilot to linearly vary any parameter he wants like control system gain or stick sensitivity
W-MIX	Airmass velocity mixed with accelerometer info. Not a good signal in the hover.
U-MIX	Ditto
V-MIX *Ditto*	
AMOD-L	Actuator Model (safety)
AMOD-R	ditto
AMOD-TR	ditto
AERR-TR	Actuator Error (safety)
AERR-COL	ditto
AMOD-COL	ditto
AERR-L	ditto
AERR-R ACT-ERR	ditto

Table C.2: FTD Variables - Continued

UDOT	Rate of a velocity - X direction
VDOT	Rate of a velocity - Y direction
WDOT	Rate of a velocity - Z direction
HT-THETA	N/A
PSI1	N/A
LH-ACT	Aircraft Actuator position - Left Hand
RH-ACT	Aircraft Actuator position - Right Hand
TR-ACT	Aircraft Actuator position - Tail Rotor
COL-ACT	Aircraft Actuator position - Collective
H-MIX	mixed baro hight
DELA	Latitude stick input DELE Long stick input
DELR	Pedal stick input
DELP	Collective stick input
PSI	Euler angle
TP-THETA	N/A
TP-PHI	N/A
TP-PSI	Not defined
RADALT	Raw Adalt signal
AY	Not defined
P-DYN	Dynamic Pressure
P-STAT	Not defined
AX	Acceleration
AX2	Not defined
AZ2	Not defined

Table C.3: FTD Variables - Continued

ALPHA	Angle of Attack
BETA	Angle of sideslip
T-TOT	Total Temperature
HTPSI	Head Tracker - Psi
HTTHETA	Head Tracker - Theta
HTPHI	Head Tracker - Phi
HRA	Radalt Hight
UD-MIX	Doppler velocity mixed with acceleration to a high frequency component
	This is the signal that is used in the hover as opposed to U-MIX - (u component)
VD-MIX	Same Doppler velocity - (v component)
WD-MIX	Same Doppler velocity - (w component)
HDOT-MIX	Rate of change of hight, based on doppler and radalt
HI-MIX	Not defined
AX	Acceleration fore/aft
AZ	Acceleration normal
HT-THETA	N/A
HT-PSIW	N/A
PSIW	Heading info
T-PSI	N/A
T-PHI	N/A
T-THETA	N/A
PHI-T	Not defined
THETA-T	Not defined
PSI-T	Not defined
ROLL-LIM	N/A
MODP	On-board aircraft model Roll Rate
MODQ	On-board aircraft model Pitch Rate
MODTHETA	On-board aircraft model angle -theta
MODPHI	On-board aircraft model angle -roll
MODR	On-board aircraft model angle -rate yaw

Appendix D

A Result on the Rank of Solutions to a Kind of Algebraic Riccati Equation

Please note that the notation used in this appendix has no bearing on those used in the rest of the thesis.

Consider the following algebraic Riccati equation (ARE) of Y:

$$AY + YA^T - YC^TCY = 0.$$

Such AREs are met in the S-over-KS (mixed sensitivity) H_∞ design.

Without loss of generality, we may assume that the matrix A has been transformed, by appropriate orthonormal transformations, into its Schur form as

$$A = \begin{bmatrix} A_{us} & A_{12} \\ 0 & A_s \end{bmatrix}$$

where $A_{us} : n_1 \times n_1$, is the anti-stable part and $A_s : n_2 \times n_2$, is the stable part. The stabilizing solution Y of the above ARE is such that $Y \geq 0$ and $A - YC^TC$ is stable. We may have the following result.

Lemma D.1 *The rank of the stabilizing solution Y equals to n_1, the number of unstable* *eigenvalues of* A. *And,* Y *is of the form*

$$Y = \begin{bmatrix} Y_{11} & 0 \\ 0 & 0 \end{bmatrix}$$

where $Y_{11} : n_1 \times n_1$ *and* $Y_{11} > 0$.

