Cranfield UNIVERSITY

Anna Lubomirova Blumel

Robust Fuzzy Autopilot Design Using Multi-objective Optimisation for a Highly Non-linear Missile

Department of Aerospace, Power & Sensors

PhD Thesis

Royal Military College of Science
Shrivenham

Cranfield UNIVERSITY

Department of Aerospace, Power & Sensors

PhD Thesis

Academic Year 2000

Anna Lubomirova Blumel

Robust Fuzzy Autopilot Design Using Multi-objective Optimisation for a Highly Non-linear Missile

Supervisor
Professor Brian A. White

March 2001

A.L.Blumel@rmcs.cranfield.ac.uk

Royal Military College of Science
Shrivenham

Abstract

The thesis is focussed on designing a robust nonlinear autopilot design for a highly nonlinear missile system in the presence of parametric uncertainties. First, Feedback Linearization is applied to the nominal missile model which produces an equivalent linear system. Applying linear control techniques, an outer loop is designed to drive the controlled variables to reach the required demand, hence the missile can follow a desired trajectory. Unfortunately the control law produced by the feedback linearization is not robust in the presence of uncertainties and hence in a real flight scenario will not be valid, and will exhibit nonlinear behavior for small changes in system parameters. Fuzzy logic trajectory control is then used in the outer loop to improve the robustness of the feedback linearization technique. An evolutionary genetic algorithm is then used to optimise the fuzzy control parameters. Multiple solutions (alternative fuzzy controllers) are obtained by using a Pareto based approach with non-dominated sorting. This has been combined with the reference point approach to incorporate preference information into the genetic algorithm to direct the search towards feasible areas which satisfy specified ranges on each objective. The design meets objectives defined on the closed loop performance: steady state error, rise time settling time and maximum percentage overshoot. From the multiple solutions the designer can choose the one which satisfies specified requirements. Fuzzy scheduled controllers are also used to control side-slip velocity for a large range of multiple demands. The design has been exercised for multi-model airframe dynamics at vertex points defined by 16 variables.

Contents

1 Introduction **1**
- 1.1 Problem definition . . . 1
- 1.2 Literature review on existing control techniques . . . 3
  - 1.2.1 Conventional methods . . . 3
  - 1.2.2 Artificial intelligence . . . 7
  - 1.2.3 Hybrid techniques . . . 13
  - 1.2.4 Summary . . . 15
- 1.3 Aims of the thesis and its structure . . . 15
- 1.4 Contributions . . . 18

2 Non-linear system. An aerospace application **19**
- 2.1 The Missile Motion Dynamics . . . 19
- 2.2 Horton Model Dynamics . . . 22
- 2.3 Aerodynamic coefficients for different flight conditions . . . 24
- 2.4 Polynomial form . . . 27
- 2.5 Open-Loop Stability Analysis . . . 32
- 2.6 Cross-coupling effect . . . 34
- 2.7 Nonlinearity . . . 36
- 2.8 Multi-modelling airframe dynamics . . . 37
  - 2.8.1 Parametric uncertainties . . . 37
  - 2.8.2 Sensitivity Analysis . . . 41
- 2.9 Cartesian to Polar coordinates . . . 43
  - 2.9.1 Missile model dynamics in Cartesian coordinates . . . 43
  - 2.9.2 Missile model dynamics in Polar coordinates . . . 43
- 2.10 Closed loop autopilot requirements . . . 44
- 2.11 Conclusions . . . 45

3 Feedback Linearization **46**
- 3.1 Introduction . . . 46
- 3.2 Feedback Linearization theory . . . 47
  - 3.2.1 Feedback Linearization process . . . 47
  - 3.2.2 Tracking Control . . . 48
  - 3.2.3 Normal forms . . . 50
  - 3.2.4 Examples of Input/Output Linearization . . . 52
  - 3.2.5 Summary . . . 54

3.3 Trajectory control design for SISO system . . . 56
3.3.1 Design 1: Tracking lateral acceleration via lateral velocity . . 56
3.3.2 Design 2: Tracking lateral acceleration via augmented acceleration . . . 62
3.4 Trajectory control design for MIMO system . . . 66
3.4.1 Design 1: Cartesian coordinates . . . 66
3.4.2 Design 2: Polar coordinates . . . 72
3.5 Conclusions . . . 77

4 Robust Fuzzy Autopilot Design 78
4.1 Hybrid Fuzzy Nonlinear Control . . . 78
4.1.1 Fuzzy Logic philosophy . . . 79
4.1.2 Fuzzy trajectory controller for the feedback linearized system . . . 84
4.2 Optimisation of the Fuzzy Logic Controller . . . 85
4.2.1 Evolutionary Algorithm . . . 85
4.2.2 GA tuning the FLC parameters . . . 87
4.2.3 Results for the scalar optimisation problem . . . 93
4.3 Fuzzy Gain Scheduling . . . 98
4.3.1 Polynomial fit of the multiple demands for FLCs scaling factors 98
4.3.2 Results for a large range of lateral acceleration demands . . . 103
4.4 Conclusions . . . 107

5 Multi-objective optimisation using GA 108
5.1 Multi-objective optimisation problem . . . 109
5.1.1 Ideal vector . . . 110
5.1.2 Pareto Optimum . . . 110
5.1.3 Pareto front . . . 111
5.2 Review of multi-objective GAs-based approaches . . . 111
5.3 GA strategy for finding non-dominated solutions . . . 115
5.4 Optimistic Reference point approach . . . 117
5.4.1 Closed loop performance criteria . . . 119
5.4.2 Function of losses - preference information . . . 120
5.4.3 Decision Making . . . 121
5.4.4 Results using the reference approach . . . 126
5.4.5 Results for Fuzzy logic scheduled controllers multiple demands case - 1g, 5g, 10g, 15g . . . 132
5.5 The objectives defined as fuzzy constraints-penalties . . . 134
5.5.1 Results of fuzzy multi-objective optimisation . . . 135
5.6 Concluding remarks . . . 135

6 Conclusions, Discussions and Future work 138
6.1 Future work . . . 144
6.2 Publications . . . 145

A Nomenclature **146**
A.1 **Abbreviations** 146
A.2 Variables 148

B Physical Parameters of Horton Missile Model **153**

C Non-linear functions of the state space model **154**

D Lie Algebra **156**
D.1 Lie Derivative and Lie Bracket 156
D.2 Feedback linearization of MIMO systems 157
D.3 Complex Nonlinear Derivative 158

E Fuzzy logic glossary **159**

F Software implementation **161**

List of Figures

1.1 Guidance scenario 1
1.2 Nonlinear manoeuvre 2
1.3 Thesis structure 17

2.1 Definition of incidence angles 20
2.2 Airframe axis and nomenclature 22
2.3 Fin Coupling Moment coefficients 26
2.4 Set of four Centre of pressure coefficients X_{cp} 29
2.5 Set of four Normal Force coefficients C_{z_w} 30
2.6 Roll damping coefficient C_{m_q} 32
2.7 Pole and Zero Spectrum for constant $Mach = 3$ and varying incidence 33
2.8 Pole and Zero Spectrum for constant incidence = 0.1° and varying Mach number 33
2.9 Side-slip velocities and rates responses for a demand in fin angle ξ . . 34
2.10 Side-slip velocities and rates responses for a demand in fin angle ζ . . 35
2.11 1g, 5g, 20g demand 36
2.12 Aerodynamic coefficients ranges 40
2.13 Vertex points models 42
2.14 Transformation from Cartesian to Polar Coordinates 44

3.1 Tracking Control Diagram 48
3.2 Trajectory control design for **Design 1** in SISO 59
3.3 Error dynamics 60
3.4 Results for **Design 1** in SISO for $a_d = 10$ m/sec^2 and $a_d = 100$ m/sec^2 61
3.5 Trajectory control design for **Design 2** in SISO 64
3.6 Results for **Design 2** in SISO for $a_d = 10\ m/sec^2$ and $a_d = 50\ m/sec^2$ 65
3.7 Results for **Design 1** in MIMO for $a_d = 10\ m/sec^2$ and $a_d = 50\ m/sec^2$ 71
3.8 Results for **Design 2** in MIMO for $a_d = 14\ m/sec^2$ and $p_d = 0\ rad/sec$ 76

4.1 Membership functions defined for the error variable 80
4.2 Block-schematic representation of a fuzzy logic controller 81
4.3 Fuzzy-Feedback Linearized Autopilot Design 84
4.4 FLC chromosome structure with real-binary coding 90
4.5 FLC chromosome structure with real-integer coding 91
4.6 GA Multi-objective optimisation 92
4.7 Surface of two input, one output fuzzy controller 94

4.8 Results for 10 m/sec^2 lateral acceleration demand 94
4.9 Fuzzy control surface found by GA 95
4.10 Fixed gain closed loop performance 96
4.11 The autopilot test on the vertex points models 97
4.12 Multiple demands 98
4.13 Polynomial fit curves for multiple demands v_d 101
4.14 Polynomial fit curves for multiple demands a_d 102
4.15 The best tuned FLC for 15g and 5g simultaneously 104
4.16 Responses for different demands 105
4.17 Fixed gain trajectory controller for multiple demands 15g, 10g, 5g, 1g 106
4.18 Fuzzy Trajectory Controller for multiple demands 15g, 10g, 5g, 1g . . 106

5.1 Genetic algorithm structure 117
5.2 Closed loop criteria 119
5.3 Trade-off surfaces 123
5.4 Trade-off between each objective at generation 50 124
5.5 Trade-off between each objectives at generation 151 124
5.6 Trade-off between each objectives at last generation 250 125
5.7 Genetic algorithm convergence 125
5.8 Three alternative Pareto solutions with velocity control 127
5.9 Results 129
5.10 A set of lateral acceleration responses 130
5.11 Three alternative Pareto solutions with acceleration control 131
5.12 Fixed gain trajectory controller 132
5.13 Alternative solutions for multiple demands 133
5.14 Two possible alternative solutions for 5g demand 136

D.1 MIMO control design 158

Chapter 1

Introduction

1.1 Problem definition

The development of new technology in aerospace application suggest some novel shapes of flying objects which are becoming faster and faster with flexible and unexpected changes of their motion. In a typical guidance scenario, as shown in fig. 1.1, the guidance system is required to detect and defeat such dangerous targets.

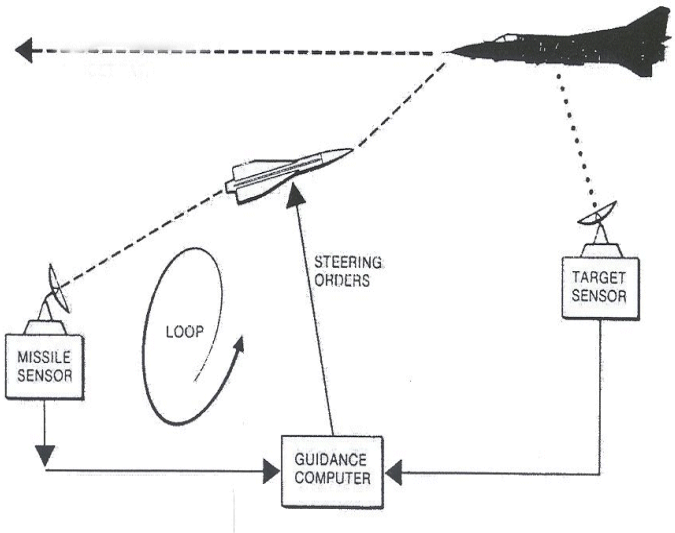


Figure 1.1: Guidance scenario

The guidance system produces commands in the form of lateral acceleration that the missile autopilot must follow accurately and fast. The performance of the guidance system relies on this performance to maintain its effectiveness. The autopilot must control the missile airframe response to this required speed and accuracy despite

large uncertainty and variability in the aerodynamic characteristics that are measured by wind tunnel tests on scaled models. As a consequence, their aerodynamic data are not accurate. There will be significant differences between the measured data and any airframe that the autopilot is required to control. Not only that, but each airframe will have its own uncertainty due to manufacturing tolerances. Hence there will be a large amount of uncertainty associated with any model of the airframe. There is also great variability in the performance of the airframe as speed and altitude vary. As the Mach number of the airframe can vary by a factor of 3 or 4 and the altitude can vary from sea level to 10Km (see fig. 1.2), a large change in dynamic pressure is present in the flight envelope. This has dramatic effects on the dynamics as the effectiveness of the wings and the control surfaces are determined by the dynamic pressure

$$\frac{1}{2}\rho V^2 \tag{1.1}$$

There are other factors that change the dynamics significantly, the greatest being the incidence at which the airframe is flying. Large changes in dynamic performance are evident as incidence changes, and the demand for large scale manoeuvres means that the missile can exhibit up to 30° of incidence for large acceleration demands. The usual way to maintain speed and accuracy is to produce a scheduled controller with altitude and Mach number and try to limit the incidence changes. This produces a set of linear controllers designs using a set of linear models. The challenge for the designer is to produce a single controller for all parts of the flight envelope. This entails dealing with a nonlinear model of the airframe and in producing a nonlinear controller. As it must also take account of the uncertainty in the aerodynamic data, it must also be robust to these uncertainties in the nonlinear model.

Figure 1.2: Nonlinear manoeuvre

1.2 Literature review on existing control techniques

In the analysis of non-linear control systems there is no general method for designing non-linear controllers. Several existing conventional as well as intelligent control techniques can be potential candidates for solving the problem stated in this thesis. Few of them are listed below:

1.2.1 Conventional methods

Gain-scheduling

A technique for transforming original system models into equivalent models of a simpler form is the so called Jacobian linearization or linearization about an equilibrium point. In this case it can be said that the linearization may not be a good approximation to the system for arbitrary configurations. Since the system is linearized about a single point, trajectory tracking can only be guaranteed in a sufficiently small ball of states about that point. There are several methods for circumventing this problem; one of the most common is gain scheduling as analysed by Shamma and Athans [1]. It was originally developed for the trajectory control of an aircraft. The idea of gain scheduling is to select a number of operating points which cover the range of the system operation. Then at each of these points, the designer makes a linear time invariant approximation to the plant dynamics and designs a linear controller for each linearized plant. Between operating points, the parameters of the compensators are then interpolated, or scheduled, thus resulting in a global compensator. To use gain scheduling, tracking controllers are designed for many different equilibrium points and gains are chosen based on the equilibrium points to which the system is nearest. Gain-scheduling is simple, and, practically successful for a number of applications. The main problem is that it has only limited theoretical guarantees of stability in non-linear operation, but it uses some loose practical guidelines such as "the scheduling variables should change slowly" and "the scheduling variables should capture the plant's non-linearities". Another problem is the computational load in a gain-scheduling design, due to the necessity of computing many linear controllers.

Feedback Linearization

An alternative technique is Feedback Linearization, known as non-linear dynamic inversion. Feedback linearization (FL) deals with techniques for transforming original system models into equivalent models of a simpler form. FL can be used as a non-linear design methodology. The main idea is to algebraically transform a non-linear system into a linear form using state feedback like in Isidori et al [2], Hunt and Sue [3], and Su [4], and then to use the well known linear design techniques to complete the control design. The purpose of dynamic inversion is to develop a feedback control law that linearizes the plant response to commands, then a non-linear con-

trol law is designed which globally reduces the dynamics of the selected controlled variables to integrators. A closed loop system is then designed to make the controlled variables exhibit specified command response and robustness requirements to the overall system. The approach can be used for both stability and tracking control problems and has been applied to a number of practical non-linear control problems. These include the control of helicopter, high performance aircraft and industrial robots by Marino and Spong [5], Wang and Vidyasagar [6]. There are few examples in the literature of the practical application of feedback linearization, electro servo-hydraulic actuator by Hahn et al [7]. Applications to aerospace systems are rare in the literature Bezick et al [8], Tahk et al [9] and Wee [10]. In their research work, the side-slip angle and the angle of attack are taken as outputs to design the control law. Then the accelerations are controlled using linear relations between body rates and accelerations at steady state.

Feedback Linearization technique requires full state measurement and desired tracking performance is only valid for exact knowledge of model parameters, however can be useful as model-simplifying device for robust non-linear control such as sliding or fuzzy logic control which are capable to provide robustness of the closed loop system.

Variable structure control techniques

Variable structure control systems (VSCS) evolved from the work in Russia of Emel'yanov and Barbashin in the early 1960s. The ideas appeared outside Russia after the mid 1970s when a book by Itkis (1976) and a survey paper by Utkin [11] were published in English. Later on they were followed by many other researchers White and Silson [12], Zinober [13], Slotine and Li [14], Edwards and Spurgeon[15]. Concepts of VSCS have been utilised in the design of robust regulators, model-reference systems, adaptive schemes, tracking systems, state observers and fault detection schemes. The ideas have successfully been applied to problems such as automatic flight control, control of electric motors, helicopter stability augmentation systems, space systems and robots. The essential feature of a variable structure controller is that is uses non-linear feedback control with discontinuities on one or more manifolds (sliding hyper-planes) in the state space or error space. This method is attractive in the design of controls for non-linear uncertain dynamic systems with uncertainties and non-linearities of unknown structure as long as they are bounded and occurring within a subspace of the state space.

Sliding Mode Control

The aim of the Sliding Controller (SMC) is to design a non-linear feedback controller for a class of non-linear systems given the extent of parametric uncertainty, disturbances and the frequency range of unmodelled dynamics. The technique has

been applied to a variety of plants with highly non-linear dynamics similar to a missile system: aircraft systems by Singh [16], ships by McGookin et al [17], underwater vehicles by Trebi-Ollennu and White [18] and space systems by Singh and Iyer [19] and has proved the ability to achieve good tracking performance in the presence of an uncertain environment. The closed loop dynamic behaviour obtained from using a variable structure control law comprises two distinct types of motion. The initial phase, occurring whilst the states are being driven towards the surface (referred to as *reaching phase*), which is in general affected by any matched disturbances present. When the states reach the surface and the sliding motion (referred to as *sliding phase*) takes place, then the system becomes insensitive to all matched uncertainty as shown by Singh and Iyer [19]. The question of control for a class of nonlinear systems which can be decoupled by state-variable feedback has been considered by Singh [16, 19] for an aircraft and spacecraft system. The control law for asymptotically decoupled control of roll angle, angle of attack and side-slip in rapid, non-linear manoeuvres has been derived and large simultaneous lateral and longitudinal manoeuvres were performed in spite of uncertainty in the stability derivatives. The synthesis of longitudinal autopilots for missiles flying at high angle of attack regimes has been presented by Thukral and Innocenti [20]. The autopilot has been tested on a small section of the flight envelope (pitch channel) consisting of a fast 180° heading reversal in the vertical plane, which required robustness with respect to uncertainties in the systems dynamics induced by large variations in dynamic pressure and aerodynamic coefficients. Weil and Wise [21] have demonstrated the use of variable structured system control to design the longitudinal autopilot for a missile under combined aerodynamic surface (fin) and reaction jet control. High gain feedback using singular perturbation analysis is used to design the reaction jet switching surfaces and fin control law. Sliding control technique has been applied to design a pitch-axis control system for high performance aircraft by Hedrick and Gopalswamy in [22]. The control objectives were to track pilot g commands, while satisfying flying quality specifications. In the pitch axis problem, the dominant non-linearities are the aerodynamic coefficient variation with angle of attack and saturation of the actuator's position and rate response. In addition to that Fossen and Sagatun [23] have described the use of multi-variable sliding mode control in dynamic positioning of underwater vehicle (ROV). Trebi-Ollennu [24] has also shown that this method has great potential for controlling the ROV attitude and position with excellent robustness properties against parametric uncertainties and unmodelled dynamics.

Few advantages of this technique can be mentioned here: Only single design is required over the entire operating range of the vehicle so there is no need for a series of linearized controllers. Stability is maintained in Lyapunov sense. SMC has excellent robustness properties against parametric uncertainties when matching conditions are satisfied. In practice the switching, chattering control law should be replaced by a smooth approximation which can be very inconvenient. Another drawback can be pointed as the need of complete state information which may not always be available. SMC is a successful technique for controlling missiles, however

most researchers have only considered controlling angle of attack or angular velocities.

Back-stepping approach

Another technique defined as a different version of variable structure control is the back-stepping approach. This technique has been approached by Kanellakopoulos, Kristic and Kokotovic in [25] working at Berkeley California University USA, lately followed by other researchers like Fossen and Svein [26], Rios-Bolivar et al [27], Song and Kim [28]. The output tracking problem of a class of observable minimum-phase uncertain non-linear systems has been considered by Rios-Bolivar et al [29], and a solution based on a suitable combination of input-output linearization and the adaptive back-stepping control design procedure has been proposed. This approach can be applied to a large class of non-linear systems, including those that are not transformable into the parametric-pure and parametric-strict feedback forms, typically considered in the applications of the back-stepping procedure. The controlled smooth transition of the angular velocity of a non-linear DC-motor has been presented as an application example. A non-linear vectorial backstepping control law for commercial ships has been considered by Fossen and Svein [26]. Vectorial back-stepping is done in three steps corresponding to the state vectors of the ship dynamics, kinematics and actuator dynamics. Emphasis is placed on compensation of the actuator dynamics since the bandwidth of the propellers, thrusters and rudders is often close to the bandwidth of the ship dynamics. Global exponential tracking of the (x and y) positions and the yaw angle of a surface ship has been proven by applying Lyapunov stability analysis. Also a globally, uniformly asymptotically stable non-linear control law for dynamic positioning of ships has been derived by Aslaug and Fossen [30]. They have avoided linearization and gain-scheduling techniques. However a non-linear observer was used to produce noise-free estimates of velocity and position from noisy position measurements. Global uniformly asymptotic stability was proven by using the Lyapunov stability theory. Also an adaptive non-linear control design was applied by Song and Kim [28] to the pitch acceleration controller for a missile model. Missile motion is modelled to be non-linear with unknown parameters and uncertainties. Based on the model, an adaptive back-stepping method has been adopted which guaranteed uniform boundedness despite model uncertainties. This design has been exercised on a very simplified missile model.

Back-stepping approach is a very promising technique for an autopilot design of missiles which are highly non-linear in aerodynamics with unknown parameters. This approach is very robust to parametric uncertainties. By properly chosen Lyapunov function a global asymptotic stability can be proved. Conversely to Sliding Mode Control no chattering effect is involved. However, there is a need of an observer for the estimation procedure which is definitely not very appreciated by real engineers especially when a fast response is required from the missile autopilot design. Also this technique is an adaptive procedure and is a question of reliability to be implemented on a missile board.

1.2.2 Artificial intelligence

Since 1989 the Japanese have built the so called LIFE association for research and development of processing intellectual information. The president at that time, Katsushige Mita stated in few words the importance of the direction in such fields: "Operations of present computers depend on simple yes-no logic namely binary logic, which is different from the information processing inherent in human thinking. Therefore, evaluation based on common sense and flexible judgement is considered difficult to achieve by computers, hence intensive research is now aimed at the realization of artificial intelligence". Katsushige Mita (President of LIFE association). Ten years later the advanced technology in Japan has proved worthwhile.

An intelligent system should be able to cope with a variety of unexpected changes and environments which requires learning and adaptation ability. Such a system can be referred to as an intelligent control system where technology plays a major role in modern flight control design and implementation. One goal of the intelligent control approach is to make advanced control systems easier to design. Another goal is to make them less vulnerable to uncertainties in system parameters and to *unknown* environment. Two very popular approaches for performing non-linear control based on fuzzy logic and neural networks are reviewed in detail. In addition, the opportunities to combine the useful features of each and to improve their performance using evolutionary algorithms are also considered. Fundamental concepts of these three techniques have been found by Linkens and Nyongesa [31].