Proof: The matrix C can be partitioned accordingly as $C = \begin{bmatrix} C_1 & C_2 \end{bmatrix}$, and similarly for Y,

$$Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix}$$

The Riccati equation is thus

$$0 = \begin{bmatrix} A_{us} & A_{12} \\ 0 & A_s \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} + \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} A_{us}^T & 0 \\ A_{12}^T & A_s^T \end{bmatrix} - \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} C_1^T \\ C_2^T \end{bmatrix} \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix}$$

Consider the (2-2)block of the above equation:

$$0 = A_s Y_{22} + Y_{22} A_s^T - \begin{bmatrix} Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} C_1^T \\ C_2^T \end{bmatrix} \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} Y_{12} \\ Y_{22} \end{bmatrix}$$

It can be re-written as

$$A_s(-Y_{22}) + (-Y_{22})A_s^T = - \begin{bmatrix} Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} C_1^T \\ C_2^T \end{bmatrix} \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} Y_{12} \\ Y_{22} \end{bmatrix}$$

From the property of a Lyapunov equation and from the assumption that A_s is stable, we have $-Y_{22} \geq 0$. However, on the other hand, $Y_{22} \geq 0$ from the requirement that $Y \geq 0$. Hence, we have

$$Y_{22} = 0.$$

Consequently,

$$\begin{bmatrix} Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} C_1^T \\ C_2^T \end{bmatrix} = 0$$

And, the following matrix must be stable,

$$\begin{aligned}
&\begin{bmatrix} A_{us} & A_{12} \\ 0 & A_s \end{bmatrix} - \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} C_1^T \\ C_2^T \end{bmatrix} \begin{bmatrix} C_1 & C_2 \end{bmatrix} \\
&= \begin{bmatrix} A_{us} & A_{12} \\ 0 & A_s \end{bmatrix} - \begin{bmatrix} Y_{11}C_1^T + Y_{12}C_2^T \\ 0 \end{bmatrix} \begin{bmatrix} C_1 & C_2 \end{bmatrix} \\
&= \begin{bmatrix} A_{us} - (Y_{11}C_1^T + Y_{12}C_2^T)C_1 & A_{12} - (Y_{11}C_1^T + Y_{12}C_2^T)C_2 \\ 0 & A_s \end{bmatrix}.
\end{aligned}$$

Now, consider the (2-1) block of the equation which is

$$\begin{aligned}
&0 = A_s Y_{12}^T + (Y_{12}^T A_{us}^T + Y_{22} A_{12}^T) - \begin{bmatrix} Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} C_1^T \\ C_2^T \end{bmatrix} \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} Y_{11} \\ Y_{12}^T \end{bmatrix} \\
&= A_s Y_{12}^T + Y_{12}^T A_{us}^T - Y_{12}^T C_1^T (C_1 Y_{11} + C_2 Y_{12}^T) \\
&= A_s Y_{12}^T + Y_{12}^T (A_{us} - (Y_{11}C_1^T + Y_{12}C_2^T)C_1)^T
\end{aligned}$$

The matrix coefficient $(A_{us} - (Y_{11}C_1^T + Y_{12}C_2^T)C_1)^T$ is stable from the above deduction. Because A_s is stable as well, this Sylvester equation of Y_{12} has a unique solution. Since zero is a solution, $Y_{12} = 0$. Thus

$$Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} = \begin{bmatrix} Y_{11} & 0 \\ 0 & 0 \end{bmatrix}$$

where Y_{11} is the stabilizing solution of

$$A_{us} Y_{11} + Y_{11} A_{us}^T - Y_{11} C_1^T C_1 Y_{11} = 0.$$

Obviously, if A is stable, the stabilizing solution is $Y = 0$, and when A is anti-stable, we can have a stabilizing solution $Y > 0$.

Bibliography

[1] M. F. Al-Malki and Da-Wei Gu. Design and Analysis of Helicopter Fault Tolerant Flight Control System -The Lateral Case. *IFAC Symposium on Aerospace Applications, Saint-Petrsburg, Russia*, 2004.

[2] M. F. Al-Malki and Da-Wei Gu. Sensors Fault Detection, Isolation with Application to High Performance Helicopter. *International Conference on Simulation and Modelling, Marbella, Spain*, 3-5 September 2003.

[3] M. F. Al-Malki and Da-Wei Gu. An ANN-Based Sensors' Fault Detection, Isolation, and Accommodation for Bell-205 Helicopter. Technical Report -Leicester University, UK, August 2001.

[4] Stewart W. Baillie Arthur W. Gubbels, J.Murray Morgan. Modifications to the NRC Bell-205 Airborne Simulator Safety System in Response to a Recent Incident.

[5] N. Bailey. *The Helicopter Piolt Manual Volume 1 and 2*. Airlife Publishing Ltd, 1992.

[6] D. Bates. *Lecture Notes on Robust Optimal Multivariable H_∞ Control*. 2004.

[7] D. Bates and I. Postelthwaite. *A Course in Robust Multivariable Control of Aerospace System*. Delft University Press, 2002.

[8] J. H. Blackelock. *Automatic Control of Aircraft and Missiles*. 1965.

[9] M. Bodson. Multivariable Adaptive Algorithms for Reconfigurable Flight Control. *IEEE TRX on Control System Technology*, Vol 5(No. 2):217–229, March 1997.

[10] M. Brown and C. Harris. *NeuroFuzzy and Adaptive Modelling and Control.* Prentice-Hall, 1994.

[11] J. J. Burken. Reconigurable Flight Control Designs with Application to the X-33 Vehicle. *AIAA Proccedings on GNC*, pages 951–965, 1999.

[12] A. J. Calise and M. Sharma. Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft. *AIAA Guidance, Navigation, and Control Conference*, pages 88–97, 1998.

[13] J. Chen and R. Patton. *Robust Model-based Fault Diagnosis for Dynamic System.* Kluwer Academic Piblishers, 1999.

[14] DERA. *Configuration of the DERA HEILISIM Model to Represent the Flight Dynamics of the NRC Bell-205 Helicopter (U).* DERA, Farnborough - Hampshire GU14 0LX UK, 1998.

[15] D. Saad (editor). *On-line Learning.* Cambridge Press.

[16] A. Calise et al. Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft. *AIAA GNC*, 1998.

[17] A. Calise et al. Fault Tolerant Flight Control via Adaptive Neural Network Augmentation. *AIAA-98*, pages 1722–1728, 1998.

[18] A. Calise et al. Nonlinear Adaptive Flight Control using Neural Networks. *AIAA Guidance, Navigation, and Control Conference, New Orleans, LA*, August 1997.

[19] A. Smerlas et. al. Design and Fligt Testing of an h_∞ Controller for the NRC Bell 205 Experimental Fly By Wire Helicopter. *AIAA GNC*, pages 1023–1033, 1998.

[20] B. Kim et al. Nonlinear Flight Control Using Neural Netwroks. *Journal of Guidance, Control, and Dynamics*, 20(1), 1997.

[21] B Stevens et. al. *Aircraft Control and Simulation.* John Wiely, 1992.

[22] D. Del Gobbo et. al. Re-engineering Fault Tolerance Requirements: A Case Study in Specifying Fault Tolerant Flight Control Systems. *IEEE Fifth International Symposium on Requirements Engineering*, pages 236–244, 2001.

[23] D. Walker et. al. Robust Control of the Longitudinal and Lateral Dynamics of the BELL-205 Helicopter. *International Journal of Control*, 63:(4,):2742–2746, 1999.

[24] E. Moya et. al. Neural PCA Based Fault Diagnosis. *Proceedings of the European Conference*, pages 809–813, 9 2001.

[25] I. Postelthwaite et. al. H_∞ Control of the NRC Bell 205 Fly By Wire Helicopter. *Journal of The American Helicopter Society*, pages 276–284, 1999.

[26] J. Boskovic et al. Multiple-Model Adaptive Flight Control Scheme for Accommodation of Actuator Failures. *Journal of Guidance, Control and Dynamics*, 25(4): 712–724, August 2002.

[27] J. Brinker et al. Nonlinear Simulation of Tailless Advanced Fighter Aircraft Reconfigurable Flight Control Law. *AIAA Guidance, Navigation and Control conference*, .

[28] J. Brinker et al. Reconfigurable Flight Control for a Tailless Advanced Fighter Aircraft. *AIAA Proceedings*, 1998.

[29] L. Ljung et al. *Systems Identification: Theory for user*. Springer-Verlag, 1999.

[30] M. Hagan et. al. Neural Networks for Control. *American Control Conference*, 1999.