Fuzzy Logic

Control systems should have the capability to gain increasing knowledge of the system through operational experience, without the interference of human operators. The knowledge-based control techniques use reasoning mechanisms to determine the control action from the knowledge stored in the system and from the available measurements. These systems can improve the robustness of current control systems by incorporating knowledge that cannot be accommodated in analytic models upon which conventional control algorithms are based. A common type of knowledge-based control is the rule-based control, for which the control actions are described in terms of *if-then* rules. The principle of designing a fuzzy logic controller is to integrate an empirical knowledge and operator experience into the controllers by using fuzzy sets and fuzzy rules. The theory was developed by Zadeh [32] and then invented for control purpose by Lee [33]. Much of the expert's knowledge contains linguistic terms such as *small, negative, positive, etc.,* which can be represented by fuzzy sets. Using fuzzy sets and fuzzy operations it is possible to design a fuzzy reasoning system which can act as a controller. The control strategy is stored in the form of **if-then** rules in a rule base structure. The rules represent an approximate static mapping from inputs (e.g. errors) to outputs (control actions) and are determined by using expert knowledge of the process. The first industrial applica-

tion of fuzzy logic control was in a cement kiln control designed by Holmblad and Ostergaard [34]. The rules representing the controller actions were derived from the cement kiln operator's handbook. Since then, fuzzy logic control has been applied to various systems in the chemical process industry, consumer electronics, automatic train operation, and many other fields listed in Driankov et al [35]. For example the RCAM problem, as formulated in [36], investigated the use of knowledge-based control techniques for a realistic flight control problem. The hybrid controller structure was proposed by Schram [37] in which the inner loop consisted of classical attitude controllers and the outer loop was developed by using *pilot heuristics* of flying an aircraft. The fuzzy logic has provided a transparent interface between the low-level attitude control of aircraft and high-level reasoning of human pilots. A compromise was found in which performance and robustness properties were good with the penalty of excessive vertical and lateral acceleration. In addition Schram et al [38] introduced multiple fuzzy controllers in an adaptive control scheme to a failure tolerant control. Smooth transition between the control modes, of possibly different structure, has been automatically achieved in the case of a gradual degradation of control system components. This approach has been demonstrated on a non-linear, six degrees of freedom model of a transport aircraft under realistic assumptions about actuator dynamics and the results have shown that good performance has been achieved in case of severe actuator failures. An application of FLC to a supersonic missile has been investigated by Schroeder and Liu [39], but assuming the pitch plane autopilot is a linear-time invariant system. A fuzzy logic based MIMO roll rate controller has been designed by Chiu et al [40] for Rockwell International's advanced technology wing aircraft model. The FLC has produced commands to six surface deflections to control roll rate and four torsion moments. FLC has also been applied to angle, elevation and azimuth rates at Nasa Jonson Space Centre.

FLC has been useful when applied to control uncertain non-linear systems. Fuzzy reasoning builds the understanding of *imprecision into the process* which could be either *parametric uncertainty*, *unmodelled dynamics* or *imprecise measurement values*, hence can provide the ability to control a system in *uncertainty* or *unknown* environments which is one of the most important characteristics of an intelligent control system. Fuzzy logic control is a knowledge-based system that derives control actions based on input-output relationship, therefore, estimation of the system parameters is not required. FLC can model complex non-linear functions and derive smooth control action for uncertain system behaviour. However, if the initially chosen control parameters such as membership functions and rule base structure are not satisfactory in terms of closed loop performance, then it is necessary to use "trial and error" design philosophy, which may not always be convenient. It may be an expensive process computationally speaking. In such a case, an appropriate technique is required to optimise the fuzzy logic control parameters. Although fuzzy strategies suffer from some limitations, they can produce robust control design in the presence of parametric uncertainties and we suggest fuzzy logic based control as an appropriate technique to be used further in this study.

Neural Networks

Neural Networks (NNs) have shown great promise in solving non-linear control problems because of their universal approximation capability, as detailed by Hunt et al [41]. This powerful property has inspired the development of many neural-network based controllers without significant prior knowledge of the system dynamics. Artificial NNs are based on the attempt to mimic the brains operation in a particular way with a move away from hard, exact mathematical calculations towards generalising fuzzy computation, as given by Greenfield [42]. The brain's powerful thinking, remembering and problem solving capabilities have inspired many scientists to attempt computer modelling of its operation. There are several categories of neural controllers in the published literature such as: supervised control, neural adaptive control by Sanner and Slotine [43], back-propagation through time by Collins and Dror [44], adaptive critic architecture also known as learning control. An interesting approach is learning with critic algorithm given by Widrow et al [45]. The learning controller is described in terms of two-component combination. These components are the controller and the trainer. One perform tasks of a pattern recognition and control parameter selection, and the other to work as a teacher, which observes system performance and adjusts category boundaries in the controller. Neural networks have been used by McKelvey [46] to *model the unknown feedback control law* of an optimal flight control problem. The network uses "black box" structure and it is trained with the back-propagation learning method. In addition an adaptive critic based Neural network architecture has been applied to an autopilot by Balakrishnan and Biega [47]. Their approach has adapted two networks: a supervisor (critic) that assesses the outputs of the controller network and an action neural network controller for modelling the control law. Napolitano and Kincheloe [48] have proposed the implementation of on-line learning neural controllers in the autopilot control laws of a modern high-performance military aircraft. One advantage of their design is avoiding the precomputation, storing, and interpolation between thousands of feedback gains of a typical flight control system. Another advantage is the ability *to compensate for non-linearities and model uncertainties.* The traditional gain-scheduling-based-control laws for typical autopilot functions are replaced by on-line learning neural architectures, trained with the extended back-propagation algorithm. This algorithm has shown significant improvements over the conventional back-propagation method in learning, speed and accuracy. On-line local learning capabilities of the neural controllers have been demonstrated. Finally most relevant to our research is the work by McDowell et al [49] for hybrid neural-adaptive bank-to-turn lateral autopilot, described for a short-range command-to-line-of-sight (CLOS) surface-to-air missile. In order to achieve consistent tracking performance over the flight envelope, a multi-input/multi-output (MIMO) Gaussian radial basis function network has been employed. The hybrid neural autopilot was evaluated in three dimensional (six-degree of freedom) simulation studies against realistic pitch acceleration and roll rate profiles generated from a typical CLOS guidance scenario.

Few important advantages in using neural networks for controlling non-linear systems can be mentioned here: Firstly, the dynamics of the controlled system does not need to be completely known for the design of the controllers or for the modelling of the system. Secondly, the potential of on-line learning is a very powerful feature for controlling any process in real time. In addition NNs have the ability for adaptation and interpolation as well as the ability of parallel computation and an universal approximation capability, which altoghether make them an attractive and useful technique for solving a variety of non-linear control problems. Finally neural networks have very useful properties such as the associative storage and retrieval of knowledge. They can be trained to approximate any function sufficiently well. Conversely to such attractive characteristics, the applications of neural networks as elements of real-time control systems could be very limited for the following reasons: The closed loop system behaviour does not have formal mathematical characterisation; NNs have unstructured nature of black-box learning, hence cannot be certified. Also large numbers of iterations over the desired mapping are required before the network adequately reproduces the required responses. In conclusion from an academic point of view NNs are a very promising technique which can improve the performance and the robustness of the missile system. However from an engineering point of view this technique is not an appropriate control method to be implemented on a missile board as they cannot be certified.

Neuro-Fuzzy Control

Fuzzy logic controllers have several important benefits in that they do not require a complete analytical model of a dynamic system. They provide knowledge-based heuristic controllers for complex systems, and they can be analytically validated. However they are not well suited to learning. This means that fuzzy logic systems cannot meet the goals of adaptation to changes in system dynamics or to unmodelled dynamic characteristics, and they cannot gain increased performance through learning. On the other hand artificial neural networks have been successfully used to model and approximate various non-linear relationships and systems. Neural networks can be trained to learn the mapping between the input and the output domains based on observations without requiring knowledge of the structure of the underlying systems. They can exploit the inherent parallelism associated with fuzzy algorithms because of the lack of dependencies on control rules. Once the network is trained it can process the rules in parallel. They have shown to possess the ability to adapt to dynamic environmental changes through continuous training. The application of knowledge-based control techniques for flight control by Steinberg [50, 51] has indicated that techniques like neural networks and fuzzy systems can provide appropriate tools for non-linear identification by Linse and Stengel [52], control of aircraft by Napolitano and Kincheloe [48], helicopters by Sugeno et al [53] and spacecraft by Berenji et al [54], or flight control reconfiguration by Napolitano et al [55]. In these applications, neural networks generally serve as non-linear, sometimes adaptive, models while fuzzy systems are often used as supervisory, expert systems. Few

relevant research activities are enumerated as follows: An interesting combination of artificial neural networks and fuzzy logic controllers have been addressed by Hariri and Malik [56] to power system stabilizer, where the method retains all the advantages of adaptability, rapidity and robustness. By using neural network as a structure for the fuzzy logic controller, the design time of conventional FLC can be significantly reduced, membership functions and fuzzy rules of the controller can be generated automatically to meet the prespecified performance, i.e. tuning of the FLC control parameters has been solved. Compared to a conventional neural network, the training time was decreased, since *a priori* knowledge in the form of fuzzy *if-then* rules was employed. Shin and Vishnupad [57] have applied neuro-fuzzy techniques to a complex manufacturing processes. The underlying non-linear process has been modelled by NNs and the process control has been performed by FLC. The fuzzy rules have been automatically generated from the trained NN and fuzzy control has been performed by Mamdani implication. The simulation results have provided a robust and accurate way of controlling complex processes without knowledge about the model. Even when the process has changed dynamically, the NNs have learnt the functional relationships between input and output domains through continuous training and the fuzzy controller has derived the control actions. A different type of NNs have been used by Geng and McCullough in [58] called cerebellar model arithmetic computer NNs (CMAC) with a faster learning rate than conventional NNs and a limited amount of computation required at any point in the learning process. The researchers have used the strengths of CMAC and Fuzzy control schemes and applied for the use in the design of advanced missile control systems. The fuzzy CMAC has the capability of incorporating human knowledge into the system and processing information based on fuzzy inference rules. The flight control system has been evaluated using a series of non-linear simulations driven by the mathematical models of HAVE DASH II Bank to turn missile, to examine the stability, high angle of attack and flight path angle tracking.

In the conventional fuzzy design, the user must tune the membership functions of fuzzy sets defined in the input and output universe of discourse by trial and error. This drawback has been eliminated with neuro-fuzzy networks. Due to the supervised learning methods it is possible to optimise the antecedent and consequent parts of a linguistic rule based fuzzy system. The neuro-fuzzy systems are universal approximators of any non-linear functions, as proved by Buckley and Hayashi [59]. There is no need of trial and error procedure to tune the control parameters of the fuzzy logic controller as self learning inherently exist. These systems can be certified, can have high learning speed and be able to process the rules in parallel. By combining fuzzy logic and neural network the controller becomes more robust to imprecise information and external disturbances and an improvement of the performance can be guaranteed. However a major drawback is the design complexity. They may be very expensive and the question of being implemented on a missile board is still an open one for engineers.

Fuzzy-Genetic Algorithms

As pointed out earlier the membership functions of a fuzzy logic controller can be defined by trial and error or by an experts knowledge. The use of a neural network depends highly on the availability of sufficient data representing the input-output mapping, but in a situation where such data cannot be obtained an alternative approach would be necessary. One such approach is to test hypothetical trial solutions on the system and generate better solutions on the basis of the performances using evolutionary techniques. Genetic algorithms, which are modelled on natural evolutionary strategies, is methodology that has been introduced as a learning and optimisation technique under such conditions. They use operations found in natural genetics to guide them through the paths in the search space, can provide means to search poorly understood and irregular spaces and has been successfully applied to variety of function optimisations, self-adaptive and learning systems. By using GAs a randomised global search in a solution space is possible. In this space a population of candidate solutions, encoded as chromosomes is evaluated by a fitness function in terms of its performance. The best candidates 'evolve' and pass some of their characteristics to their 'offsprings'. A group of researchers, KrishnaKumar et al [60] have investigated a hybrid technique for synthesising fuzzy logic controllers as a stability augmentation system. This technique combines the control capabilities of fuzzy logic with the learning capabilities of genetic algorithms, to yield a fuzzy logic controller optimised to satisfy desired handling quality requirements. An optimal control model is used to provide the closed-loop handling quality metrics. Genetic algorithms are used to optimise the attributes of the fuzzy logic controller. These attributes include the control parameters such as membership functions and the rule base structure. The hybrid technique was implemented and tested off-line using a wide envelope FA/18 longitudinal model. The results proved the following: first, robustness of the hybrid technique in finding suitable FLC for different operating points with minimal user interaction; second, robustness of the optimised FLC to operate at different operating conditions with no gain scheduling; third, the ability of the GA in finding a suitable FLC with as few as 10 rules in the rule base. Another successful application of optimising control parameters but of a Sliding Mode controller has been investigated by McGookin et al [17]. It involves the performance of a control system for course changing manoeuvres of an oil tanker non-linear system. SMC theory has been used to define the structure of the controller where the GAs have been used to optimise key control parameters in order to obtain satisfactory performance. Trebi-Ollennu and White [18] have applied multi-objective fuzzy genetic algorithm optimisation approach to non-linear control system design. The technique has shown to provide an effective, efficient and intuitive framework for selecting parameters of a modern non-linear robust controller applied to remotely-operated underwater vehicles.

GAs have been recognised to be a powerful tool for learning in many control applications and especially with fuzzy logic where they have applied to the process of learning control rules, also selecting of rules and tuning of their membership

functions. An important notice to be made here is that a good solution depends on setting the objective function correctly. A major drawback of the technique is that GAs are computationally inefficient as many trials are necessary until finding the right solution. New high technology is able to produce still faster solutions. The implementation of these algorithms is made possible by the recent advances in technology along with the progress in parallel microprocessors equipment which can provide the availability of efficient and fast learning algorithms. As a conclusion we can highly recommend that this technique can guarantee reliability and can be useful for optimising missile trajectory control parameters.

1.2.3 Hybrid techniques

Neuro-Sliding Control

In a hybrid design both techniques will contribute in the following way: neural networks can model the complex dynamics of the non-linear function, while SMC can overcome some model residual terms and increase the robustness of the closed loop system. A neural network approach has been proposed by Cao et al [61] to determine the sliding mode equation and the control inputs. The approach involves the application of the single layer perceptron model and the Lyapunov stability theory. The advantage is that it can overcome the difficulty of determining the sliding mode equations. Another research group Qin et al [62] tackled the problem of robustness for a MIMO affine non-linear control system in which uncertainties are only bounded. A state feedback controller has been constructed where the non-linear closed loop system has been finitely attracted by a given neighbourhood of equilibrium state. The controller consists of two parts: the first one is a static nominal controller obtained by the variable structure control; the second one is a dynamic compensator obtained by the learning approach of an artificial neural network. The role of nominal controller is to make the non-linear nominal system arrive quickly in the neighbourhood of the sliding surface. The role of the dynamic compensator is to attenuate the influence of uncertainties on the system stability. Another robust controller design of non-linear dynamic systems has been proposed by Chiou et al [63] by combining SMC and Productive Networks. An attitude control problem of a spacecraft has been used to demonstrate the effectiveness of the proposed method. Essentially, the SMC utilizes a high-speed switching control action to drive the non-linear plant's state trajectories towards a specific hyper-plane in the state space. It will also maintain the state trajectories sliding on the specific hyper-plane for all time. Most relevant to our problem, Fu et al [64] have achieved an adaptive robust neural-network-based control approach for bank-to-turn missile autopilot design. The Lyapunov theory has been used to complete the closed loop stability proof. This scheme is a combination of neural networks and sliding mode control techniques. The former has modelled some unknown non-linear functions, whereas the latter has been used to overcome some modelling residual terms. To summarise, the technique does not require a priori training phase, the sliding parameters can be updated on-line gradually and continuously. Chattering and high gain can be

avoided, good accuracy and robustness can be achieved. There is a question-mark about Neural Networks being implemented on a board, but from an academic point a view the combination of these techniques is quite a powerful tool for designing a non-linear robust control for a missile system.

Fuzzy-Sliding Control

As for the sliding mode control, the bounds of uncertainties must be estimated in order to guarantee the stability of the closed-loop system and also its engineering application requires a chatter-free sliding mode control. Fuzzy control, as one of the most effective methods using expert knowledge, cannot be used for inference but also approximate any real continuous function over a compact set to arbitrary accuracy. There is similarity when comparing the SMC with boundary layer to an FLC whose rules have been derived from the phase plane as explained in Palm [65]. Since it is possible to define the dynamics of the error along a switching line by choosing the dynamical equation defining the sliding mode, it is straight forward to construct the control rules along the switching line and this can be done by simulating the error dynamics independent of the plant. Once the control rules are established along the switching line, the rules can be defined in the two semi-planes on either side of the switching one. The concept of fuzzy sliding mode controller was first suggested by Palm [65]. An adaptive fuzzy sliding mode control method has been applied to the control of the vertical motion of a mine hunting ROV by Trebi-Ollennu et al [66]. The effects of parameter variation of the ROV has been considered, and performance and robustness to uncertainty has been assessed. The effectiveness of the technique has been demonstrated by its ability to decouple pitch and heave of the ROV subjected to parameter variations. An adaptive fuzzy system has been used by Sun et al [67] as an adaptive approximator for the non-linear robot dynamics. They have proved that the fuzzy system is using the switching function and its derivative of the sliding mode as inputs, hence it can approximate the plant non-linear dynamics in the neighbourhood of the switching hyper-plane. Thus the fuzzy controller design has been simplified, and at the same time the fuzzy control rules have been obtained easily by the reaching condition due to the sliding mode control. The fuzzy adaptive control scheme based on sliding mode can maintain the invariant property of the sliding mode control and alleviate chattering without the sacrifice of robustness. The best features of self-organizing fuzzy control and sliding mode control have been combined by Lu and Chen [68] to achieve rapid and accurate tracking control. The chatter encountered by most sliding-mode control schemes was alleviated without sacrificing invariant properties. For verification of the scheme they have performed experiments on a magnetic leviation system where regulation and tracking was performed for validation. The fuzzy controller has been designed to learn and compensate for non-linearities and uncertainties, thus allowing a reduction of the sliding-mode controller switching gains. The final control system design is very robust to modelling imprecision and external disturbances. Due to the limitations of the techniques, the tuning of the fuzzy logic parameters is required

and there is a need of a suitable learning medium in order to increase the robustness and adaptability.

1.2.4 Summary

The hybrid techniques, based on conventional and artificial intelligent nature, are quite powerful and useful for solving non-linear control design problems. We propose that a combination of feedback linearization method and a fuzzy trajectory controller would be an interesting useful and new approach to solve the problem stated earlier in the thesis. The former would cancel the plant non-linearities and the latter would exercise the robustness of the closed loop system when a multiple model description of the airframe aerodynamics is used. An optimisation algorithm would then be required to determine the fuzzy control parameters. We suggest genetic algorithms based on evolutionary nature to be examined as they are useful when applied to multi-modal noisy search spaces. Finally in order to meet closed loop performance criteria such as: steady state error, overshoot, settling and rise time, the optimisation problem can be addressed from multi-objective point of view.

1.3 Aims of the thesis and its structure

This thesis has been driven by the following two aims:

1. To design an autopilot system for lateral acceleration and velocity control of a highly non-linear missile. The control system should be robust in the presence of parametric uncertainties and should be valid for a large range of multiple demands up to 15g pull of lateral acceleration.

2. To obtain multiple solutions - the alternative trajectory controllers which will allow the designer the freedom to choose the one which satisfy specified requirements. This would require the use of multi-objective optimisation to determine the trajectory control parameters.

Thesis structure

The structure of the thesis has been outlined in four stages as shown in figure (1.3):

Stage 1 is detailed in Chapter 2, which describes the complexity of the highly non-linear missile system. It is a real research model developed by Matra BA Co, which is described by look up tables that define the non-linear characteristics of the aerodynamics. It describes a full 5 degree of freedom model in parametric format with severe cross-coupling and non-linear behaviour. A polynomial model has been produced to match the parametric model as close as possible in a least squares sense. This polynomial model is in the form of polynomial relationships that are then used for control synthesis. Autopilot design requirements are specified. A set

of convex models is produced that map the vertex points in a high order parameter space (of the order of 16 variables). The multiple model description of the airframe aerodynamics is tested for sensitivity on the aerodynamic coefficients. In order to examine manoeuvrability the model is described in Cartesian and Polar coordinates.

Stage 2 is detailed in Chapter 3, which uses Feedback Linearization to transform the non-linear system dynamics into a linear form by using state feedback and a simple linear control technique can be used in the outer loop. An Approximate Feedback Linearization is used for lateral motion control. The main difference from other research work is that instead of using angles or body rates as outputs for the linearization process, lateral velocities and body accelerations are used. The design retains the order and the relative degree of the system in the linearization process, hence produces a linearized system with no internal or zero dynamics. Both SISO (the reduced 4^{th} order system, without interaction between lateral motion and roll) and MIMO (full 5^{th} order) systems are considered. Desired tracking performance is achieved assuming an exact knowledge of the nominal model parameters such as: aerodynamic coefficients and missile configuration parameters (i.e., reference area, Mach number, mass, moment of inertia).

Stage 3 is detailed in Chapter 4, which deals with a design of robust trajectory control in presence of parametric uncertainties. Unfortunately Feedback Linearization cannot guarantee desired performance in a real flight scenario when there are either parameter variations or external disturbances. Conversely fuzzy logic theory is useful when dealing with vague and imprecise information, hence it is used here to build a fuzzy logic trajectory controller to improve the robustness of the closed loop system. Then an evolutionary optimisation approach such as genetic algorithm is used to determine the membership function distribution and the rule base structure of the fuzzy logic controller. The robust design is tested on the multiple model description of the airframe aerodynamics with significant parametric uncertainties. Also fuzzy logic scheduled controllers for a missile autopilot design have been examined. The fuzzy logic input output scaling factors have been determined by using polynomial fit for a large range of multiple acceleration demands and a magnitude of 1g up to 15g has been examined.

Stage 4 is detailed in Chapter 5. A multi-objective evolutionary optimisation of the trajectory control parameters is used. The design meets objectives related to closed loop performance such as: steady state error, overshoot, settling and rise time. The last three objectives are also treated as fuzzy constraints (i.e. penalties), so the designer can analyse the behaviour of the optimiasation process depending on the way objectives have been handled. Multiple solutions are obtained simultaneously by using non-dominated sorting for forming the Pareto front, combined with a reference point approach to incorporate preference information into the genetic algorithm to direct the search towards feasible areas which satisfy specific values of the objectives. This allow the designer the freedom to choose solutions and investigate the properties of the system.

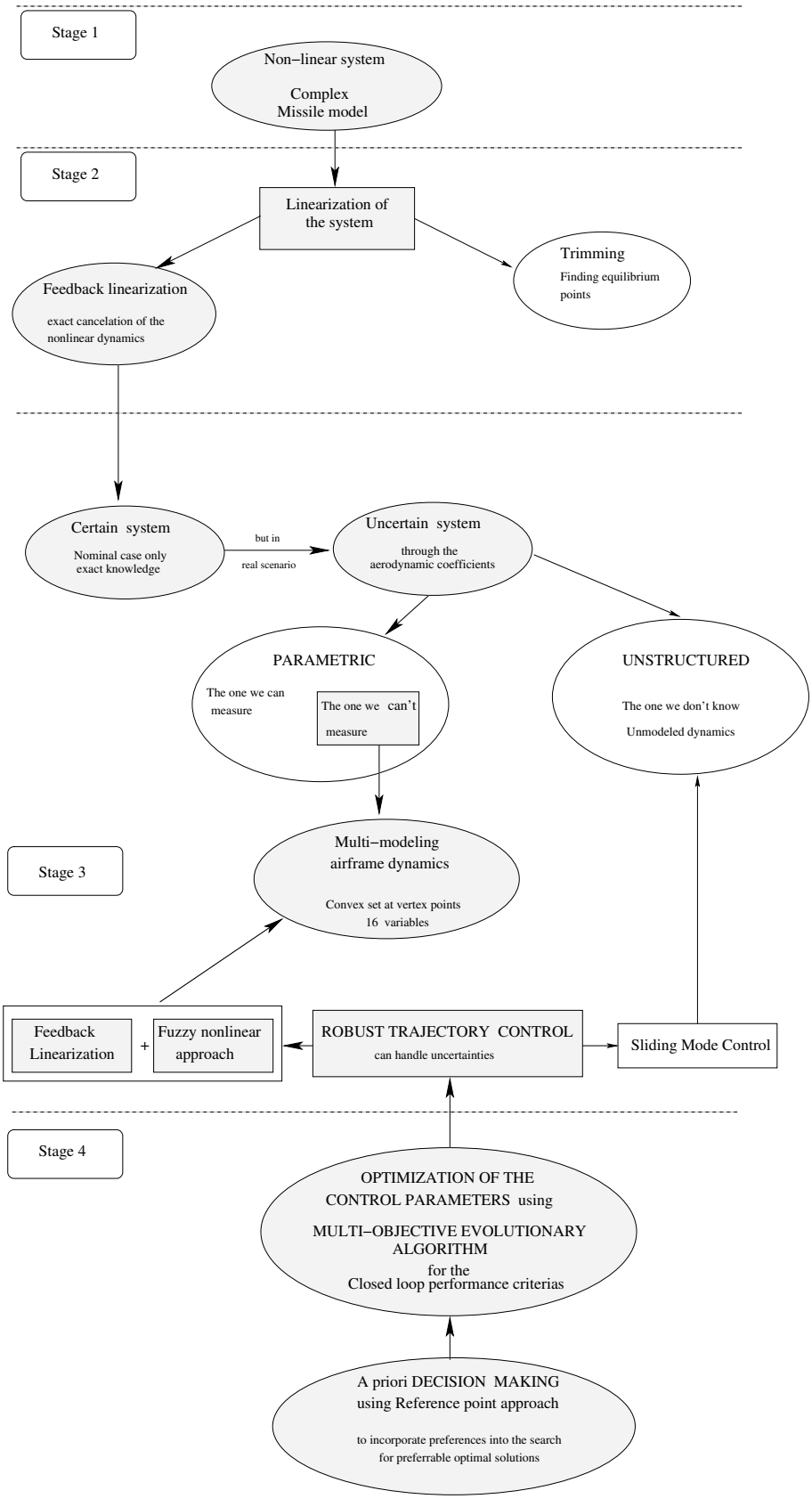


Figure 1.3: Thesis structure

1.4 Contributions

Main contributions

• A Fuzzy-Feedback Linearization non-linear autopilot is designed for highly non-linear manoeuvrable missile to cover a large range of parametric uncertainties of the multi-model description of the airframe dynamics, Chapter 4. A set of convex models is produced that map the vertex points in a high order parameter space (of the order of 16 variables). A detailed sensitivity analysis of the missile behaviour for each aerodynamic coefficient is examined, Chapter 2.

• Multi-criteria genetic algorithm optimisation is used to determine the membership functions and rule base structure of the fuzzy logic trajectory controller. This produces a design that meets objectives related to closed loop performance such as: steady state error, overshoot, rise and settling time. Both side-slip velocity and lateral acceleration control are considered. An unique way to incorporate preference information for each objective into the genetic algorithm is proposed to direct the search towards feasible area for finding solutions which satisfy specified requirements. An optimistic reference point approach is applied in a combination with a Pareto based non-dominating sorting technique, Chapter 5. The Pareto based non-dominating sorting algorithm is used from external source.

• Multi-objective optimisation of the fuzzy logic scheduled controllers is applied to the missile autopilot design. The fuzzy logic input output scaling factors are determined by using polynomial fit for a large range of multiple acceleration demands. A magnitude of 1g up to 15g is examined, Chapter 4.

• Lateral acceleration is controlled through side-slip velocity demand for the autopilot system considering the nominal model, Chapter 3.

Joint contributions

• Side-slip velocity autopilot design is achieved using Approximate Feedback Linearization for nominal model case. Both SISO and MIMO systems are examined. Lateral acceleration is controlled through side-slip velocity demand, Chapter 3.

• Applying Feedback Linearization to control directly lateral acceleration produces relative degree zero, which means all the states are unobservable and the system would be uncontrollable. Hence the augmented acceleration is defined as an output for the linearization process to produce relative degree equal with the order of the system to avoid internal dynamics. Lateral acceleration control is achieved through augmented acceleration using Approximate Feedback Linearization for the non-linear control design of the SISO system (i.e. yaw plane), Chapter 3.

• Both, roll and lateral acceleration are controlled by using Polar control for the MIMO system, Chapter 3.

Chapter 2

Non-linear system. An aerospace application

The research considered in the thesis is based on a fast, 1000 m/sec, highly non-linear manoeuvrable missile, developed by Matra BA Co. It is a real research model which is described by look up tables that define non-linear characteristics of the aerodynamics. It describes a full 5 degree of freedom model in parametric format with severe cross-coupling and non-linear behaviour. From the polynomials for 0° and 45° roll angle a linear interpolation has been done for the aerodynamic coefficients, hence rendered as a model in polynomial form.

2.1 The Missile Motion Dynamics

In this Section 2.1 the missile motion dynamics are described in general. The equations of motion, describing the angular and translational dynamics, are derived from Newton's Second Law of Motion expressed in the following form:

$$\begin{array}{ll} \sum Forces & = d\sum(TranslationalMomentum)/dt \\ \sum Moments & = d\sum(AngularMomentum)/dt \end{array} \tag{2.1}$$

where the translational and angular dynamics are described in details in the Horton report [69]. The aerodynamic forces and moments acting on the airframe are non-linear functions of longitudinal and lateral velocities, control surface deflection, body rates, etc, and they can be evaluated from empirical techniques, computational flow dynamics or wind tunnel test. In general, aerodynamic forces in body axes conform to the relationship (2.2), and similarly aerodynamic moments in body axes conform to (2.3).

$$Force = \frac{1}{2}\rho V_o^2 SC \tag{2.2}$$

$$Moment = \frac{1}{2}\rho V_o^2 SCd \tag{2.3}$$

where C is the aerodynamic force or moment coefficient, V_o - total velocity of the airframe, d- reference diameter, ρ the air density and S -reference area. A detailed

aerodynamic representation breaks the forces and moments down into a number of independent influences and defines derivatives, which have been adopted in the aerospace community and are used here. The suffix N_v indicates the influence giving rise to the derivative thus (2.4), C_{N_v} is a yaw moment derivative dependent on yaw velocity v.

$$C_{N_v} = \frac{\partial C_n}{\partial v} \tag{2.4}$$

As the missile manoeuvres it will generate lateral velocities v, w. The angles that these velocity vectors form are termed incidence angles, and these are illustrated in fig. 2.1.

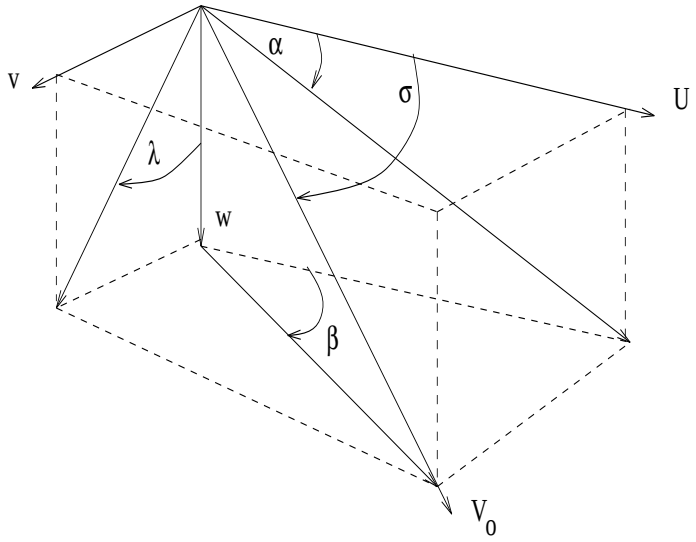


Figure 2.1: Definition of incidence angles

Where $V_o, \alpha, \beta, \sigma, \lambda$ are detailed in the following table:

Meaning	Formula
Total velocity	$V_o = \sqrt{U^2 + v^2 + w^2}$
Pitch incidence	$\alpha = tan^{-1}\frac{w}{U}$
Yaw incidence	$\beta = tan^{-1}\frac{v}{U}$
Total incidence	$\sigma = cos^{-1}\frac{U}{V_o}$
Aerodynamic roll	$\lambda = tan^{-1}\frac{v}{w}$
Lateral velocities	v, w

With no control surface deflection the effect of lateral velocity is to generate a lateral force which is distributed along the body/wing/tail assembly. However, this distributed force can be considered as a single force acting at a single resultant position which is termed the centre of pressure. The distance between the centre of pressure and the centre of gravity is termed the static margin. The lateral force,

acting through the static margin x_{sm} forms a lateral moment. Variations of x_{sm} are associated with the airframe stability. If the centre of pressure is behind the centre of gravity then the static margin is negative, the reverse arrangement produces a positive static margin. With a lateral force which increases it will be seen that the negative static margin produces a moment which tends to reduce the incidence and the airframe, is thus, statically stable. The positive static margin, however, produces a moment which tends to increase the incidence and the airframe is, thus, statically unstable. Since the centre of pressure varies with aerodynamic conditions and the centre of gravity varies with the fuel burnt then the airframe might be stable or unstable at different times in its flight. One role of the autopilot is to produce a stable, controllable missile in situations where the airframe is statically unstable.

The angular and translational dynamics of the model are cross-coupled and described by the full set of equations 6DOF:

$$\begin{aligned}
\dot{p} &= \frac{L}{I_x} + \frac{1}{I_x}(I_y - I_z)qr \\
\dot{q} &= \frac{M}{I_y} + \frac{1}{I_y}(I_z - I_x)pr \\
\dot{r} &= \frac{N}{I_z} + \frac{1}{I_z}(I_x - I_y)pq \\
\dot{u} &= \frac{X}{m} - wq + vr \\
\dot{v} &= \frac{Y}{m} - ur + wp \\
\dot{w} &= \frac{Z}{m} - pv + uq
\end{aligned} \tag{2.5}$$

where the forces (X,Y,Z) and the moments (L,M,N) are defined as:

$$\begin{aligned}
L &= l_p p + l_\zeta \zeta + l_\eta \eta + l_\xi \xi \\
M &= m_q q + m_w w + m_\eta \eta + m_\xi \xi \\
N &= n_r r + n_v v + n_\zeta \zeta + n_\xi \xi \\
X &= x_u u + x_p p + x_\xi \xi \\
Y &= y_v v + y_r r + y_\zeta \zeta \\
Z &= z_w w + z_q q + z_\eta \eta
\end{aligned} \tag{2.6}$$

where ζ, η, ξ are the inputs to the system. ζ is the rudder angle, η is the elevator angle and ξ is the aileron angle.

The definition of the axis systems (see fig. 2.2) in which the linear and angular motions are derived, is necessary, if the equations of motion and response characteristics of a homing missile are to be obtained.

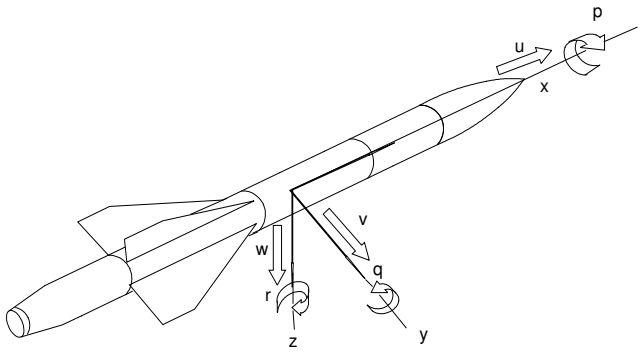


Figure 2.2: Airframe axis and nomenclature

The body axis set is, by definition, located at the centre of gravity of the missile and fixed to the body, so rotates and translates with it. The 'x' axis is taken forward, the 'y' axis out of the right hand wing and the 'z' axis downward forming a right hand set.

2.2 Horton Model Dynamics

The linearized airframe characteristics have been considered by Horton [69] and are used as a benchmark model for this study. The nonlinear model which is most likely to be the real scenario case has been developed and described in this section. The lateral motion is derived from the model defined in the report by Horton, while the roll model is derived from graphical relationships relating the moments generated by aileron, rudder and elevator action of the cruciform fin configuration. These relationships are used to generate a *parametric model* that is used for simulation and analysis. From this model a *polynomial model* is produced to match the *parametric model* as closely as possible in a least-squares sense. This *polynomial model* is in the form of polynomial relationships that are then used for control synthesis and which is also defined in this section.

Some assumptions for the Horton model have been made that lead to some simplifications such as:

• A rigid body of the missile is assumed for all flight conditions.

• A constant forward velocity $U = 1000\ m/sec$ with an approximate Mach $\simeq 3$ value is considered, so $\dot{u} = 0$. Only lateral motions are of interest hence a reduced

5 degree of freedom model is examined through this study.

• The skid-to-turn airframe of this missile has got symmetry about both y and z axes which leads to some simplification. Hence all products of inertia are zero and qr, pr, pq terms are discarded, also the moments of lateral inertias in the y and z axes are equal ($I_y = I_z$).

• The motion of the missile is roll-controlled and wp, vp are not included here.

According to the above mentioned assumptions the equations for the model are defined with the corresponding simplifications in (2.7),(2.8) and (2.9). As both horizontal and vertical lateral motions are symmetric in format, both will be dealt with together, taking into account the appropriate sign changes in derivatives for each lateral direction.

The vertical motion is defined by:

$$\begin{aligned}
\dot{q} &= m_q q + m_w w + m_\xi \xi + m_\eta \eta \\
&= I_y^{-1}(\frac{1}{4}\rho V_o S d^2 C_{m_q} q + \frac{1}{2}\rho V_o S d C_{m_w} w + \frac{1}{2}\rho V_o^2 S d C_{m_\xi}\xi + \frac{1}{2}\rho V_o^2 S d C_{m_\eta}\eta) \\
&= \frac{1}{2} I_y^{-1}\rho V_o S d(\frac{1}{2} d C_{m_q} q + C_{m_w} w + V_o C_{m_\xi}\xi + V_o C_{m_\eta}\eta) \\
\dot{w} &= m^{-1}(z_w w + z_q q + z_\xi \xi + z_\eta \eta) + uq \\
&= m^{-1}(\frac{1}{2}\rho V_o S C_{z_w} w + \frac{1}{2}\rho V_o^2 S C_{z_\xi}\xi + \frac{1}{2}\rho V_o^2 S C_{z_\eta}\eta) + uq \\
&= \frac{1}{2} m^{-1}\rho V_o S(C_{z_w} w + V_o C_{z_\xi}\xi + V_o C_{z_\eta}\eta) + uq
\end{aligned} \tag{2.7}$$

The horizontal motion is defined by:

$$\begin{aligned}
\dot{r} &= n_r r + n_v v + n_\xi \xi + n_\zeta \zeta \\
&= I_z^{-1}(\frac{1}{4}\rho V_o S d^2 C_{n_r} r + \frac{1}{2}\rho V_o S d C_{n_v} v + \frac{1}{2}\rho V_o^2 S d C_{n_\xi}\xi + \frac{1}{2}\rho V_o^2 S d C_{n_\zeta}\zeta) \\
&= \frac{1}{2} I_z^{-1}\rho V_o S d(\frac{1}{2} d C_{n_r} r + C_{n_v} v + V_o C_{n_\xi}\xi + V_o C_{n_\zeta}\zeta) \\
\dot{v} &= m^{-1}(y_v v + y_r r + y_\xi \xi + y_\zeta \zeta) - ur \\
&= m^{-1}(\frac{1}{2}\rho V_o S C_{y_v} v + \frac{1}{2}\rho V_o^2 S C_{y_\xi}\xi + \frac{1}{2}\rho V_o^2 S C_{y_\zeta}\zeta) - ur \\
&= \frac{1}{2} m^{-1}\rho V_o S(C_{y_v} v + V_o C_{y_\xi}\xi + V_o C_{y_\zeta}\zeta) - ur
\end{aligned} \tag{2.8}$$

and the roll motion by:

$$\begin{aligned}
\dot{p} &= l_p p + l_\zeta \zeta + l_\eta \eta + l_\xi \xi \\
&= I_x^{-1}(\frac{1}{4}\rho V_o S d^2 C_{l_p} p + \frac{1}{2}\rho V_o^2 S d C_{l_\zeta}\zeta + \frac{1}{2}\rho V_o^2 S d C_{l_\eta}\eta + \frac{1}{2}\rho V_o^2 S d C_{l_\xi}\xi) \\
&= \frac{1}{2} I_x^{-1}\rho V_o S d(d C_{l_p} p + V_o C_{l_\zeta}\zeta + V_o C_{l_\eta}\eta + V_o C_{l_\xi}\xi)
\end{aligned} \tag{2.9}$$

where the axes (x, y, z), rates (r, q, p) and velocities (u, v, w) are defined in fig. 2.2 and where ζ, η, ξ are the inputs to the system and are defined in the Appendix A. Equations (2.7), (2.8) and (2.9) describe the dynamics of the body rates and velocities under the influence of external forces (C_{y_v}) and moments (C_{n_v}) acting on the frame. These forces and moments derived from a wind-tunnel measurements are non-linear functions of Mach number, longitudinal and lateral velocities, control surface deflection, aerodynamic roll angle and body rates. The aerodynamic coefficients, $(C_{y_v}, C_{y_\zeta}, X_{cp}$ and $C_{n_r})$, are presented by polynomials shown in the next section. The physical parameters of the Horton Missile are shown in the Appendix B:

2.3 Aerodynamic coefficients for different flight conditions

The aerodynamic coefficients $(C_{y_v}, C_{y_\zeta}, X_{cp}$ and $C_{n_r})$ are presented by polynomials for 0° and 45° roll angles. These polynomials are fitted to the set of curves taken from look-up tables for different flight conditions. The look-up tables are a set of curves in the plane of total incidence, σ, and Mach number, M.

Centre of Gravity X_{cg} and Centre of Pressure X_{cp}

The centre of gravity is given by the formula:

$$x_{cg} = 1.3 + \frac{m}{500} \tag{2.10}$$

where m is the mass of the missile. The polynomial for the *Centre of Pressure* for different roll angles is given by:

$$\begin{aligned} x_{cp_0} &= 1.3 + 0.1M + 0.2|\sigma| \\ x_{cp_{45}} &= 1.3 + 0.1M + 0.3|\sigma| \end{aligned} \tag{2.11}$$

Side-slip Normal Force Coefficient - C_{z_w}, C_{y_v}

A set of normal force curves due to side-slip velocity in the plane of incidence for aerodynamic roll angles of 0° and 45° are given by the polynomial (2.12), where M is the Mach number and σ is the total incidence.

$$\begin{aligned} C_{y_v} &= C_{z_w} \\ C_{z_{w0}} &= -25 + 1.0M - 60|\sigma| \\ C_{z_{w45}} &= -26 + 1.5M - 30|\sigma| \end{aligned} \tag{2.12}$$

Missile Rate Normal Force Coefficient - C_{y_r}

This coefficient is normally small [70], as given by Blakelock, and does not effect the dynamic response of the missile significantly. It is assumed to be zero in this study. Hence:

$$C_{y_r} = 0 \tag{2.13}$$

Fin Normal Force Coefficient - C_{y_ζ}, C_{z_η}

The rudder and elevator control forces are proportional to fin angle, and are expressed as derivatives which are functions of incidence,σ, Mach number, M, and aerodynamic roll angle, λ. A set of derivatives for roll angle of $\lambda = 0^o$ and $\lambda = 45^o$ are given by the polynomials in (2.14).

$$\begin{aligned} C_{y_\zeta} &= C_{z_\eta} \\ C_{z_{\eta 0}} &= -10 - 1.6M + 2.0|\sigma| \\ C_{z_{\eta 45}} &= -10 - 1.4M + 1.5|\sigma| \end{aligned} \tag{2.14}$$

Side-slip and Control Moments C_{n_v}, C_{m_w}

The yawing and pitching moment coefficients are derived from the normal force coefficients ($C_{y_v}, C_{y_\zeta}, C_{y_\xi}$). The static margin ($x_{sm}$), fin moment arm ($x_{sf}$) for lateral motion and roll moment arm (x_{sr}) for roll motion are as follows:

$$\begin{aligned} C_{n_v} &= s_m C_{y_v} \\ C_{n_\zeta} &= s_f C_{y_\zeta} \\ C_{n_\xi} &= s_r C_{y_\xi} \end{aligned} \tag{2.15}$$

The static margin, x_{sm}, is defined as the difference between the centre of gravity position, x_{cg}, and the centre of pressure position, X_{cp}, measured from the nose of the missile. Similarly the fin moment arm, x_{fm}, is defined as the difference between the centre of gravity position, x_{cg}, and the centre of pressure of the fin, x_f, again measured from the nose of the missile. Hence:

$$\begin{aligned} s_m &= d^{-1} x_{sm} \\ s_f &= d^{-1} x_{sf} \\ s_r &= d^{-1} x_{sr} \end{aligned} \tag{2.16}$$

where

$$\begin{aligned} x_{sm} &= (x_{cg} - X_{cp}) \\ x_{sf} &= (x_f - x_{cg}) \\ x_{sr} &= 1.5\frac{d}{2} \end{aligned} \tag{2.17}$$

and $x_f = 2.6m$ and the missile reference diameter is given by $d = 0.2m$. The roll moment arm, x_{sr}, is assumed to be about 1.5 times the radius of the missile.