[31] M. R. Napolitano et al. Kalman Filters and Neural-Network Schemes for Sensor Validation in Flight Control Systems. *Journal of Guidance, Control and Dynamics*, Vol 14 No. 1, .

[32] M. R. Napolitano et al. New Technique for Aircraft Flight Control Reconfiguration. *IEEE TRX on Control Systems Technology*, 6(5):596–611, 1998.

[33] P. Petkov et. al. *Computational Methods for Linear Control System*. Prentice-Hall, 1991.

[34] P. W. Lancaster et al. Linear Algebra, Control, and Stability. *Lectures Notes - Department of Mathematics and Sstatistics*, 1999.

[35] R. Patton et. al. *Issues of Fault Diagnosis System.* Springer-Verlag, 2000.

[36] R. Rysdyk et al. Fault Tolerant Flight Control via Adaptive Neural Network Augmentation. *AIAA Guidance, Navigation, and Control 1998*, 1998.

[37] R. Schultz et al. On-Line Least Square Training for the Underdetermined Case. .

[38] Robert K. Heffly et. al. *A Compilation and Analysis of Helicopter Handling Qualities Data. Volume one: Data Compilation.* NASA Scientific and Technical Information Branch.

[39] S. Bittanti et. al. *The Riccati Equation.* Springer-Verlag, 1991.

[40] S. Lawerence et. al. What size neural network gives optimal generalization convergence properties of backpropagation. Technical Report UMIACS-TR-96-22 and CS-TR-3617, Institute of Advanced Computer Studies - University of Maryland, USA, June 1996.

[41] S. M. Bradtke et al. Multivariable Adaptive Algorithms for Reconfigurable Flight Control. *Machine Learning*, 1996.

[42] S. Teffner-Cluasen et. al. The Application of μ Analysis and Synthesis to the Control of an ASTVOL Aircraft. *In Proceeding of the EUROACO Workshop on Recent Results in Robust Adaptive Control, Florance-Italy*, pages 304–322, Sept 1995.

[43] T Troudet et al. Neurocontrol Design and Analysis for a Multivariable Aircraft Control Problem. *Journal of Guidance, Navigation and Control*, 18(4), July-August 1993.

[44] T. W. Long et al. Sensor Fusion and Failure Detection using Virtual Sensors. *Proceding of American Control conference*, 1999.

[45] V. Ionescu et al. General Matrix Pencil Techniques for the Solution of Alegebraic Riccati Equations - A Uunified Approach. *IEEE Transaction on Automatic Control*, 42(8), 1997.

[46] Y. Li et al. Neuro-controller Design for Nonlinear Fighter Aircraft Manuever Using Fully Tuned RBF Networks. *Automatica*, 37:1293–1301, 2001.

[47] Y. Li et al. Robust H_∞ Controller Design for Aircraft Auto-Landing. *IEEE Transaction on Aerospace Electronic Systems*, 40(1), 2004.

[48] Y. Ochi et al. Application of Restrucurable Flight Control System to Large Transport Aircraft. *Journal Of Guidance, Control and Dynamics*, Vol 18 No. 2.

[49] Bruce A. Francis. *A Course in H_∞ Control Theory*. Springer-Verlag, 1986.

[50] M. Green and D. Limebeer. *Linear Robust Ccontrol*. Prentics-Hall, 1995.

[51] Da-Wei Gu and P. Petkov. *Robust Control Systems Design*. In-print, 2004.

[52] Jian Liang Wang Guang-Hong Yang and Yeng Chai Soh. Reliable H_∞ Controller Design for Linear Systems with Sensors Failures. volume 37th of *Proceeding on the 37th IEEE Conference on Decison Control*, pages 2822–2827, Florida, USA, December 1998. IEEE.

[53] Jian Liang Wang Guang-Hong Yang and Yeng Chai Soh. Reliable LQG Control with Sensor Failures. volume 38th of *Proceeding on the 38th IEEE Conference on Decison Control*, pages 3564–3568, Arizona, USA, December 1999. IEEE.

[54] M. Gupta et. al. (editor). *Neuro-Control Systems*. IEEE Press, 1994.

[55] J. Han H. Zhao and X. Zheng. Weighting Function Selection In H_∞ Mixed Sensitivity Design. *Proceedings of 1^{st} FPNI-PhD Symposium - Hamburg*, pages 181–188, 2000.