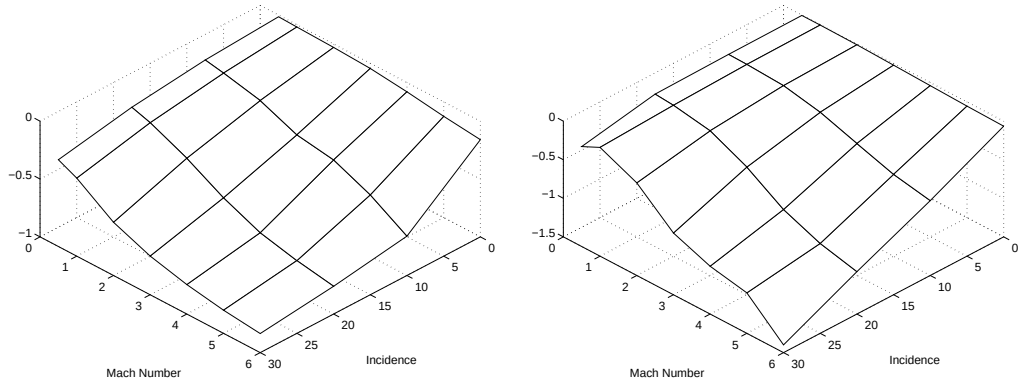
Damping Moment Coefficients - $C_{n_r}, C_{m_q}, C_{l_p}$

The yawing and pitching damping moments are proportional to body rate and are also expressed as a derivative. This moment contribution is small compared to other sources and is modelled as independent of aerodynamic roll angle. It displays variation with Mach number, M, and incidence, σ, and is defined by the polynomial:

$$\begin{aligned} C_{n_r} &= C_{m_q} \\ C_{m_q} &= -500 - 30M + 200|\sigma| \end{aligned} \tag{2.18}$$

The roll damping moment is undefined from BADL data. For this study it has been arbitrarily set at:

$$C_{l_p} = -500 \tag{2.19}$$



a)Rudder/Aileron Roll Coupling b)Aileron/Rudder Yaw Coupling

Figure 2.3: Fin Coupling Moment coefficients

Fin Coupling Moment Coefficients - C_{l_η}, C_{n_ξ}

These are derived from BADL data supplied in the form of simple relationships relating aileron, rudder and elevator moments. They are given as a ratio of rudder/aileron roll coupling:

$$\frac{l_\eta}{l_\xi} = \frac{C_{l_\eta}}{C_{l_\xi}} \tag{2.20}$$

and a ratio of aileron/rudder yaw coupling:

$$\frac{n_\xi}{n_\eta} = \frac{C_{n_\xi}}{C_{n_\eta}} \tag{2.21}$$

These are shown in fig. 2.3, left and right respectively. Both coupling moments are functions of Mach Number and incidence.

2.4 Polynomial form

The equations defined in the parametric model in the previous section can be represented in polynomial form by the following set of equations using the dynamic parameters, incidence angle, σ, Mach number, M, and aerodynamic roll angle, λ.

Aerodynamic Roll Angle Interpolation

Several of the coefficients are functions of aerodynamic roll angle. The parametric relationships are given at roll angles of $0°$ and $45°$. Horton uses a sinusoidal interpolation technique which can be modelled by the relationship:

$$C_{ij} = 0.5(C_{ij}^{0}\lambda^{0} + C_{ij}^{45}\lambda^{45}) \tag{2.22}$$

where:

$$\begin{aligned} \lambda^{0} &= (1 + \cos(4\lambda)) \\ \lambda^{45} &= (1 - \cos(4\lambda)) \end{aligned} \tag{2.23}$$

and the coefficients C_{ij}^{0} and C_{ij}^{45} are the parametric equations at 0^o and 45^o respectively. This interpolation is used in the polynomial fit for aerodynamic roll dependent coefficients.

Centre of Pressure and Centre of Gravity

The centre of gravity is modelled by the polynomial equation:

$$x_{cg} = x_{cg_0} \tag{2.24}$$

where:

Coefficient	Value
x_{cg_0}	$1.3 + \frac{m}{500}$

This is a copy of the parametric relationship and does not involve any polynomial fitting. The centre of pressure is a function of the aerodynamic roll angle, λ. Using the aerodynamic roll angle interpolation technique, it can be modelled by the polynomial equation:

$$\begin{aligned} X_{cp}(M,\lambda) &= X_{cp_0} + X_{cp_M}M + X_{cp_\sigma}(\lambda)|\sigma| \\ X_{cp_\sigma}(\lambda) &= X_{cp_\sigma}^{0}\lambda^{0} + X_{cp_\sigma}^{45}\lambda^{45} \end{aligned} \tag{2.25}$$

or:

$$X_{cp}(M,\lambda) = X_{cp_0} + X_{cp_M}M + X_{cp_\sigma}^{0}\lambda^{0}|\sigma| + X_{cp_\sigma}^{45}\lambda^{45}|\sigma| \tag{2.26}$$

where:

Coefficient	Value
X_{cp_0}	1.3
X_{cp_M}	0.1
$X^0_{cp_\sigma}$	0.2
$X^{45}_{cp_\sigma}$	0.3

The carpet plot for this function is shown in fig. 2.4a, plotted as a function of incidence and roll angle for different mach numbers, with 2.4b plotted as a function of Mach number and roll angle for different incidence angles, and 2.4c plotted as a function of Mach number and incidence against different roll angles.

The static margin and the fin moment arm can thus be modelled in polynomial form as:

$$\begin{aligned} s_m(M,\lambda) &= d^{-1}(X_{cp_0} + X_{cp_M}M + X^0_{cp_\sigma}\lambda^0|\sigma| + X^{45}_{cp_\sigma}\lambda^{45}|\sigma| - x_{c_{g0}}) \\ s_f &= d^{-1}(x_f - x_{c_{g0}}) \end{aligned} \tag{2.27}$$

where:

Coefficient	Value
x_f	2.6
d	0.2

Side-slip Normal Force Coefficients - C_{z_w}, C_{y_v}

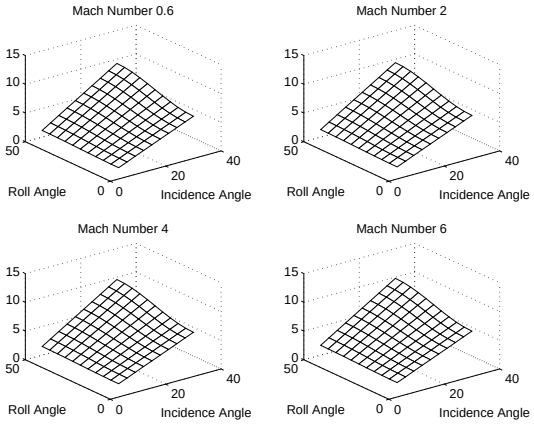
The polynomial equations defining the side-slip normal force are given by:

$$\begin{aligned} C_{z_w} &= 0.5(C^0_{z_w}\lambda^0 + C^{45}_{z_w}\lambda^{45}) \\ C^0_{z_w} &= C^0_{z_{w0}} + C^0_{z_{wM}}M + C^0_{z_{w\sigma}}|\sigma| \\ C^{45}_{z_w} &= C^{45}_{z_{w0}} + C^{45}_{z_{wM}}M + C^{45}_{z_{w\sigma}}|\sigma| \end{aligned} \tag{2.28}$$

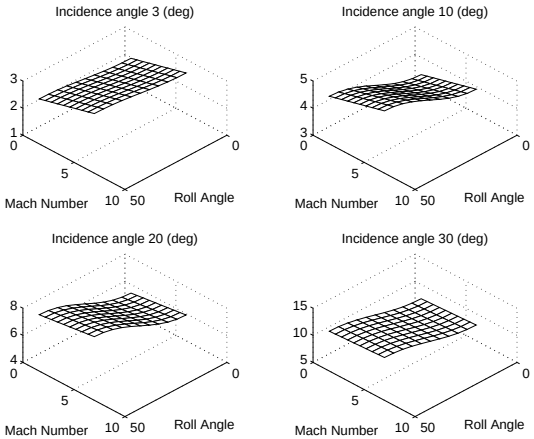
where:

Coefficient	Value
$C^0_{z_{w0}}$	-25
$C^0_{z_{wM}}$	1
$C^0_{z_{w\sigma}}$	-60
$C^{45}_{z_{w0}}$	-26
$C^{45}_{z_{wM}}$	1.5
$C^{45}_{z_{w\sigma}}$	-30

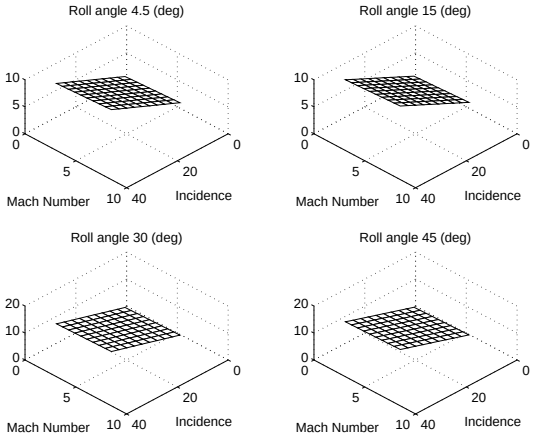
The carpet plot for this function is shown in figures 2.5a, 2.5b, and 2.5c in the same format as the centre of pressure coefficient.



a) for different Mach Numbers

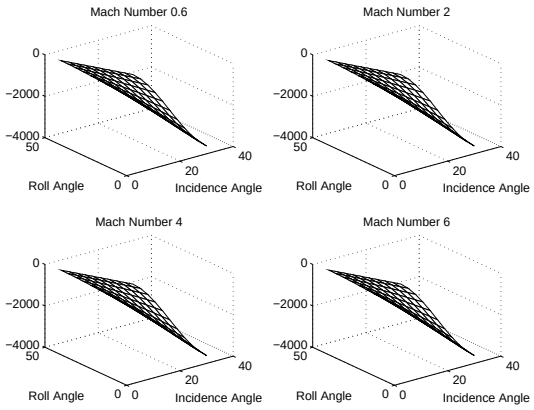


b) for different incidence angles σ

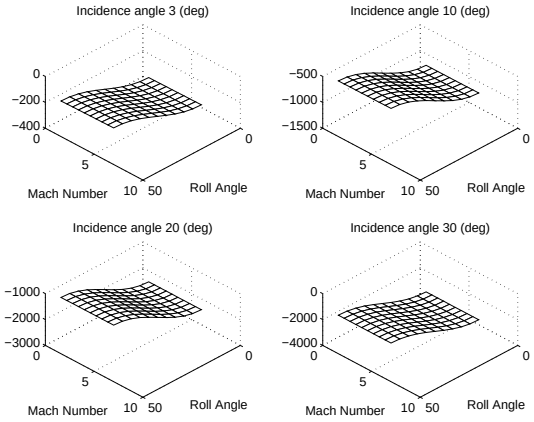


c) for different roll angles λ

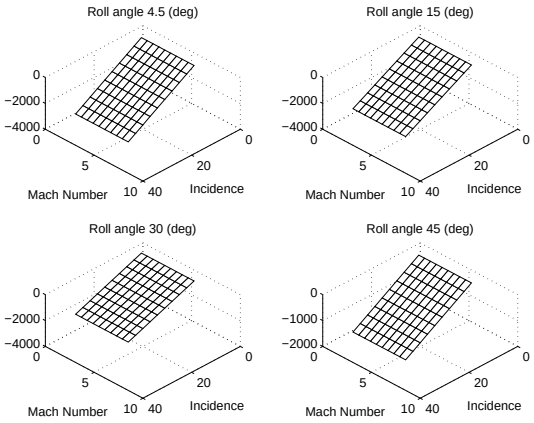
Figure 2.4: Set of four Centre of pressure coefficients X_{cp}



a) for different Mach Numbers



b) for different incidence angles (σ)



c) for different Roll Angles λ

Figure 2.5: Set of four Normal Force coefficients C_{z_w}

Fin Normal Force Coefficients - C_{z_η}, C_{y_ζ}

The fin normal force coefficient is modelled in a similar way to give:

$$\begin{aligned} C_{z_\eta} &= C_{z_{\eta_0}} + 0.5(C^0_{z_\eta}\lambda^0 + C^{45}_{z_\eta}\lambda^{45}) \\ C^0_{z_\eta} &= C^0_{z_{\eta M}} M + C^0_{z_{\eta\sigma}}|\sigma| \\ C^{45}_{z_\eta} &= C^{45}_{z_{\eta M}} M + C^{45}_{z_{\eta\sigma}}|\sigma| \end{aligned} \tag{2.29}$$

where:

Coefficient	Value
$C_{z_{\eta 0}}$	10
$C^0_{z_{\eta M}}$	-1.6
$C^0_{z_{\eta\sigma}}$	2
$C^{45}_{z_{\eta M}}$	-1.4
$C^{45}_{z_{\eta\sigma}}$	1.5

The carpet plot for this function is shown by White [71], in the same format as the centre of pressure and side-slip normal force coefficient.

Damping Moment Coefficients - $C_{m_q}, C_{n_r}, C_{l_p}$

The lateral moments can be modelled directly in polynomial form as:

$$C_{m_q} = C_{m_{q0}} + C_{m_{qM}} M + C_{m_{q\sigma}}|\sigma| \tag{2.30}$$

where:

Coefficient	Value
$C_{m_{q0}}$	-500
$C_{m_{qM}}$	-30
$C_{m_{q\sigma}}$	200

The carpet plot for this function is shown in fig. 2.6. The roll damping coefficient is:

$$C_{l_p} = C_{l_{p_0}} \tag{2.31}$$

where:

Coefficient	Value
C_{l_0}	-500

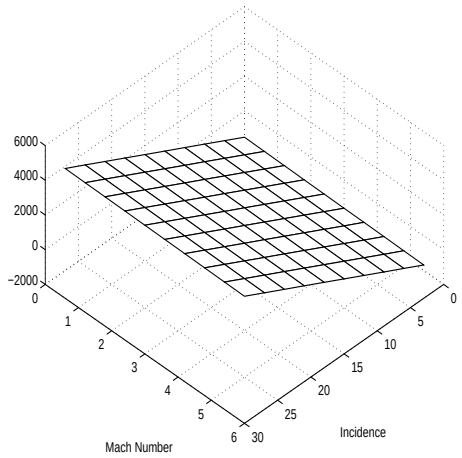


Figure 2.6: Roll damping coefficient C_{m_q}

2.5 Open-Loop Stability Analysis

The open-loop stability analysis of the system is divided into two steps. In the first one we consider the lateral flight control design, i.e. we study the single-input single-output system. The spectrum of the poles and zeros for the open-loop single-input single-output system (SISO) is shown in figures 2.7, 2.8 for different flight conditions. In this case all the aerodynamic coefficients are described by affine polynomials of incidence, σ, and Mach number, M. In order to examine the effect on the system of those two variables, a spectrum of poles and zeros for constant Mach number (Mach number = 3) and varying σ (up to 30°) is considered. Then a spectrum of poles and zeros for Mach number varying from 2 to 4 is examined, while a constant value of $\sigma = 0.1°$ is maintained.

Fig. 2.7 shows the open-loop stability for large variations of total incidence. For most of the regime the missile is statically stable, as given by Horton [69]. For low values of speed, less than Mach 2, the airframe becomes statically unstable see fig. 2.8. Also fig. 2.8 shows the operating envelope of large variations in Mach number, which is the indication for forward speed of the missile. The incidence is used as a state variable so it is important to show the open loop stability for the operating range. The control law of the autopilot design is derived for variations in incidence of 0.1° to 1° and fixed Mach number = 3, as is detailed in Chapter 3 and Chapter 4.

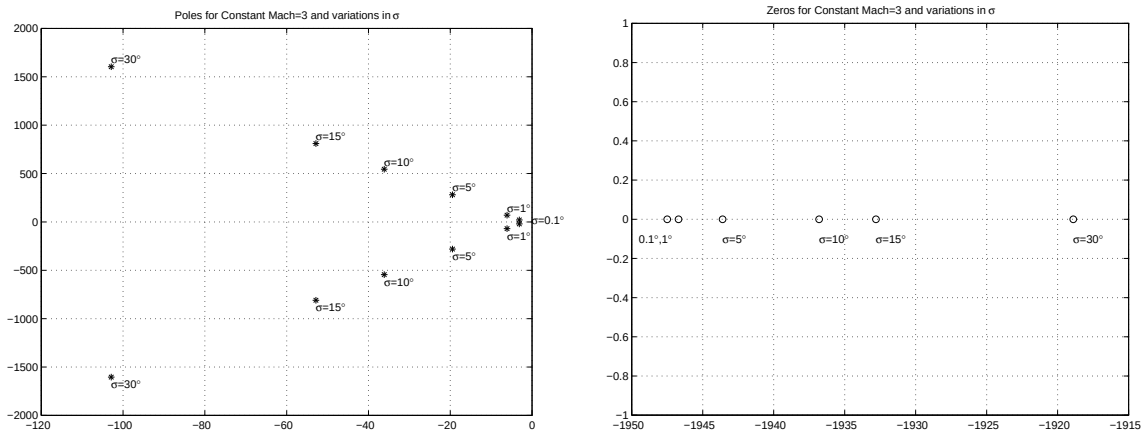


Figure 2.7: Pole and Zero Spectrum for constant $Mach = 3$ and varying incidence

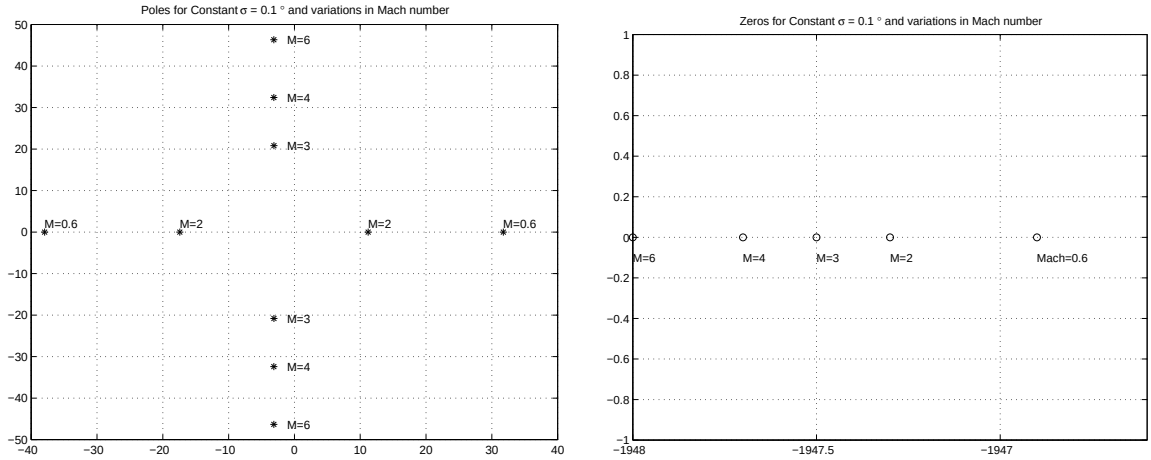


Figure 2.8: Pole and Zero Spectrum for constant incidence = 0.1° and varying Mach number

2.6 Cross-coupling effect

It is necessary to determine how strong the coupling effect is between the different channels (yaw, pitch and roll). The simulation results for a step input demand only on fin angle ξ are shown in fig. 2.9. If the system is not coupled, an input demand on fin angle ξ should have no effect on pitch $(\dot{w}, \dot{q})$ or yaw $(\dot{v}, \dot{r})$ channel, which is not the case here. The other case for a step input demand only on fin angle ζ, is examined too. The simulation results for all inputs and state variables are shown in fig. 2.10 (e.g. the system is again excited through a single channel - yaw). Again if the system is not coupled demand in fin angle ζ should have no effect on the other two channels. However we can well see the coupling effect distributed along the other two channels. The responses for symmetrical velocities side-slip, v, and vertical, w, again prove the symmetry of both channels. A strong, severe cross-coupling between all three channels has been demonstrated.

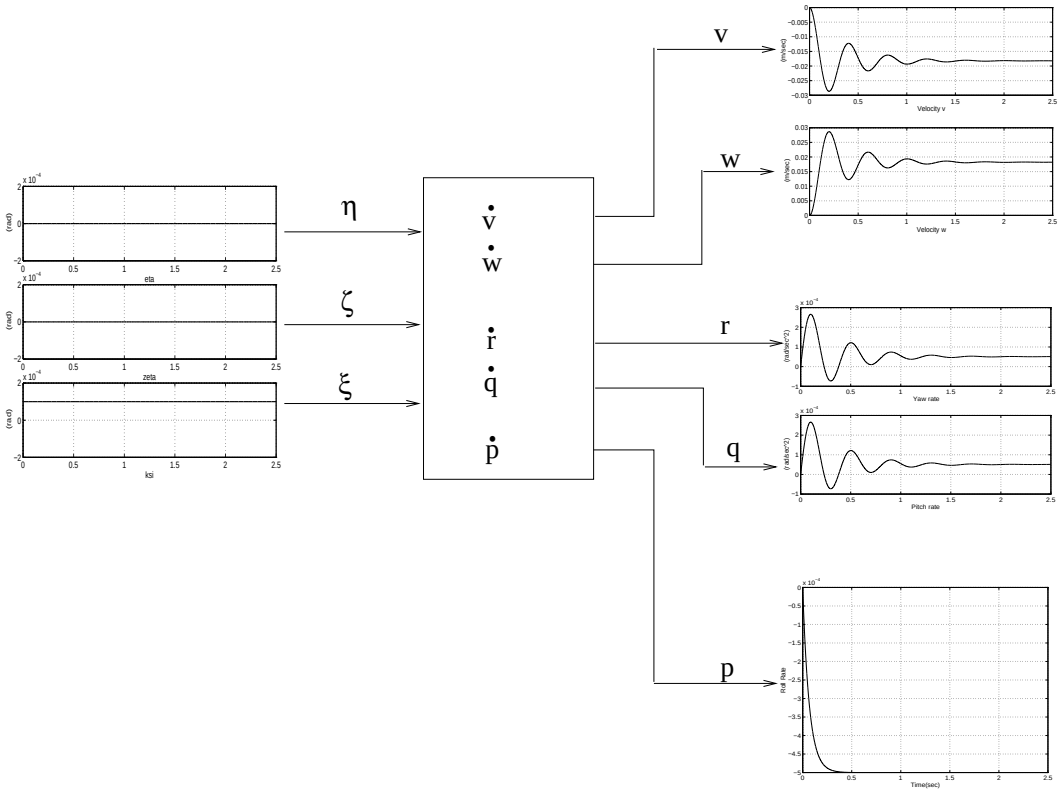


Figure 2.9: Side-slip velocities and rates responses for a demand in fin angle ξ

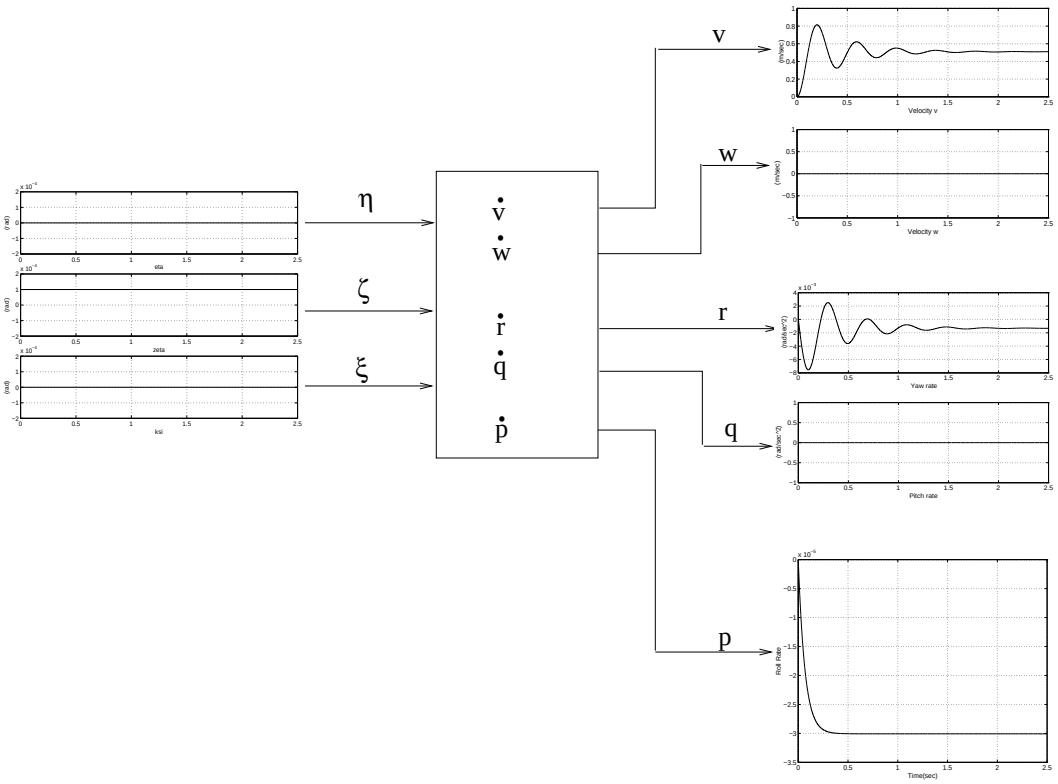


Figure 2.10: Side-slip velocities and rates responses for a demand in fin angle ζ

2.7 Nonlinearity

The nonlinear behaviour of the system is inherent for a highly manoeuvrable missile. It is caused by the complex dynamics of its motion. Attention has been paid to how a certain fin angles ζ or ξ can affect the side-slip velocity or lateral acceleration responses and demonstrated in fig. 2.11. Let us now consider the open loop dynamics

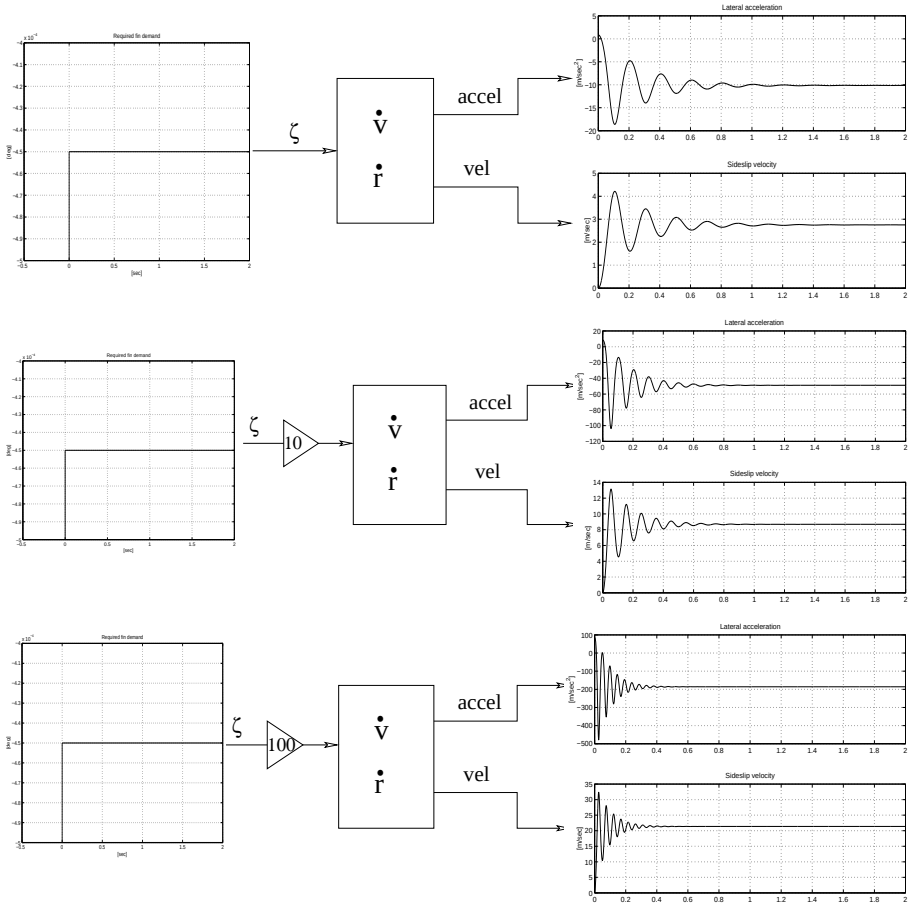


Figure 2.11: 1g, 5g, 20g demand

for a SISO system in the yaw channel. The non-linear differential equations for $\dot{v}$ and $\dot{r}$ are described earlier in (2.8). If a constant input demand is required, then the missile will accelerate at 10 m/sec^2 with the corresponding side-slip velocity of 2.5 m/sec shown in fig. 2.11-top. Increasing the input demand to the rudder by 10 or 100 times does not produce a proportionate response in the acceleration and velocity as it would in a linear system. These simulations are also a demonstration for two types of nonlinearities: input to state (ζ to v) and state to controlled output (v to a_v). The latter relationship is given by the following dynamic equation:

$$\begin{aligned}
\alpha &= \dot{v} + Ur \\
\alpha &= V^o(C_{yv}v + V_oC_{y\zeta}\zeta) - Ur + Ur \\
&= V^o[(C_{yv_0} + C_{yv_M}M + C_{yv_\sigma} \mid \sigma \mid)v + V_o(C_{y\zeta_0} + C_{y\zeta_M}M + C_{y\zeta_\sigma} \mid \sigma \mid)\zeta] \\
&= V^o[(\bar{C}_{yv}v + \bar{C}_{yv_\sigma} \mid v \mid v + V_o\bar{C}_{y\zeta}\zeta + V_o\bar{C}_{y\zeta_\sigma} \mid v \mid \zeta]
\end{aligned} \tag{2.32}$$

and is also an indication that it is possible to achieve lateral acceleration control indirectly through the side-slip velocity which is further addressed in Chapter 3. Also the responses of the open loop system are settled within $1.2s$ and much faster $0.5s$ for higher demands, whereas the closed loop requirement for settling time of the response is around $0.12s$.

It is important to understand and effectively control the nonlinear behaviour of the system as the missile manoeuvres in a large dynamic range and changes speed continually.

2.8 Multi-modelling airframe dynamics

2.8.1 Parametric uncertainties

The modelling errors can be separated into two types: parametric and unstructured. Parametric uncertainty refers to modelling errors, under the assumption that the actual plant is of the same order as the model, where the numerical values of the coefficients to the differential equation, which are related to the physical parameters of the system, between the actual plant and the model are different. In the case of unstructured uncertainty, the modelling errors refer to the difference in the dynamics between the finite dimensional model and the unknown and possibly infinite dimensional actual process.

The uncertainties we are dealing with are parametric and structured, but we cannot measure them. We know where they may come from but we are not certain which ones are causing the model parameters variations. For example, in a real flight scenario, for every instance of this missile type, the aerodynamic functions taken in wind tunnel measurements may deviate from their nominal values. The variations are parametric uncertainties of the non-linear system. In the presence of parametric uncertainties the state-space form of the non-linear system can be written in a compact format as:

$$\begin{aligned}
\dot{x} &= f(x) + \triangle f(x) + (g(x) + \triangle g(x))u \\
y &= h(x)
\end{aligned} \tag{2.33}$$

The reduced order system for yaw plane without roll coupling has been considered:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} f_{x_1}(x) + \triangle f_{x_1}(x) \\ f_{x_2}(x) + \triangle f_{x_2}(x) \end{bmatrix} + \begin{bmatrix} g_{x_1}(x) + \triangle g_{x_1}(x) & 0 \\ g_{x_2}(x) + \triangle g_{x_2}(x) & 0 \end{bmatrix} + \begin{bmatrix} u_1 \end{bmatrix} \tag{2.34}$$

A set of convex models is produced that map the vertex points in a high order parameter space (of the order of 16 variables) shown in detail in equations: (2.35) and (2.36). The multiple model description of the airframe aerodynamics can be further expressed in a parametric form as:

$$\begin{aligned} \dot{x}_1 &= (a_1 + \triangle a_1)x_1 + (a_2 + \triangle a_2)x_1^2 + (a_3 + \triangle a_3)x_2 + ((a_4 + \triangle a_4)x_1 + a_5 + \triangle a_5)u_1 \\ \dot{x}_2 &= (b_1 + \triangle b_1)x_1^3 + (b_2 + \triangle b_2)x_1^2 + (b_3 + \triangle b_3)x_1 + (b_4 + \triangle b_4)x_1x_2 + (b_5 + \triangle b_5)x_2 \\ &+ ((b_6 + \triangle b_6)x_1 + b_7 + \triangle b_7)u_1 \end{aligned}$$

For the equations of lateral velocities $\dot{v}$ and $\dot{w}$, the parameters $\triangle a_1, \ldots, \triangle a_5$ are shown in equation (2.35). For the equations of yaw $\dot{r}$ and pitch $\dot{q}$ rates, the parameters $\triangle b_1, \ldots, \triangle b_7$ are shown in equation (2.36). Both a_i and b_i are functions of the aerodynamic coefficients: $a_i, b_i = f(C_{y_v}, X_{cp}, C_{y_\zeta}, C_{n_r})$ and can take any values randomly generated within the vertex points. Hence more than 1000 models can be exercised and the control system tested for robustness.

$$\begin{aligned} a_1 + \triangle a_1 &= \frac{1}{2m}\rho V_o S(\bar{C}_{yv_0} + \triangle \bar{C}_{yv_0}) \\ a_2 + \triangle a_2 &= \frac{1}{2m}\rho V_o S(\bar{C}_{yv_\sigma} + \triangle \bar{C}_{yv_\sigma}) \\ a_3 + \triangle a_3 &= U_f \\ a_4 + \triangle a_4 &= \frac{1}{2m}\rho V_o^2 S(\bar{C}_{y\zeta_\sigma} + \triangle \bar{C}_{y\zeta_\sigma}) \\ a_5 + \triangle a_5 &= \frac{1}{2m}\rho V_o^2 S(\bar{C}_{y\zeta_0} + \triangle \bar{C}_{y\zeta_0}) \end{aligned} \tag{2.35}$$

$$\begin{aligned} b_1 + \triangle b_1 &= -(\frac{1}{2I_{yz}})\rho V_o S(\bar{X}_{cp_\sigma} + \triangle \bar{X}_{cp_\sigma})(\bar{C}_{yv_\sigma} + \triangle \bar{C}_{yv_\sigma}) \\ b_2 + \triangle b_2 &= -(\frac{1}{2I_{yz}})\rho V_o S((\bar{X}_{cp_0} + \triangle \bar{X}_{cp_0})(\bar{C}_{yv_\sigma} + \triangle \bar{C}_{yv_\sigma}) + (\bar{X}_{cp_\sigma} + \triangle \bar{X}_{cp_\sigma})(\bar{C}_{yv_0} + \triangle \bar{C}_{yv_0})) \\ b_3 + \triangle b_3 &= -(\frac{1}{2I_{yz}})\rho V_o S(\bar{X}_{cp_0} + \triangle \bar{X}_{cp_0})(\bar{C}_{yv_0} + \triangle \bar{C}_{yv_0}) \\ b_4 + \triangle b_4 &= (\frac{d^2}{4I_{yz}})\rho V_o S(\bar{C}_{nr_\sigma} + \triangle \bar{C}_{nr_\sigma}) \\ b_5 + \triangle b_5 &= (\frac{d^2}{4I_{yz}})\rho V_o S(\bar{C}_{nr_0} + \triangle \bar{C}_{nr_0}) \\ b_6 + \triangle b_6 &= (\frac{d}{2I_{yz}})\rho V_o^2 S S_f(\bar{C}_{y\zeta_\sigma} + \triangle \bar{C}_{y\zeta_\sigma}) \\ b_7 + \triangle b_7 &= (\frac{d}{2I_{yz}})\rho V_o^2 S S_f(\bar{C}_{y\zeta_0} + \triangle \bar{C}_{y\zeta_0}) \end{aligned} \tag{2.36}$$

The tables shown below represent the polynomials for the aerodynamic coefficients in the supersonic range for different roll angles 0° and 45°. They are a set of curves in the plane of total incidence, σ and Mach number, M. In these tables the C_{y_v} polynomials present the *normal force* curves, the X_{cp} present the *centre of pressure* curves, C_{y_ζ} present the *rudder and elevator control forces* curves, and finally the C_{n_r} present the *damping yawing and pitching moments* curves which are reasonably proportional to body rates.

Normal force	$C_{y_v} = -25 + 1.0M - 60\sigma$
Control surfaces	$C_{y_\zeta} = -10 - 1.6M + 2.0\sigma$
Centre of pressure	$X_{cp} = 1.3 + 0.1M + 0.2\sigma$
Damping moment	$C_{n_r} = -500 - 30M + 200\sigma$

Table 2.1: Roll angle = 0°

Normal force	$C_{y_v} = -26 + 1.5M - 30\sigma$
Control surfaces	$C_{y_\zeta} = -10 - 1.4M + 1.5\sigma$
Centre of pressure	$X_{cp} = 1.3 + 0.1M + 0.3\sigma$
Damping moment	$C_{n_r} = -500 - 30M + 200\sigma$

Table 2.2: Roll angle = 45°

where:

$$\begin{aligned} C_{y_v} &= \bar{C}_{yv_0} + \bar{C}_{yv_\sigma} \\ \bar{C}_{yv_0} &= C_{yv_0} + C_{yv_M}\frac{V_o}{SoS} \\ \bar{C}_{yv_\sigma} &= C_{yv_\sigma}\frac{180}{V_o\pi} \end{aligned} \tag{2.37}$$

$$\begin{aligned} C_{y_\zeta} &= \bar{C}_{y\zeta_0} + \bar{C}_{y\zeta_\sigma} \\ \bar{C}_{y\zeta_0} &= C_{y\zeta_0} + C_{y\zeta_M}\frac{V_o}{SoS} \\ \bar{C}_{y\zeta_\sigma} &= C_{y\zeta_\sigma}\frac{180}{V_o\pi} \end{aligned} \tag{2.38}$$

$$\begin{aligned} C_{n_r} &= \bar{C}_{nr_0} + \bar{C}_{nr_\sigma} \\ \bar{C}_{nr_0} &= C_{nr_0} + C_{nr_M}\frac{V_o}{SoS} \\ \bar{C}_{nr_\sigma} &= C_{nr_\sigma}\frac{180}{V_o\pi} \end{aligned} \tag{2.39}$$

$$
\begin{aligned}
X_{cp} &= \bar{X}_{cp_0} + \bar{X}_{cp_\sigma} \\
\bar{X}_{cp_0} &= X_{cp_0} + X_{cp_M} \frac{V_o}{SoS} \\
\bar{X}_{cp_\sigma} &= X_{cp_\sigma} \frac{180}{V_o \pi}
\end{aligned} \tag{2.40}
$$

A large excursion on perturbations of the aerodynamic coefficients $(C_{y_v}, C_{y_\zeta}, X_{cp}, C_{n_r})$ has been introduced into the system within the range of 0° to 45° aerodynamic roll angles.

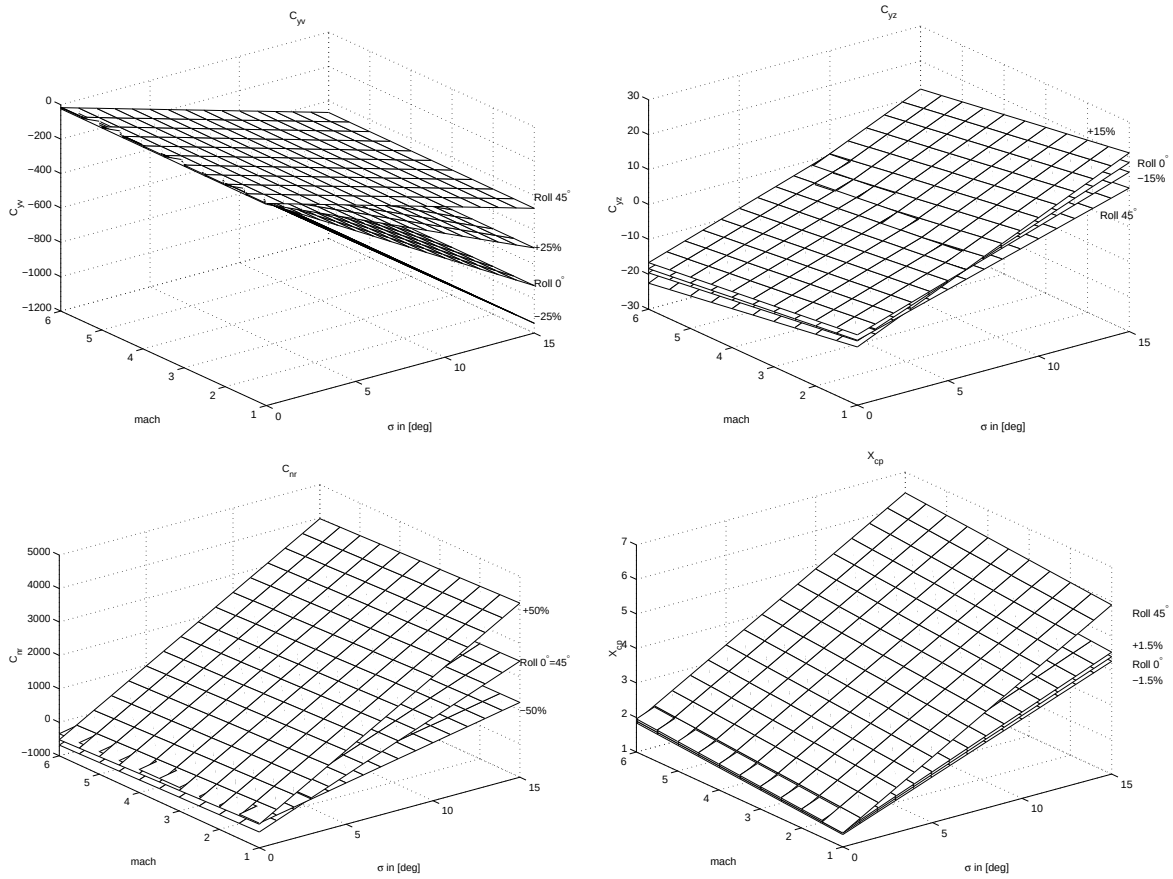


Figure 2.12: Aerodynamic coefficients ranges

2.8.2 Sensitivity Analysis

The variations in aerodynamic coefficients ($C_{y_v}, C_{y_\zeta}, X_{cp}, C_{n_r}$) have introduced parametric uncertainties into the non-linear system. They are shown in fig. 2.12 and the polynomials are presented in tables 2.1 and 2.2. In order to explore the complexity of the problem we have assessed the open and closed loop system performance for different autopilot demands (1g, 5g, 10g, 15g) and we have examined the amount of perturbations allowed in each coefficient before the system's behaviour goes unstable or exceeds 10% steady state error on side-slip velocity. For simplicity we have studied the single plane (lateral or vertical motion) when the roll angle is 0°. A set of models of vertex points is shown on fig. 2.13 for 10 combinations of the four aerodynamic coefficients. Since we have determined that changes in the coefficient, C_{n_r}, does not affect missile stability, only eight combinations of (min/max) ranges are considered, one random set and one with the nominal coefficients. So a 1000 models can be generated randomly within the (min/max) ranges and tested for robustness. Also the side-slip velocity and acceleration responses of the open loop system are shown in fig. 2.13. Up to 40% deviation from the nominal value of side-slip velocity response and up to 55% deviation from the nominal value of lateral acceleration response has been found for a large range of unit step demands on rudder or elevator (e.g. 1g, 5g, 10g, 15g).

It has been found that some coefficients can allow larger percentage variation from the nominal case than others. Within the system we are able to tolerate ±50% uncertainty in each of $C_{y_v}, C_{y_\zeta}, C_{n_r}$ before the system dynamics goes unstable. Also, the aerodynamic coefficient C_{y_v} can vary by up to ±25% before the side-slip velocity exceeds 10% steady state error within the feedback linearized loop. For similar performance, C_{y_ζ} can vary by up to ±15%, and the most sensitive coefficient, X_{cp}, can vary by ±1.5%. These are all found by extensive simulations. The centre of pressure coefficient X_{cp} and the control surface coefficient C_{y_ζ} have most significant effect on the closed loop performance (the system is very sensitive to small changes), while the damping moment contribution in C_{n_r} is small and the system is almost insensitive to it and can be assumed independent of the aerodynamic roll angle.

The sign of the static margin $x_{sm} = x_{cg} - X_{cp}$ can tell us whether the system is stable or not. The centre of gravity point is at $1.55m$ measured from the nose. For a minimum side-slip velocity demand of 2.7 m/sec, the missile is heading at very little incidence, $\sigma = 0.1°$. For that value of σ, the centre of pressure coefficient is $X_{cp} = 1.62m$ measured from the nose and the static margin $x_{sm} = -0.07$ is negative, hence the airframe is statically stable. A change of -130% in the $\bar{X}_{cp_\sigma}$ term of the X_{cp} coefficient ($X_{cp} = \bar{X}_{cp_0} + \bar{X}_{cp_\sigma}$) is critical for the stability of the missile. This change will move the centre of pressure point to $1.53m$ which will produce a positive static margin of $x_{sm} = 0.02$. Hence when the $\bar{X}_{cp_\sigma}$ term of X_{cp} is varying, the sign of the static margin changes from negative to positive and the missile becomes unstable.