[56] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, 1994.

[57] S. Haykin. *Neural Networks: A comprehensive Foundation*. IEEE Press, 1994.

[58] M. Ching I. Postelthwaite and Da-Wei Gu. Weighting Function Selection In H_∞ Design. *IFAC 11th Conference*, 1990.

[59] S. Ooung I. Postelthwaite and D.-W Gu. H∞ Control System Design: a Critical Assessment Based on Iindustrial Applications. VIII(6):329–333.

[60] al. J. Leitner et. Analysis of Adaptive Neural Networks for Helicopter Control. *AIAA Journal of Guidance, Navigation, and Control*, Vol 20 No 5, Sept-Oct 1997.

[61] W. D. Johnson. *Helicopter Theory*. Dover, 1980.

[62] A. Laub. A Schur Method for Solving Riccati Equations. *IEEE Transaction on Automatic Control*, 24(6), 1979.

[63] Mathworks. *Matlab Neural Networks Toolbox Reference Manual*. Mathworks, 1992-2001.

[64] D. McLean. *Automatic flight Control*. Prentice-Hall, 1990.

[65] Ghassan Murad, Ian Postelthwaite, and Da-Wei Gu. A Robust Design Approach to Integrated Controls and Diagnostics. *1996 IFAC World Congress*, June 30-July 5 1996.

[66] O. Neils. *Nonlinear System Identification*. Prentice-Hall, 2001.

[67] A. Oliva. Flight Control Reconfiguration Following Sensor Failure. *AIAA GNC Conference*, pages 1–8, 2001.

[68] M. Osterom and R. Babuska. Aircraft Sensor Management and Flight Control Law Reconfiguration - Fuzzy Logic Approach. *AIAA Guidance, Navigation, and Control Conference and Exhibtion*, pages 1–11, 2001.

[69] A. Packard and J. Doyle. The Complex Structured Singular Value. *Automatica*, 29 (1), 1993.

[70] G. D. Padfield. *Helicopter Flight Dynamics:The Theory and Application of Flying Qulaities and Simulation Modeling*. AIAA Press, 1996.

[71] R. Patton. Where Are We in Fault-Tolerant Control. *Safeprocess 2000*, 2000.

[72] F. Poon and Da-Wei Gu. Robust State Estimation and Fault Detection. Technical Report, 1999.

[73] R. W. Prouty. *Helicopter Performance, Stability and Control.* Krieger Publishing Co, 1986.

[74] Jure V. Medanic Robert J. Veillette and William R. Perkins. Design of Reliable Control Systems. *IEEE TRX on Automatic Control*, 37(3):290–304, 1992.

[75] Sigurd Skogestad and Ian Postelthwaite. *Multivariable Feedback Control: Analysis and Design.* Wiely, fifth edition, 2000.

[76] A. Smerlas. *Robust Multivariable Control of Helicopters: From Mathematical Models to Flight Tests.* PhD thesis, Leicester,UK., March 1999.

[77] M. C. Turner. *Robust control of Systems Subject to Input Nonlinearities with Application to High Performance Helicopter.* PhD thesis, Leicester University, UK. June 2000.

[78] Mattheew C Turner. Private Communication. *Leicester University, Engineering Department - Control Systems Lab*, 2004.

[79] Robert J. Veillette. Reliable Linear-Quadratic State-Feedback Control. *Automatica*, 31(1):137–143, 1995.

[80] A. J. Wagtendonk. *Principles of Helicopter Flight.* Aviation Supplies Academics, Inc, 1995.

[81] D. Walker. On the Structure of a Two-Degree-of-Freedom H_∞ Loop Shaping Controller. *International Journal of Control*, 63:(6,).

[82] J. Watton. Private Communication. *Cardiff University, UK*, 2004.

[83] Kemin Zhou with John Doyle. *Essentials of Robust Control.* Prentice-Hall, 1998.

[84] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice-Hall, Upper Saddle River, New Jersey 07458, 1995.

[85] K. Zhou and Zhang Ren. A New Controller Architecture for High Performance, Robust and Fault-Tolerant Control. *IEEE Transaction on Automatic Control*, 46 (10):1613–1618, 2001.