	Cyv	Cy_ζ	Cnr	Xcp	
1	max	max	max	max	
2	min	min	min	min	
3	max	max	max	min	A_{min}
4	max	min	max	max	V_{min}
5	min	max	max	max	
6	min	min	max	max	A_{max}
7	max	min	max	min	
8	min	max	max	min	V_{max}
9	ran	ran	ran	ran	
10	nom	nom	nom	nom	

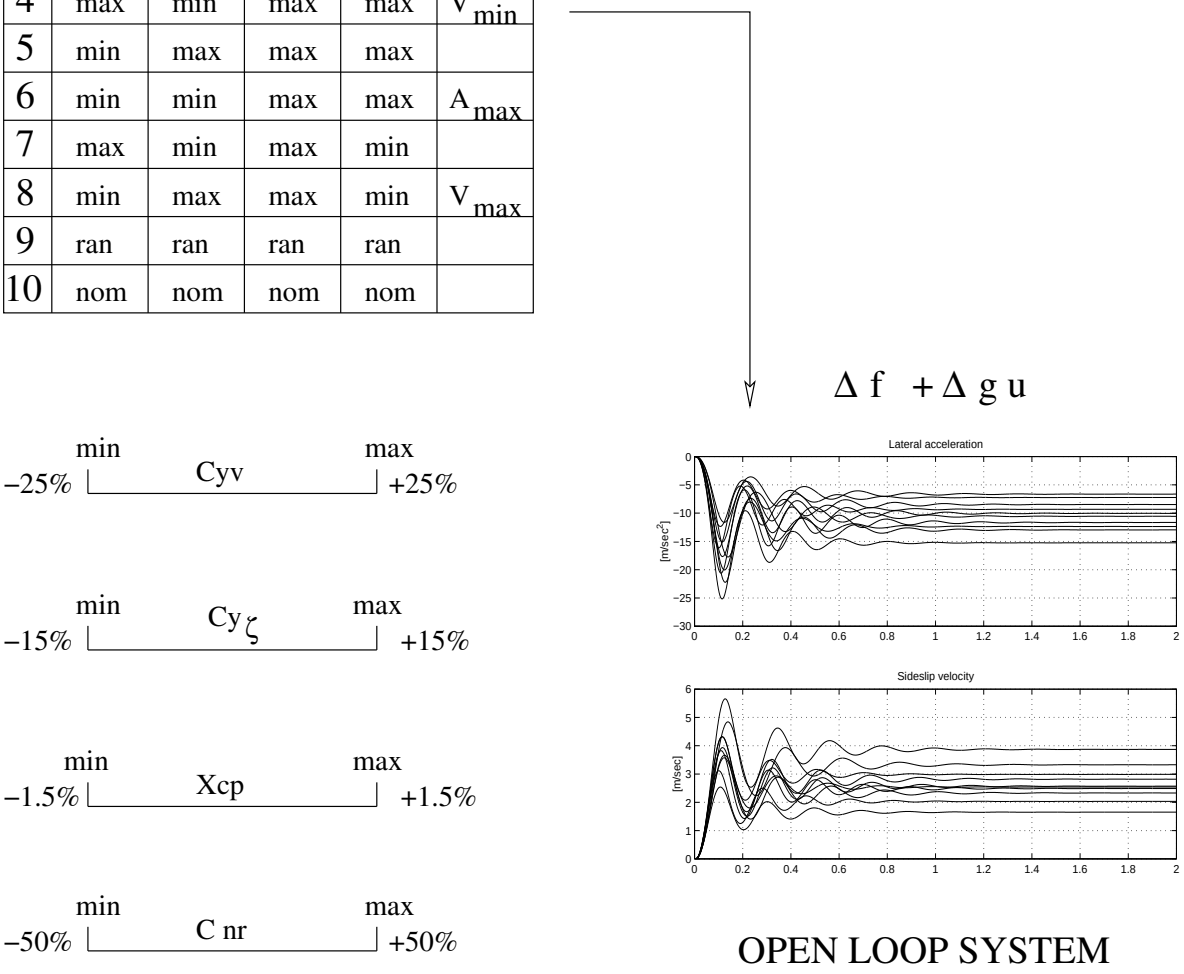


Figure 2.13: Vertex points models

2.9 Cartesian to Polar coordinates

2.9.1 Missile model dynamics in Cartesian coordinates

The equations of motion in respect to the total incidence σ,

$$\sigma = \frac{\sqrt{v^2 + w^2}}{V_o}\frac{180}{\pi} \tag{2.41}$$

are the following:

$$\begin{aligned}
\dot{v} &= f_v(w,v,r) + g_v(w,v,r)\zeta \\
\dot{r} &= f_r(w,v,r) + g_{r_\zeta}(w,v)\zeta + g_{r_\xi}(w,v)\xi \\
\dot{w} &= f_w(v,w,q) + g_w(v,w,q)\eta \\
\dot{q} &= f_q(v,w,q) + g_{q_\eta}(v,w,q)\eta + g_{q_\xi}(v,w)\xi \\
\dot{p} &= f_p(p) + g_{p_\eta}(v,w,)\eta + g_{p_\xi}(v,w)\xi + g_{p_\zeta}(v,w)\zeta
\end{aligned} \tag{2.42}$$

The functions f_v, f_w, f_r, f_q, f_p and g_v, g_w, g_r, g_q, g_p are given by equations (C.1) in Appendix C. These equations will be used to derive the parametric format for the Cartesian multi-input/multi-output system (MIMO) for control synthesis in the next chapter.

2.9.2 Missile model dynamics in Polar coordinates

The great majority of missiles, including the model considered by Horton, have a cruciform cross-section with two pairs of wings and two pairs of control surfaces. The guidance system issues two commands, one up-down and the other left-right and these two commands are fed to the elevators and rudders respectively. However if there is only one set of control surfaces and wings, the commands have to be issued not in Cartesian, but in Polar form. Some missile types can only have one set of wings and if the missile has to manoeuvre to the right and up in polar form the commands are given by the flight direction, z, and the angle of orientation, λ. In other words the missile has to roll through the angle, λ, and manoeuvre in this roll orientation.

The missile system is transformed in Polar coordinates, with the flight direction given by $z = \sqrt{v^2 + w^2}$ and the angle of orientation given by $\lambda = arctan\frac{v}{w}$.

$$\begin{aligned}
\dot{r} &= \frac{1}{2}I_z^{-1}\rho V_o S d(\frac{1}{2}dC_{n_r}r + C_{n_z}z + V_o C_{n_\xi}\xi + V_o C_{n_\zeta}\zeta) \\
\dot{q} &= \frac{1}{2}I_y^{-1}\rho V_o S d(\frac{1}{2}dC_{m_q}q + C_{m_z}z + V_o C_{m_\xi}\xi + V_o C_{m_\eta}\eta)
\end{aligned}$$

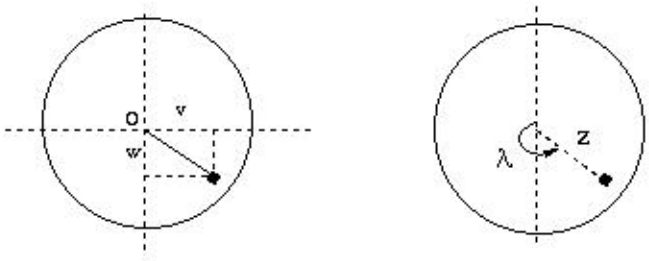


Figure 2.14: Transformation from Cartesian to Polar Coordinates

$$\begin{aligned}
\dot{p} &= \frac{1}{2} I_x^{-1} \rho V_o S d (dC_{l_p} p + V_o C_{l_\zeta} \zeta + V_o C_{l_\eta} \eta + V_o C_{l_\xi} \xi) \\
\dot{z} &= \frac{1}{2} m^{-1} \rho V_o S (C_{yz_z} z + V_o C_{yz_\eta} \eta + V_o C_{yz_\zeta} \zeta) + u(cos(\lambda) q - sin(\lambda) r) \\
\dot{\lambda} &= \frac{1}{2} m^{-1} \rho V_o S (V_o C_{\lambda_\eta} \eta + V_o C_{\lambda_\zeta} \zeta) + u z^{-1} (cos(\lambda) q - sin(\lambda) r)
\end{aligned} \tag{2.43}$$

This will be used to derive the parametric format for control synthesis in the next chapter.

2.10 Closed loop autopilot requirements

The next step would be to design an autopilot system to regulate the motion of the missile such that the commanded manoeuvres generated by the guidance system are followed, hence desired trajectory can be achieved. It is usually preferable to have autopilots with high bandwidths that allow fast and precise command responses.

• It is important to ensure that the autopilot closed-loop dynamics are much faster and better damped than the inherent airframe response. The usual design aim is to achieve autopilot bandwidths that are two to three times faster than the open-loop airframe dynamics. In this case the closed loop time response should be around $0.2s$.

• Closed loop performance: The response of an autopilot must be as fast as possible with the minimum of overshoot so that any command is attained quickly and is of the required magnitude. For low g demands only, a slight overshoot of short duration is usually acceptable, since it can compensate for loss of acceleration during the initial transient. For high g demands, overshoot is usually unacceptable since the airframe structural load limit may be exceeded, or an uncontrollable flight region may be entered. The response character of the autopilot is quantified in terms of rise time, settling time and the maximum percentage overshoot, hence the following desired metrics are required:

• Steady state error accuracy: 2%

• Rise time: ($0.05s$ to $0.07s$);

• Settling time: ($1.2s$ to $1.8s$)

• Maximum percentage overshoot: 10%

• Robustness requirements: The dynamic response of a missile may be simulated by using a suitable mathematical model of the system. This model is usually arrived at through processing wind tunnel data pertaining to the airframe in question (such data are usually subject to experimental and instrumentation errors), from empirical formulae, or from Computational Fluid Dynamics techniques. The resultant dynamic model may differ therefore from the actual dynamics of the missile, due to variations in the aerodynamics, the effects of linearization, unmodelled effects, changes in the flight conditions, or simply build-to-build variations. Any autopilot design must maintain adequate stability and satisfactory performance in the presence of such uncertainties.

• The signals physically available for feedback control such as lateral accelerations, rates and incidence are usually measured by accelerometers, gyroscopes and Pitot tube respectively.

2.11 Conclusions

This chapter has detailed the complexity of the highly non-linear missile system. It is a real research model developed by Matra BA Co., which is described by look-up tables that define the non-linear characteristic of the aerodynamics. It describes a full 5 degree of freedom model in parametric format with severe cross-coupling and non-linear behaviour. A polynomial model has been produced to match the parametric model as closely as possible in a least squares sense. This polynomial model is in the form of polynomial relationships that are then used for control synthesis. A set of convex models is produced that map the vertex points in a high order parameter space (of the order of 16 variables). The multiple model description of the airframe aerodynamics is tested for sensitivity on the aerodynamic coefficients. Also, in order to examine manoeuvrability the model is described in Cartesian and Polar coordinates.

In order for the missile system to follow a required trajectory, in other words to respond accurately and rapidly to a large range of acceleration demands, an appropriate control algorithm design (i.e. an autopilot system) is necessary. One way to achieve that is by linearizing the equation of motion about equilibrium conditions as Horton [69] has done. Another way would be to keep the nonlinear system as it is and apply global linearization via state feedback which is considered in Chapter 3.

Chapter 3

Feedback Linearization

3.1 Introduction

As stated earlier in the literature review of Chapter 1, the main idea of Feedback Linearization (FL) techniques is to algebraically transform a non-linear system dynamics into a linear form by using state feedback, with *Input/State Linearization* corresponding to complete linearization or *Input/Output Linearization* to partial linearization by Isidori et al [2], by Su [4], by Hunt and Sue [3]. This differs entirely from conventional linearization (i.e. Jacobian linearization) in that FL is achieved by exact state transformations and feedback, rather than by linear approximations of the dynamics. Feedback Linearization can be used for both stabilization and tracking control problems, single-input and multiple-input systems, and has been successfully applied to a number of practical nonlinear control problems.

Chapter 3 provides a description of Feedback Linearization, including the theory, its application for control design and its limitations. Then an approximate Input/Output Linearization method for controlling a the nonlinear missile system that is input-output linearizable is examined. The design retains the order and the relative degree of the system in the linearization process, hence producing a linearized system with no internal or zero dynamics.

Both SISO and MIMO systems have been considered in Section 3.3 and Section 3.4. In the SISO case two trajectory control designs are studied. The main difference from other research work is that instead of using angles or body rates as outputs for the linearization process, lateral velocities and body accelerations are used. Lateral velocity is directly related to the lateral acceleration, as in steady state a constant incidence angle is associated with a constant lateral acceleration. The chosen output for the second design has a linear relationship with the controlled one, hence better closed loop performance has been achieved when higher demands are required. Two different ways of presenting the nonlinear control design in Polar and in Cartesian coordinates have been considered in the MIMO design and their advantages and disadvantages have been analyzed. An additional controlled output for the roll channel has also been examined.

3.2 Feedback Linearization theory

3.2.1 Feedback Linearization process

Consider input-output linearization of a single-input nonlinear system described by the state space representation:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{3.1}$$

where y is the system output, with $f(x)$ and $g(x)$ being the smooth vector fields. According to Slotine and Li [14], a linear input-output relation is generated by differentiating the output function y repeatedly until the input u appears. This is shown here by following the notations of Differential Geometry addressed in Appendix D:

$$\dot{y} = \nabla h(f + gu) = L_f h(x) + L_g h(x)u \tag{3.2}$$

If $L_g h(x) \neq 0$ for some $x = x_0$ in Ω_x then, by continuity, that relation is valid in a finite neighbourhood Ω of x_0. In Ω, the input transformation

$$u = \frac{1}{L_g h}(-L_f h + \nu) \tag{3.3}$$

results in a linear relation between y and ν, namely $\dot{y} = \nu$. If $L_g h(x) = 0$ for all x in Ω_x, $\dot{y}$ is differentiated again to obtain:

$$\ddot{y} = {L_f}^2 h(x) + L_g L_f h(x)u \tag{3.4}$$

If $L_g L_f h(x) = 0$ for all x in Ω_x, $\ddot{y}$ is differentiated again until for some integer r, the following is true:

$$L_g L_f^{r-1} h(x) \neq 0 \tag{3.5}$$

for some $x = x_0$ in Ω_x, where the above relation is valid in a finite neighbourhood Ω of x_0. In Ω, the control law

$$u = \frac{1}{L_g L_f^{r-1} h}(-L_f^r h + \nu) = \frac{1}{\beta(x)}(-\alpha(x) + \nu) \tag{3.6}$$

is applied to

$$y^r = L_f^r h(x) + L_g L_f^{r-1} h(x)u \tag{3.7}$$

and the resulting relationship from reference signal ν to output is:

$$y^r = \nu \tag{3.8}$$

By using equation (3.6), which is a nonlinear state feedback (where $L_g L_f^{r-1} h(x)$ and $L_f^r h(x)$ are functions of x), a linear system is obtained from reference signal to output. This is not a linear approximation, it is often called an exact input-output linearization. Further on, the simple pole-placement controller can be extended to asymptotic tracking as studied by Hahn et al [7] and described in the next section.

3.2.2 Tracking Control

When a tracking control task is required, the reference signal (the new input of the linearization) is derived such as:

$$\nu = -k_0 y - k_1 \dot{y} - \ldots - k_{n-1} y^{n-1} \tag{3.9}$$

with the k_i chosen such that the polynomial $p^n + k_{n-1}p^{n-1} + \ldots + k_0$ has all its roots strictly in the left half complex plane (i.e. is Hurwitz), leading to the exponentially stable dynamics described by:

$$y^{(n)} + k_{n-1} y^{(n-1)} + \ldots + k_0 y = 0 \tag{3.10}$$

which implies that $y(t) \to 0$ as given by Slotine and Li [14]. The reference signal (the new input ν) has been designed such as:

$$\nu = y_d^{(n)} - k_0 e - k_1 \dot{e} - \ldots - k_{n-1} e^{n-1} \tag{3.11}$$

to satisfy the closed loop error dynamics within the outer loop, so the autopilot system is able to track desired output $y_d(t)$. This is shown in fig. 3.1 for a second order system. The reference signal is:

$$\nu = y_d^{(2)} - k_0 e - k_1 \dot{e} \tag{3.12}$$

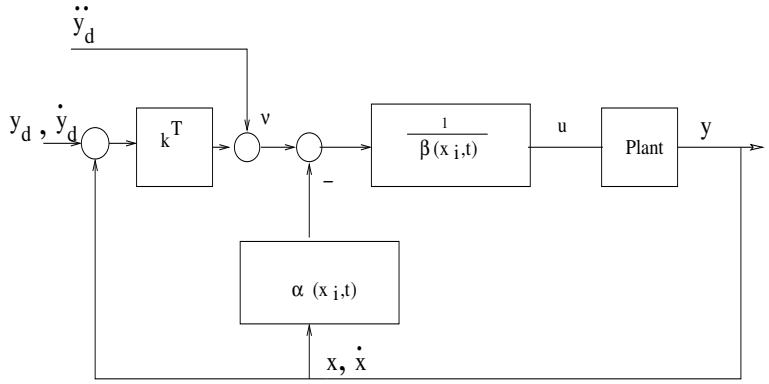


Figure 3.1: Tracking Control Diagram

where $e(t) = y(t) - y_d(t)$ is the tracking error. This leads to exponentially convergent tracking with error dynamics given by:

$$\ddot{e} + k_1 \dot{e} + k_0 e = 0 \tag{3.13}$$

Then, by using the control law

$$u = \frac{1}{L_g L_f^{r-1} h}(-L_f^r h + y_d^{(n)} - k_0 e - k_1 \dot{e} - \ldots - k_{n-1} e^{n-1}) \tag{3.14}$$

desired tracking performance is achieved. Feedback linearization of MIMO systems is obtained similarly to the SISO case by Slotine [14] and is described in Appendix D.2.

Input/State Linearization
The number of differentiations (r) required for the input u to appear is called the *relative degree* of a nonlinear system. If the relative degree associated with the *Input-Output Linearization* is the same as the order of the system, the non-linear system is fully linearized which is the case of *Input/State Linearization*.

For linear systems the relative degree is related to well-known properties. For a SISO linear system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{3.15}$$

or

$$y(s) = G(s)u(s) \tag{3.16}$$

the relative degree can be calculated as the difference in degree between denominator and numerator of the transfer function $G(s)$.

If the relative degree of a nonlinear system is equal to the order of the system, an exact feedback linearization is achieved. The standard approach in feedback linearization given by Slotine and Li [14] is to use h to define the required change of coordinates.

For our system we define a series of functions ϕ_i related to $h(x)$ by:

$$\begin{aligned} \phi_1(x) &= h(x) \\ &\vdots \quad\quad \vdots \\ \phi_i(x) &= L_f^{i-1} h(x) \end{aligned} \tag{3.17}$$

Setting $\mu = \phi(x)$, the new equations are:

$$\begin{aligned} \dot{\mu}_i &= \phi_{i+1}, \qquad i = 1, \ldots, n-1 \\ \dot{\mu}_n &= \alpha(x) + \beta(x)u \end{aligned} \tag{3.18}$$

By using the control law $u = \beta(x)^{-1}(\nu - \alpha(x))$, the relation $\dot{\mu}_n = \nu$ is linear, with ν as an input to the linearized system, hence an exact state linearization is achieved. The description in equation (3.18) is often regarded as a *canonical form* for nonlinear systems.

Input/Output Linearization
By means of input-output linearization, the dynamics of a nonlinear system is decomposed into an external (input-output) part and an internal ("unobservable") part. Since the external part consist of a linear relation between y and ν (or equivalently the controllability canonical form between y and u), it is easy to design the input ν so that the output y behaves as desired. The internal part is called *internal dynamics* because it cannot be seen from the external input-output relationship. Then, the question is whether the internal dynamics will also behave well, i.e. whether the internal states will remain bounded.

If the relative degree is smaller than the system order, then the non-linear system is only partly linearized which is the case in *Input-Output Linearization*. This requires to transform the system into new set of states *called Normal forms* and whether the controller can be applied depends on the stability of the internal dynamics (the modes which are unobservable by the linearization process). Since the control design must account for the whole dynamics and therefore can not tolerate the instability of internal dynamics, the internal behaviour has to be addressed carefully.

3.2.3 Normal forms

When the relative degree r is defined and $r < n$ where n is the order of the system, the nonlinear system can be transformed into new coordinates called *Normal form*. To determine the normal form we can follow the same process as in Input/State linearization see equation (3.18), but the difference will be that we have an unobservable part of the system. The change into new coordinates means that:

$$\mu_1 = y, \quad \mu_2 = \dot{y} \quad , \ldots , \quad \mu_r = y^{r-1} \tag{3.19}$$

and the system description becomes:

$$\begin{aligned}
\dot{\mu}_1 &= \mu_2 \\
\dot{\mu}_2 &= \mu_3 \\
&\vdots \\
\dot{\mu}_r &= \alpha(x) + \beta(x)u \\
\dot{\mu}_{r+1} &= \psi_1(\mu, u) \\
&\vdots \\
\dot{\mu}_n &= \psi_{n-r}(\mu, u) \\
y &= \mu_1
\end{aligned} \tag{3.20}$$

for some functions α, β and ψ. The linearizing feedback is:

$$u = \frac{\nu - \alpha(x)}{\beta(x)} \tag{3.21}$$

which gives the following closed loop system:

$$\begin{aligned}
\dot{\mu}_1 &= \mu_2 \\
\dot{\mu}_2 &= \mu_3 \\
&\vdots \\
\dot{\mu}_r &= \mu \\
\dot{\mu}_{r+1} &= \psi_1(\mu, u) \\
&\vdots \\
\dot{\mu}_n &= \psi_{n-r}(\mu, u) \\
y &= \mu_1
\end{aligned} \tag{3.22}$$

We can see that the whole system has not been linearized. The $\mu_{1,\ldots,r}$ part of the dynamics of system (3.22) are in the form of integrator chains of length r. There is possibly still some nonlinear dynamics affecting the state variables $\mu_{r+1}, \ldots, \mu_n$. This dynamics is not visible in the output and is called the internal dynamics of the system (3.22). This dynamics depends on the output states μ and it is unobservable (it cannot be seen from the external input-output relationship). When we design the controller, the external $\mu_i \ \ (i = 1, \ldots, r)$ part is used because there is a relation between y and u hence is easy to design an input so that the output behaves as desired. However when the controller is applied to both the external $\mu_i \ \ (i = 1, \ldots, r)$ and the internal $\mu_i \ \ (i = r+1, \ldots, n)$ part of the system, the performance of the closed loop system will degrade since we haven't taken into account the part $\mu_i \ \ (i = r+1, \ldots, n)$ when designing the controller. It is important to study the stability of the internal dynamics. If it is unstable the system will become unstable too. However if it is stable, the system will remain stable, but the controller can't guarantee closed loop performance, as some part of the system was ignored when designing the controller.

It is difficult to directly determine the stability of the internal dynamics because it is nonlinear and coupled to the external closed-loop dynamics. The study of the internal dynamics stability can be simplified by studying the zero dynamics instead. The *zero dynamics* is defined to be the internal dynamics of the system when the system output is kept at zero by the input. Two useful remarks can be made about the zero-dynamics of nonlinear systems. First the zero-dynamics is an intrinsic feature of a nonlinear system, which does not depend on the choice of control law or desired trajectories. Second, examining the stability of zero-dynamics is much easier than examining the stability of internal dynamics, because the zero-dynamics only involves the internal states while the internal dynamics is coupled to the external dynamics and desired trajectories.

The internal dynamics associated with the input-output linearization corresponds to the last $(n - r)$ equations $\dot{\mu} = w(\mu, \psi)$ of the *normal form*. Generally, this dynamics depends on the output states μ. An intrinsic property of the nonlinear system can be defined by considering the system's internal dynamics when the control input is

such that the output y is maintained at zero. Studying this so-called *zero dynamics*, some conclusions about the stability of the internal dynamics can be made.

3.2.4 Examples of Input/Output Linearization

In the following examples we have shown:
1. An input/output linearization which rendered the system with no zero dynamics;
2. An input/output linearization with stable zero dynamics;
3. An input/output linearization with unstable zero dynamics.

1. No Zero dynamics

Consider the non-linear system

$$\begin{aligned} \dot{x_1} &= x_1^2 x_2 \\ \dot{x_2} &= 3x_2 + u \end{aligned} \tag{3.23}$$

For the given non-linear system by choosing an output for the linearization process we can show that the system will result in no zero dynamics if the relative degree of the equivalent linear system is equal to the order of the system.

Define the output to be:

$$y = x_1 \tag{3.24}$$

By differentiating twice in order to achieve an input-output relationship we get:

$$\ddot{y} = 2x_1 x_2 \dot{x_1} + {x_1}^2 \dot{x_2} = 2{x_1}^3 {x_2}^2 + 3{x_1}^2 x_2 + u \tag{3.25}$$

The required static feedback for linearized closed loop input/output behaviour is given by:

$$u = \frac{1}{\beta}(\nu - \alpha) = \frac{1}{1}(\nu - 2{x_1}^3 {x_2}^2 - 3{x_1}^2 x_2) \tag{3.26}$$

which will cancel the nonlinearity. The original system is of second order and the relative degree is equal to 2, so there are no zero dynamics involved and the stability of the linearized system can be guaranteed.

2. Stable Zero dynamics

Let consider another system:

$$\begin{aligned} \dot{x_1} &= x_2^3 \\ \dot{x_2} &= u \\ y &= x_1 + x_2 \end{aligned} \tag{3.27}$$

Introducing new states, gives

$$z_1 = x_1 + x_2, \qquad z_2 = x_2 \tag{3.28}$$

This will get the system in the following form:

$$\begin{aligned} \dot{z_1} &= z_2^3 + u \\ \dot{z_2} &= u \\ y &= z_1 \end{aligned} \tag{3.29}$$

The feedback is

$$u = \nu - z_2^3 \tag{3.30}$$

where ν is the reference signal. This results in:

$$\begin{aligned} \dot{z_1} &= \nu \\ \dot{z_2} &= -z_2^3 + \nu \\ y &= z_1 \end{aligned} \tag{3.31}$$

We see that:

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

and the dynamics which is not visible in the output signal is:

$$\dot{z_2} = -z_2^3 + \nu \tag{3.33}$$

It is easy to see that this system is globally stable for any constant ν.

3. Unstable Zero dynamics

Consider instead the system:

$$\begin{aligned} \dot{x_1} &= -x_2^2 + u \\ \dot{x_2} &= u \\ y &= x_1 \end{aligned} \tag{3.34}$$

In this case is unnecessary to make a coordinate change since we already have $x_1 = y$ and the system has relative degree 1. The feedback is:

$$u = x_2^2 + \nu \tag{3.35}$$

giving the following form of the system:

$$\begin{aligned} \dot{x_1} &= \nu \\ \dot{x_2} &= x_2^2 + \nu \\ y &= x_1 \end{aligned} \tag{3.36}$$

There are problems already for $\nu = 0$. A small initial value of x_2 gives a solution that rapidly approaches infinity. This means that the control signal will also tend to infinity.

In order to produce linearized systems that have no internal dynamics, techniques which preserve the dynamic order of the system such as approximate feedback linearization are needed. A few ways of achieving this are given in the summary.

3.2.5 Summary

Feedback linearization can be used for both stability and tracking control problems, for both single-input (SISO) and multiple-input systems (MIMO), and has been successfully applied to a number of practical nonlinear control problems, both as a system analysis tool and as a controller design tool, just to name a few: Hahn et al [7], Suykens and Vandewalle [72], Henson and Seborg [73], Bezick et al [8]. With dynamic inversion, a nonlinear control law is designed which globally reduces the dynamics of the selected controlled variables to integrators. A closed loop system is then designed to make the controlled variables exhibit specified command response. However the theory has got some limitations.

Firstly, it can not be used for all nonlinear systems. The applicability of Input/State linearization is quantified by a set of stringent conditions, while Input/Output Feedback Linearization cannot be applied when the relative degree is not defined.

Secondly the full state has to be measured. The second problem is due to the difficulty of finding convergent observers for nonlinear systems and when an observer can be found, the lack of a general separation principle which guarantees that the straightforward combination of a stable state feedback controller and a stable observer will guarantee the stability of the closed-loop system.

Thirdly, one of the obstacles in the application of Input/Output Linearization is due to non-minimum phase systems which produce unstable zero dynamics. Because Input/Output Linearization relies on a nonlinear version of pole-zero cancellations, if the zero dynamics are unstable some of the unobservable states become unbounded. A not well defined relative degree leads to internal dynamics with unobservable states through the linearization. In other words, one of the main problems with applying Feedback linearization techniques is that the process produces a system with the same relative degree as the original system, but usually with an order

that is less. Indeed, the linearized system order is the same as the relative degree unless pre-compensators are used to artificially change the order and the relative degree. This process results in zero or internal dynamics, which are modes that are effectively rendered unobservable by the linearization process. If the system is non-minimum phase, then the zero dynamics are unstable. The analogy with linear systems is that a zero-pole system is linearized into an all-pole system by selecting the pole-zero excess as the order of the approximating system. In order to produce linearized systems that have no internal dynamics, techniques which preserve the dynamic order of the system are needed.

Several approaches are possible to the avoidance of internal or zero dynamics. One approach is to neglect terms in input derivatives until the required system order is reached as given by Hauser et al [74]. Another is to pre-compensate the system to increase the system relative degree artificially, and thus having some limited authority over the stability of the internal dynamics as detailed by Slotine and Li [14]. Designing systems with unstable zero dynamics can also be achieved provided the input to the system remains bounded under feedback by Lu et al [75]. A fourth way is to choose an output which has the required relative degree, and which is related to the required control output in some manner. The approach used in this thesis is a combination of the first two: to select an output that relates to the variable that is to be controlled, but which gives a greater relative degree, and to neglect small terms that allow the final relative degree to be achieved. This is applied to the nonlinear missile system and detailed in Section 3.3 and Section 3.4 for SISO and MIMO case respectively.

Finally no robustness is guaranteed in the presence of parametric uncertainty or unmodelled dynamics. This is due to the fact that the exact model of the nonlinear system is not available in performing feedback linearization. This disadvantage has been successfully compensated by using robust control technique, addressed later on in the thesis in Chapter 4.

3.3 Trajectory control design for SISO system

For the SISO missile system we have studied the single plane case. The aim is to track the missile lateral acceleration demand in both pitch and yaw planes which are treated as being uncoupled. The missile model in this section is thus described by the reduced 2DOF system assuming there is no interaction between lateral motion and roll.

Two different designs have been considered. In the first, **Design 1**, tracking and non-linear controllers are designed by defining lateral velocity as an output as it produces a higher relative degree than directly controlling lateral acceleration, which has a relative degree of zero. Lateral velocity is directly related to the lateral acceleration, as in steady state a constant incidence angle is associated with a constant lateral acceleration. In the second, **Design 2**, the augmented lateral acceleration has been used as an output for the linearization procedure, instead of the actual one. This is again to be able to achieve the same relative degree as the order of the system, to eliminate zero dynamics.

3.3.1 Design 1: Tracking lateral acceleration via lateral velocity

Both horizontal and vertical lateral motions are symmetric in format and the process of linearization to control lateral velocities is the same, hence Feedback Linearization for one of the channels is shown here. The control of the missile will be accomplished by controlling side-slip velocity. The horizontal motion has already been defined for yaw channel ($\dot{v}$ and $\dot{r}$) in equation (2.8) of Chapter 2. There is no roll interaction (no ξ term) and the equation for side-slip velocity is now:

$$\begin{aligned}
\dot{v} &= V^o(C_{yv}v + V_oC_{y\zeta}\zeta) - ur \\
&= V^o[(C_{yv_0} + C_{yv_M}M + C_{yv_\sigma} \mid \sigma \mid)v + \\
&\quad V_o(C_{y\zeta_0} + C_{y\zeta_M}M + C_{y\zeta_\sigma} \mid \sigma \mid)\zeta] - ur \\
&= V^o[(\bar{C}_{yv_0}v + \bar{C}_{yv_\sigma} \mid v \mid v + \\
&\quad V_o\bar{C}_{y\zeta_0}\zeta + V_o\bar{C}_{y\zeta_\sigma} \mid v \mid \zeta] - ur
\end{aligned} \tag{3.37}$$

where the Mach number M, and the total velocity V_o are slowly varying with:

$$\begin{aligned}
\mid \sigma \mid &= \frac{\mid v \mid}{V_o}\frac{180}{\pi} \\
M &= \frac{V_o}{SoS}
\end{aligned}$$

$$V^o = \frac{1}{2m}\rho V_o S \tag{3.38}$$

where $\bar{C}_{yv_0}, \bar{C}_{yv_\sigma}, \bar{C}_{y\zeta_0}, \bar{C}_{y\zeta_\sigma}$ are defined in equations (2.37) and (2.38) of Chapter 2.

The state space for the horizontal motion can be written in the following parametric format:

$$\begin{aligned} \dot{x}_1 &= a_1 x_1 + a_2 x_1^2 + a_3 x_2 + (a_4 x_1 + a_5) u_1 \\ \dot{x}_2 &= b_1 x_1^3 + b_2 x_1^2 + b_3 x_1 + b_4 x_1 x_2 + b_5 x_2 + (b_6 x_1 + b_7) u_1 \end{aligned} \tag{3.39}$$

where:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} \dot{v} \\ \dot{r} \end{bmatrix} \tag{3.40}$$

and the parameters $a_1, \ldots, a_5$ and $b_1, \ldots, b_7$ are defined in Appendix C.

The state space of the nonlinear system is:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h = x_1 \end{aligned} \tag{3.41}$$

or in matrix form:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} a_1 x_1 + a_2 x_1^2 + a_3 x_2 \\ b_1 x_1^3 + b_2 x_1^2 + b_3 x_1 + b_4 x_1 x_2 + b_5 x_2 \end{bmatrix} + \begin{bmatrix} a_4 x_1 + a_5 \\ b_6 x_1 + b_7 \end{bmatrix} u_1 \tag{3.42}$$

The equation (3.41) is in standard form and Input/Output Linearization technique can be applied to it. By defining the side-slip velocity x_1 as an output ($y_1 = x_1$) and by applying Feedback Linearization, only one differentiation of the output $\dot{y}_1 = \dot{x}_1$ is enough in order for the input u to appear (as $\dot{x}_1 = f(x_1, x_2) + g(x_1)u$) which can establish an Input/Output relation. In that case the relative degree (i.e. the order of the equivalent linear system) would be $r_1 = 1$, which is less than the order of the non-linear system (2^{nd}). This results in an equivalent linear system with first order internal dynamics. However, an approximate Feedback Linearization known as g-modification by Hauser et al [74] can be used instead of exact Feedback Linearization. In which case the relative degree is increased by the required order to equal the order of the non-linear system which will result in system with no internal dynamics and a tracking controller is designed without having to consider stability of unobservable modes. Using this approximation, terms are discarded in order to retain an approximate system with an equivalent order and relative degree.

By using normal coordinate transformation let $\mu_1 = \phi_1 = h(x) = x_1$. We differentiate μ_1:

$$\dot{\mu}_1 = \underbrace{a_1 x_1 + a_2 x_1^2 + a_3 x_2}_{\mu_2 = \phi_2(x)} + \underbrace{(a_4 x_1 + a_5) u_1}_{\psi_1(x_1, u_1)} \tag{3.43}$$

We then neglect $\psi_1(x_1, u_1)$ as it is a very small term compared to the rest of the missile dynamics. The relative magnitudes between ψ_1 dynamics and ϕ_2 dynamics is a ratio of $1 : 10$, which has been validated via simulation. This suggests that ψ_1 can be neglected. There is also a phisical explanation for the justification for neglecting the ψ_1 term. ψ_1 is the fin force and is phisically designed to be smaller than the ϕ_2 dynamics. ϕ_2 is the body aerodynamic force acting at the centre of pressure and is normaly an order of magnitude greater than the fin force. The main effect of ψ_1 is to produce a small force at a large distance which produces a large turning moment. The turning moment term then appears as β_1 in the $\dot{r}$ equation due to the differentiation of the ϕ_2 dynamics. Hence, although the fin force term is neglected, the fin turning moment is retained.

Hence $\dot{\mu}_1 = \mu_2$. We differentiate μ_2 to get:

$$\begin{aligned} \dot{\mu}_2 &= \underbrace{(2{a_2}^2 + a_3 b_1){x_1}^3 + (3a_1 a_2 + a_3 b_2){x_1}^2 + (a_1^2 + a_3 b_3)x_1}_{\alpha_1} \\ &\quad + \underbrace{(a_1 a_3 + a_3 b_5)x_2 + (2a_2 a_3 + a_3 b_4)x_1 x_2}_{\alpha_1} \\ &\quad + \underbrace{(2a_3 b_6 x_1 + a_3 b_7)}_{\beta_1} u_1 \end{aligned} \tag{3.44}$$

By neglecting the term ψ_1 shown in (3.43), the g vector field has been modified. The effect of neglecting the term ψ_1 in equation (3.43) is to eliminate a non-linear zero in the system within the model description, and which is not taken into account in the non-linear control design. It has been shown by White [71], this will not affect the performance of the control design in a significant manner as the zero can be approximated by:

$$z \approx -\frac{(a_4 \mid x_1 \mid + a_5)}{(2a_3 b_6 \mid x_1 \mid + a_3 b_7)} \tag{3.45}$$

The zero is negative for all values of x_1, hence will not affect the stability of the closed loop dynamics.

The linearized system can be written in compact form:

$$\begin{aligned} \mu_1 &= h \\ \dot{\mu}_1 &= \mu_2 \\ \dot{\mu}_2 &= \alpha_1 + \beta_1 u_1 \end{aligned} \tag{3.46}$$

The output (h) has been differentiated twice, hence possesses a relative degree (r) of 2. Since the relative degree is equal with the order of the system, fully linearization of the non-linear system has been achieved with no zero dynamics. The equation (3.46) represents a direct relationship between the output (h) and the input (u_1) [76]

by Wang. The required static state feedback for decoupled closed loop Input-Output behaviour is given by Kravaris and Soroush [77] as:

$$u_1 = \frac{1}{\beta_1}(\nu - \alpha_1) \tag{3.47}$$

The **linearized closed loop system** is now given by:

$$\ddot{y} = v \tag{3.48}$$

where ν is the new linearized system input, as given by Wang [76]. For the second order yaw plane system, the tracking controller in the outer loop can be investigated by choosing the resulted new control v input to be:

$$\nu = \ddot{y}_d - k_1\dot{e} - k_2 e \tag{3.49}$$

where $e = y - y_d$. The closed-loop system is thus characterised with the following error dynamics:

$$\ddot{e} + k_1\dot{e} + k_2 e = 0 \tag{3.50}$$

where k_1 and k_2 are chosen such that all roots of $s^2 + k_1 s + k_2 = 0$ are in the open left–half plane Hurwitz, which ensures $\lim_{t\to\infty} e(t) = 0$, as shown by Wang [76].

The tracking control problem of the non-linear system described by equation (3.39) has been solved using the control law in equation (3.47) with the new input defined in (3.49). Indeed, since equation (3.50) has the same order as the non-linear system, there is no part of the system dynamics which is rendered "unobservable" in the approximate Input/Output Linearization. Since there are no zero dynamics in the linearized system, the tracking problem has been solved provided the approximation is valid and the neglected terms are small as proved by Hauser et al [74].

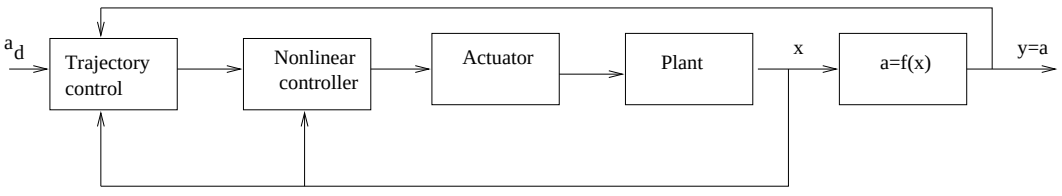


Figure 3.2: Trajectory control design for **Design 1** in SISO

The trajectory control design has been shown in fig. 3.2. A fast linear actuator with natural frequency of 250 rad/sec has been included in the non-linear system. The block of the Plant is represented by equation (3.39). The nonlinear controller is

derived by equation (3.47). The desired acceleration a_d is achieved by using the non-linear equation $a_d = f(v_d)$. The relation between lateral acceleration and side-slip velocity is:

$$\alpha = \dot{v} + Ur = a_1 v + a_2 v^2 - Ur + Ur \tag{3.51}$$

hence by finding the roots of

$$\alpha_d = a_1 v_d + a_2 v_d^2 \tag{3.52}$$

we would know what side-slip velocity is required for particular lateral acceleration demand, hence desired tracking performance can be achieved.

The error dynamics ($e = v - v_d$ and $\dot{e} = \dot{v}_d - \dot{v}$) are constructed using the α_d signal and the feedback of the actual states - side-slip velocity v, yaw rate r and acceleration a, also shown in fig. 3.3.

$$\dot{v}_d - \dot{v} = \alpha_d - Ur_d - \alpha + Ur \tag{3.53}$$

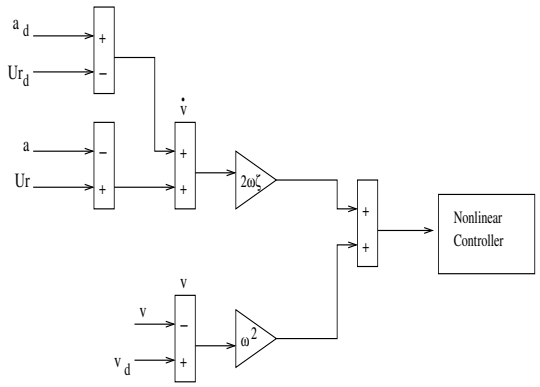


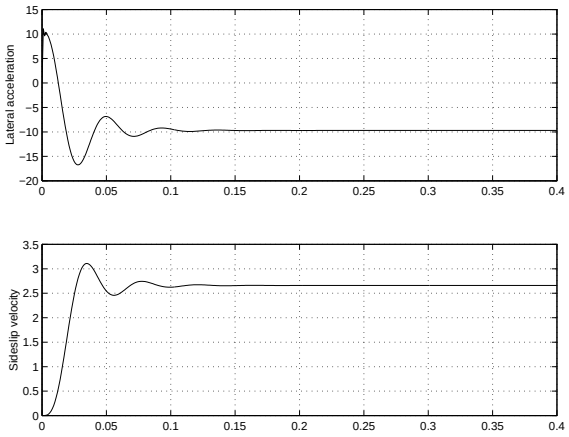
Figure 3.3: Error dynamics

The trajectory control is derived concerning the closed loop error dynamics expressed in equation (3.50). The error coefficients are chosen to satisfy Hurwitz polynomial for the second order error equation in each channel, hence $k_1 = 2\zeta w_n$ and $k_2 = w_n^2$, with $w_n = 70\ rad/sec$ and $\zeta = 0.7$. This speed of response is significantly faster than the open loop response and so should exercise the dynamics of the non-linear missile.

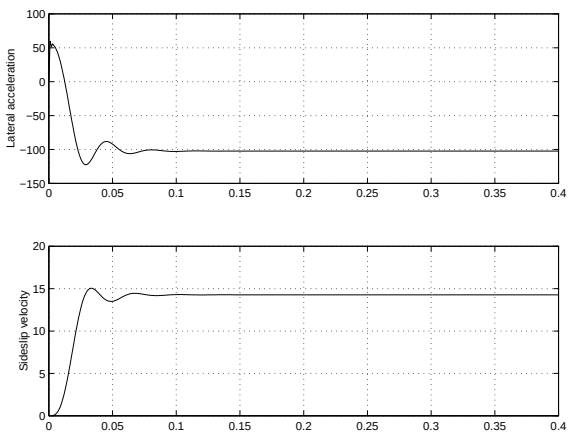
Results and few comments

The results for 1g (10 m/sec^2) and 10g (100 m/sec^2) lateral acceleration demands are shown in fig. 3.4. They show side-slip velocity and the resulting lateral acceleration responses. The side-slip velocity demand has been scaled using equation (3.52) to give required acceleration. Actuator fin angle and fin rate are also shown to make sure that no unrealistic control signals are generated. The figures show almost

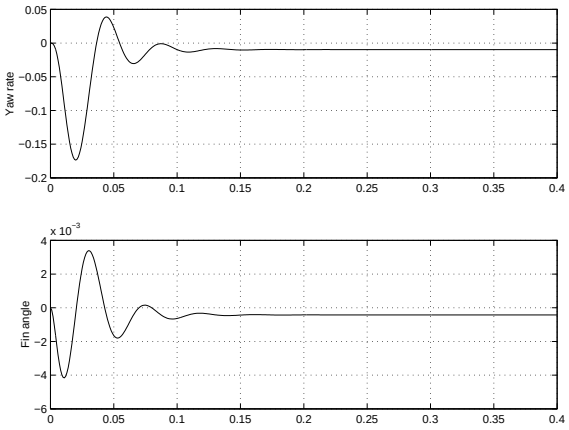
identical step responses with some variation in peaks and steady state values for the body yaw rate, the actuator movement (fin angle) and the side-slip velocity. As expected for a non-linear system, the relationship between lateral velocity and lateral acceleration is non-linear. The results also show that the actuator does not significantly affect the design. The non-linear approach is also shown to be reasonably accurate, as the predicted and actual performance are very close, as given by Tsourdos et al [78]. The non-minimum phase effect on the lateral acceleration responses can be seen as evidence that an inherent right half plane zero exists within the nonlinear system.



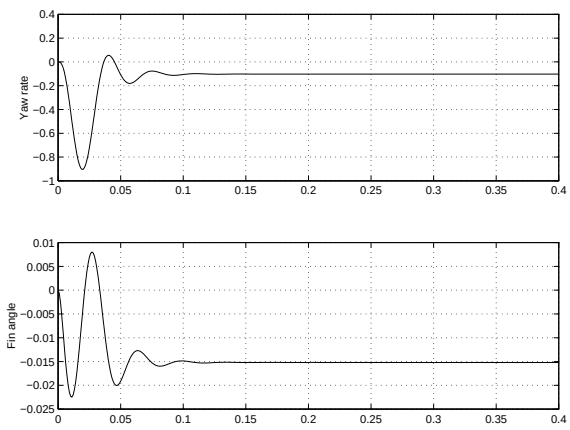
Acceleration and velocity for 1g demand



Acceleration and velocity for 10g demand



Rate and fin angle for 1g demand



Rate and fin angle for 10g demand

Figure 3.4: Results for **Design 1** in SISO for $a_d = 10$ m/sec^2 and $a_d = 100$ m/sec^2

3.3.2 Design 2: Tracking lateral acceleration via augmented acceleration

By defining lateral acceleration as an output and applying the standard Input/Output Linearization procedure, the relative degree (i.e. the order of the equivalent linear system) is $r = 0$, which is again less than the order of the non-linear system (2^{nd}). This results now in an equivalent linear system with second order internal dynamics. However, by using the augmented acceleration as the output for the non-linear controller design and by applying approximate Feedback Linearization known as g-modification, as given by Hauser et al [74], the relative degree is increased by the required order to equal the order of the non-linear system. As the system has no internal dynamics we can then design a tracking controller without having to consider stability of the unobservable modes. The dynamic equation for lateral acceleration has been derived in equation (2.32) of Chapter 2 and is given here again by:

$$\begin{aligned} \alpha &= \dot{v} + Ur \\ &= \phi(v) + \psi(v, \zeta) \end{aligned} \tag{3.54}$$

From equation (3.54), the output contains the input control fin deflection ζ by virtue of the term $\psi(v, \zeta)$. This makes the lateral acceleration have a relative degree of zero. This term, however, can be shown to be the lateral force developed by the fin. The fin's main contribution to the dynamics of the missile is to develop a turning moment, by virtue of the term $\frac{1}{2} I_{yz}^{-1} \rho V_o^2 S d C_{n\zeta} \zeta$ in equation (2.8) for r and the equivalent term in equation (2.7) for q, detailed in Chapter 2. If this term is included in the output equation, then the augmented acceleration can be represented as:

$$\begin{aligned} \bar{\alpha} &= \alpha - \psi(v, \zeta) \\ &= \phi(v) = V^o[(\bar{C}_{yv_0} v + \bar{C}_{yv_\sigma} \mid v \mid v] \end{aligned} \tag{3.55}$$

or in parametric form as: $\bar{\alpha} = a_1 x_1 + a_2 x_1^2$. The augmented acceleration $\bar{\alpha}$ is used for lateral control instead of the lateral acceleration α. The difference between the two outputs α and $\bar{\alpha}$ is now just the lateral acceleration developed by the control fin, and as such will not introduce much error in the control of the lateral acceleration, as given by White [71].

The matrix form of the non-linear system shown in equation (3.42) is the same here, but the controlled output of the non-linear system (3.41) now is:

$$y = h = a_1 x_1 + a_2 x_1^2 \tag{3.56}$$

In order to apply Input/Output Linearization and to retain the system order with no zero dynamics, an approximate Input/Output Linearization technique is applied. Let $\mu_1 = h(x) = a_1 x_1 + a_2 x_1^2$. Then by differentiating μ we get:

$$\begin{aligned}\dot{\mu}_1 &= \underbrace{(a_1^2 x_1 + 3a_1 a_2 x_1^2 + a_1 a_3 x_2 + 2a_2^2 x_1^3 + 2a_2 a_3 x_1 x_2}_{\mu_2 = \phi_1(x)} \\ &\quad + \underbrace{(a_2 a_4 x_1^3 + (a_1 a_4 + a_2 a_5) x_1^2 + a_1 a_5 x_1) u_1}_{\psi_1(x_1, u_1)} \end{aligned} \tag{3.57}$$

The $\psi_1(x_1, u_1)$ dynamics, in this case, is close to zero 0.039 confirmed via simulation, hence we can neglect it. Conversely to **Design 1**, there is no physical interpretation for neglecting the term. To take the next step we set ψ_1 to zero and $\dot{\mu}_1 = \mu_2$. Then we differentiate μ_2 to get the Input-Output relation:

$$\begin{aligned}\dot{\mu}_2 &= \underbrace{(6a_2^3 + 2a_2 a_3 b_1) x_1^4 + (12 a_1 a_2^2 + a_1 a_3 b_1 + 2a_2 a_3 b_2) x_1^3 + (a_1^3 + a_1 a_3 b_3) x_1}_{\alpha_1} \\ &\quad + \underbrace{(a_1^2 + 6a_1^2 a_2 + a_1 a_3 b_2 + 2a_2 a_3 b_3) x_1^2 + (8a_2^2 a_3 + 2a_2 a_3 b_4) x_1^2 x_2}_{\alpha_1} \\ &\quad + \underbrace{(8a_1 a_2 a_3 + a_1 a_3 b_4 + 2a_2 a_3 b_5) x_1 x_2 + (2a_2 a_3^2) x_2^2 + (a_1^2 a_3 + a_1 a_3 b_5) x_2}_{\alpha_1} \\ &\quad + \underbrace{(6a_1^2 a_4 x_1^3 + 2a_2 (3a_1 a_4 + 3a_2 a_5 + a_3 b_6) x_1^2) + (a_1 (a_1 a_5 + a_3 b_7))}_{\beta_1} u_1 \\ &\quad + \underbrace{((6a_1 a_2 a_5 + a_1^2 a_4 + a_1 a_3 b_6 + 2a_2 a_3 b_7) x_1) + (2a_2 a_3 (a_4 x_1 + a_5) x_2)}_{\beta_1} u_1 \end{aligned} \tag{3.58}$$

and the resulted system is

$$\begin{aligned}\dot{\mu}_1 &= \mu_2 \\ \dot{\mu}_2 &= \alpha_1 + \beta_1 u_1 \end{aligned} \tag{3.59}$$

The output (y) possesses a relative degree (r) of 2, since (y) has been differentiated twice for the input (u_1) to appear. The relative degree of the system is now 2, and has the same order as the original system. Therefore there are no internal dynamics. Since the total relative degree is equal to the order of the system, fully linearization of the non-linear system is achieved. The effect of neglecting the term $\psi_1(x, u_1)$ in equation (3.57) is to eliminate a non-linear zero in the system within the model description, and which is not taken into account in the non-linear design. It has been shown by White [71], this will not affect the performance of the control design in a significant manner as the zero can be approximated by:

$$z \approx -\frac{\psi_1(x)}{\beta_1(x)} \tag{3.60}$$

When the augmented acceleration is defined as a control output of the linearization procedure we have applied an approximate Input/Output Linearization in order to

retain the order of the system. In that case there is no zero dynamics involved in the design. If we don't neglect any term, then the linearization will take place by solving the 2^{nd} derivative of the output for the first derivative of the input $\dot{\psi}$. A pre-compensator will cancel the inherent zero in the Input-Output equation. An approximation to this controller that does not include the cancellation pole can be used by neglecting the $\dot{\psi}_1$ term. The zero will exist, which is not taken into the analysis and will be stable if the fin angle moment is significantly greater than the static margin. This is usually the case in most agile missiles as the static margin is made as close to zero as possible for most missiles. This will produce a stable solution and tracking performance will be satisfactory.

The equation (3.59) represents a direct relationship between the output (h) and the input (u). The required static state feedback is given by the control law previously explained in **Design 1**, see equations (3.47), hence a decoupled closed loop Input-Output behaviour is achieved. For the linearized closed loop system (3.48), the new control input has been chosen to be (3.49), so desired tracking performance has been achieved. By selecting the gains such that all roots of the closed loop error dynamics (3.50) lie in the left-half plane, asymptotic global stability is guaranteed. The closed loop error dynamics is (2^{nd}) order, hence there is no part of the system dynamics which is rendered "unobservable" in the approximate Input/Output Linearization. Since there are no zero dynamics in the linearized system, the tracking problem has been solved, as discussed by Isidori [79], Slotine and Li [14].

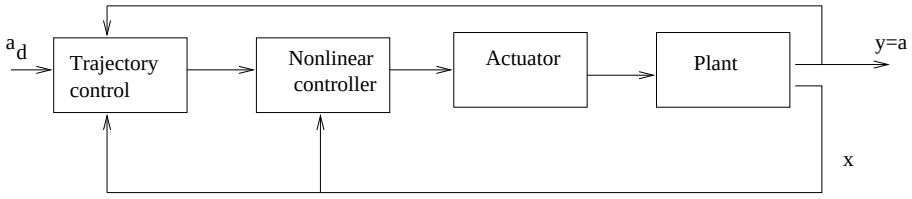
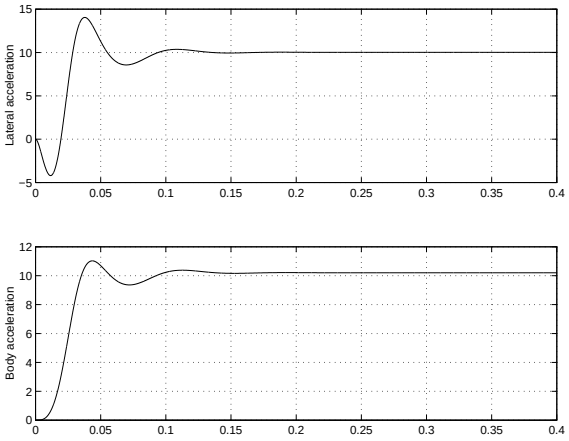


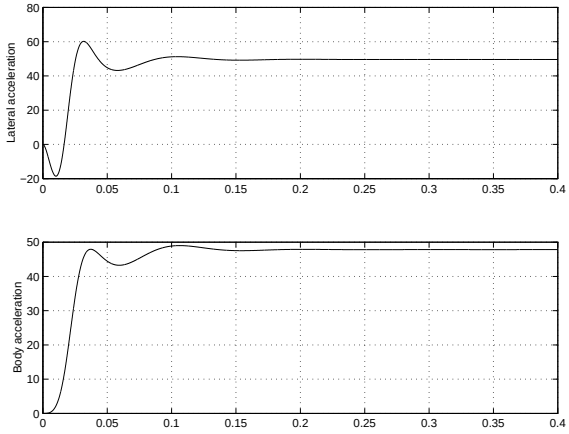
Figure 3.5: Trajectory control design for **Design 2** in SISO

The autopilot simulation is shown in fig. 3.5. The difference from **Design 1** is that the augmented acceleration is used as the linearization output. The controller design has been produced by using the augmented acceleration, but in the simulation the lateral acceleration has been used to check the validity of the made approximations. The error dynamics are constructed by using the desired lateral acceleration a_d signal and the feedback of the actual states - velocities, rates, accelerations and jerk. Also, a fast linear actuator with natural frequency of 250 rad/sec has been included in the non-linear simulation. Fixed gain trajectory controller has been used for the second order error equation (3.50) such as $k_1 = 2\zeta w_n$ and $k_2 = w_n^2$, with natural frequency $w_n = 60$ rad/sec and damping factor $\zeta = 0.65$. This speed of response is faster than the open loop response and so should exercise the dynamics

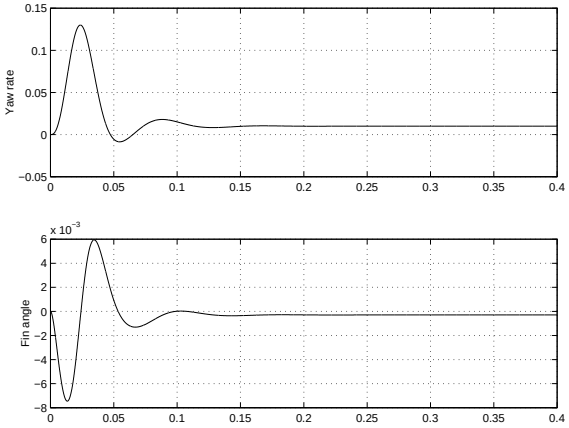
of the non-linear missile sufficiently for meaningful conclusions to be drawn. The results for 1g (10 m/sec^2) and 5g (50 m/sec^2) lateral acceleration demands are shown in fig. 3.6. These figures show almost identical step responses for both demands with some variation in peaks and steady state values for the body rate, the actuator movement and the lateral velocity. The difference between the lateral and the augmented acceleration shows that there is a good match between the two and that the steady state values are very close, as given by White et al [80]. This illustrates the small effect that the fin force has on the missile acceleration and justifies the use of the augmented body acceleration.



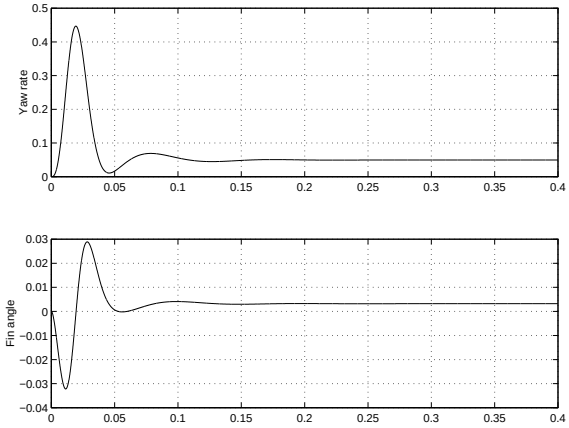
Lateral accelerations for 1g demand



Lateral accelerations for 5g demand



yaw rate and fin angle for 1g demand



yaw rate and fin angle for 5g demand

Figure 3.6: Results for **Design 2** in SISO for $a_d = 10\ m/sec^2$ and $a_d = 50\ m/sec^2$

3.4 Trajectory control design for MIMO system

The Input-Output design technique can be extended to the MIMO case. The aim here is again to track the missile lateral acceleration demand in pitch and yaw plane but to also maintain constant roll rate. The missile model in this section is described by full 6DOF system with roll interaction.

Two kinds of manoeuvrability are considered here. The first one is based on skid-to-turn (STT) motion which is presented in Cartesian coordinates and is valid for missiles with two pairs of control fins (rudders and elevators); and the second one is based on bank-to-turn (BTT) motion which requires Polar control and is valid for only one pair of control surfaces, very often used by long range, cruise missiles.

3.4.1 Design 1: Cartesian coordinates

The equations of motion, described in (2.42) in Chapter 2, are used to derive the state-space form of the non-linear system in a compact parametric format, as:

$$\begin{aligned}
\dot{x}_1 &= a_1x_1 + a_2x_1\sqrt{{x_1}^2+{x_3}^2} + a_3x_2 + (a_4\sqrt{{x_1}^2+{x_3}^2}+a_5)u_1 \\
\dot{x}_2 &= b_1x_1({x_1}^2+{x_2}^2) + b_2x_1\sqrt{{x_1}^2+{x_3}^2} + b_3x_1 + b_4x_2\sqrt{{x_1}^2+{x_3}^2} + b_5x_2 \\
&\quad +(b_6\sqrt{{x_1}^2+{x_3}^2}+b_7)u_1 - (b_9 + b_8\sqrt{{x_1}^2+{x_3}^2} + b_{10}({x_1}^2+{x_3}^2)u_3 \\
\dot{x}_3 &= a_1x_3 + a_2x_3\sqrt{{x_1}^2+{x_3}^2} - a_3x_4 + (a_4\sqrt{{x_1}^2+{x_3}^2}+a_5)u_2 \\
\dot{x}_4 &= -b_1x_3({x_1}^2+{x_3}^2) - b_2x_3\sqrt{{x_1}^2+{x_3}^2} - b_3x_3 + b_4x_4\sqrt{{x_1}^2+{x_3}^2} + b_5x_4 \\
&\quad -(b_6\sqrt{{x_1}^2+{x_3}^3}+b_7)u_2 + (b_9 + b_8\sqrt{{x_1}^2+{x_3}^2} + b_{10}({x_1}^2+{x_3}^2)u_3 \\
\dot{x}_5 &= c_1x_5 + (c_3 + c_4\sqrt{x_1^2+x_3^2})u_1 + (c_3 + c_4\sqrt{x_1^2+x_3^2})u_2 + c_2u_3
\end{aligned} \tag{3.61}$$

In a matrix form that would be:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \end{bmatrix} = \begin{bmatrix} a_1x_1 + a_2x_1\sqrt{{x_1}^2+{x_3}^2} + a_3x_2 \\ b_1x_1({x_1}^2+{x_2}^2) + b_2x_1\sqrt{{x_1}^2+{x_3}^2} + b_3x_1 + b_4x_2\sqrt{{x_1}^2+{x_3}^2} + b_5x_2 \\ a_1x_3 + a_2x_3\sqrt{{x_1}^2+{x_3}^2} - a_3x_4 \\ -b_1x_3({x_1}^2+{x_3}^2) - b_2x_3\sqrt{{x_1}^2+{x_3}^2} - b_3x_3 + b_4x_4\sqrt{{x_1}^2+{x_3}^2} + b_5x_4 \\ c_1 \end{bmatrix}$$

$$+ \begin{bmatrix} a_4\sqrt{{x_1}^2+{x_3}^2}+a_5 & 0 & 0 \\ b_6\sqrt{{x_1}^2+{x_3}^2}+b_7 & 0 & -b_9 - b_8\sqrt{{x_1}^2+{x_3}^2} \\ & & -b_{10}({x_1}^2+{x_3}^2) \\ 0 & a_4\sqrt{{x_1}^2+{x_3}^2}+a_5 & 0 \\ 0 & -b_6\sqrt{{x_1}^2+{x_3}^2}-b_7 & +b_9 + b_8\sqrt{{x_1}^2+{x_3}^2} \\ & & +b_{10}({x_1}^2+{x_3}^2 \\ (c_3 + c_4\sqrt{x_1^2+x_3^2} & (c_3 + c_4\sqrt{x_1^2+x_3^2} & c_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{3.62}$$

where:

$$x = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}^T = \begin{bmatrix} v & r & w & q & p \end{bmatrix}^T$$

$$u = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T = \begin{bmatrix} \zeta & \eta & \xi \end{bmatrix}^T$$

and the parameters $a_1, \ldots, a_5$, $b_1, \ldots, b_{10}$ and $c_1, \ldots, c_4$ are defined in Appendix C.

Equations (3.61) represent severe cross-coupling with inherent nonlinear terms within the missile dynamics. Firstly, Input/Output Linearization is used to decouple the system, and secondly, a trajectory controller is designed within the outer loop for tracking performance.

The non-linear system written in a standard form is:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h \\ &= \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \end{bmatrix} \end{aligned} \tag{3.63}$$

and the Input/Output Linearization technique can be applied to it. Like the SISO case study, in order to retain the system order with no zero dynamics, an approximate Input/Output Linearization is applied to the missile model. It is based on the second approximation method involving the modification of the function g, as presented by Hauser et al [74]. Using this approximation technique, terms are discarded in order to retain an approximate system with an equivalent order and relative degree. In other words the g vector field is modified.

Yaw plane

Let $\mu_1 = \phi_1 = h_1(x) = x_1$. Then:

$$\begin{aligned} \dot{\mu}_1 &= \underbrace{a_1 x_1 + a_2 x_1 \sqrt{{x_1}^2 + {x_3}^2} + a_3 x_2}_{\mu_2 = \phi_2(x)} + \underbrace{(a_4 \sqrt{{x_1}^2 + {x_3}^2} + a_5) u_1}_{\psi_1(x_1, u_1)} \\ \dot{\mu}_2 &= \underbrace{a_1^2 x_1 + 2a_1 a_2 x_1 \sqrt{{x_1}^2 + {x_3}^2} + a_3 x_2) + (a_2 \sqrt{{x_1}^2 + {x_3}^2})(a_2 x_1 \sqrt{{x_1}^2 + {x_3}^2} + a_3 x_2)}_{\alpha_1} \\ &\quad + \underbrace{\frac{a_1 a_2 x_1 x_3^2}{\sqrt{{x_1}^2 + {x_3}^2}} + a_2^2 x_1 x_3^2 - \frac{a_2 a_3 x_1 x_3 x_4}{\sqrt{{x_1}^2 + {x_3}^2}} + a_3 b_1 x_1^3}_{\alpha_1} \\ &\quad + \underbrace{a_3 (b_2 {x_1}^2 + b_3 x_1 + b_4 x_2 \sqrt{{x_1}^2 + {x_3}^2} + b_5 x_2}_{\alpha_1} \\ &\quad - \underbrace{(a_3 b_9 + b_8 \sqrt{{x_1}^2 + {x_3}^2} + a_3 b_{10} ({x_1}^2 + {x_3}^2))}_{\beta_2} u_3 + \underbrace{(a_3 b_6 \sqrt{{x_1}^2 + {x_3}^2} + a_3 b_7)}_{\beta_1} u_1 \end{aligned} \tag{3.64}$$

or with $\psi_1(x_1, u_1)$ set to zero

$$\begin{aligned} \dot{\mu}_1 &= \mu_2 \\ \dot{\mu}_2 &= \alpha_1 + \beta_1 u_1 + \beta_2 u_3 = v_1(x, u) \end{aligned} \tag{3.65}$$

Equation (3.65) is achieved by neglecting the term $\psi_1(x_1, u_1)$ shown in (3.64).

Pitch plane

Let $\mu_3 = \phi_3 = h_2(x) = x_3$. Then:

$$\begin{aligned} \dot{\mu}_3 &= \underbrace{a_1 x_3 + a_2 x_3 \sqrt{{x_1}^2 + {x_3}^2} - a_3 x_2}_{\mu_4 = \phi_4(x)} + \underbrace{(a_4 \sqrt{{x_1}^2 + {x_3}^2} + a_5) u_1}_{\psi_2(x_2, u_2)} \\ \dot{\mu}_4 &= \underbrace{a_1^2 x_3 + 2 a_1 a_2 x_3 \sqrt{{x_1}^2 + {x_3}^2} - a_3 x_4) + a_2 \sqrt{{x_1}^2 + {x_3}^2})(a_2 x_3 \sqrt{{x_1}^2 + {x_3}^2} - a_3 x_4)}_{\alpha_2} \\ &\quad + \underbrace{\frac{a_1 a_2 x_1^2 x_3}{\sqrt{{x_1}^2 + {x_3}^2}} + a_2^2 x_1^2 x_3 - \frac{a_2 a_3 x_1 x_2 x_3}{\sqrt{{x_1}^2 + {x_3}^2}} + a_3 b_1 x_3^3}_{\alpha_2} \\ &\quad + \underbrace{a_3 (b_2 {x_3}^2 + b3 x_3 - b_4 x_4 \sqrt{{x_1}^2 + {x_3}^2} - b_5 x_4}_{\alpha_2} \\ &\quad - \underbrace{(a_3 b_9 + b_8 \sqrt{{x_1}^2 + {x_3}^2} + a_3 b_{10} ({x_1}^2 + {x_3}^2))}_{\beta_4} u_3 + \underbrace{(a_3 b_6 \sqrt{{x_1}^2 + {x_3}^2} + a_3 b_7)}_{\beta_3} u_2 \end{aligned} \tag{3.66}$$

or with $\psi_2(x_2, u_2)$ set to zero

$$\begin{aligned} \dot{\mu}_3 &= \mu_4 \\ \dot{\mu}_4 &= \alpha_2 + \beta_3 u_2 + \beta_4 u_3 = v_2(x, u) \end{aligned} \tag{3.67}$$

For the roll plane, the roll angle (λ) has been taken as an output for the linearization process instead of the roll rate. Both roll rate and roll angle control are used in practice. This study will concentrate on roll angle control as this is the most useful in practice, when asymmetric sensors are fitted and BTT control is used.

Roll plane

Let $\mu_5 = \phi_5 = h_3(x) = x_6$, where $x_6 = \lambda$ the roll angle. Then:

$$\begin{aligned} \dot{\mu}_5 &= \underbrace{x_5}_{\phi_6 = \mu_6} \\ \dot{\mu}_6 &= \underbrace{c_1 x_5}_{\alpha_3} + \underbrace{c_2}_{\beta_7} u_3 + \underbrace{(c_3 + c_4 \sqrt{x_1^2 + x_3^2})}_{\beta_6} u_2 + \underbrace{(c_3 + c_4 \sqrt{x_1^2 + x_3^2})}_{\beta_5} u_1 \end{aligned} \tag{3.68}$$

or

$$\begin{aligned} \dot{\mu}_5 &= \mu_6 \\ \dot{\mu}_6 &= \alpha_3 + \beta_5 u_1 + \beta_6 u_2 + \beta_7 u_3 = v_3(x, u) \end{aligned} \tag{3.69}$$

The output $y_1 = h_1(x)$ possesses relative degree r_1 of 2, the output $y_2 = h_2(x)$ also possesses relative degree r_2 of 2, and the output $y_3 = h_3(x)$ possesses relative degree r_3 of 2. Hence the total relative degree of the system is equal with the summation of the r_1, r_2, r_3 and is now 6, which means the system has the same order as the original one, therefore there are **no internal dynamics**. And since the total relative degree is equal with the order of the system, fully linearization of the non-linear system has been achieved.

The effect of neglecting the terms ψ_i from equations (3.64),(3.66) is to eliminate a non-linear zero in the system within the model description, and which is not taken into account in the non-linear control design. Provided the side-slip force is not too great, as explained by White [71], this will not affect the performance in a significant manner. The zero can be approximated by:

$$z \approx -\frac{(a_4\sqrt{x_1^2 + x_3^2} + a_5)}{(a_3 b_6\sqrt{x_1^2 + x_3^2} + a_3 b_7)} \tag{3.70}$$

The explanations for the neglected terms (ψ_i), described earlier for the SISO system in **Design 1**, are valid here.

Equations (3.65), (3.67) and (3.69) represent a direct relationship between the outputs (h_i) and the inputs (u_i). The required static state feedback for decoupled closed loop Input-Output behaviour of a MIMO system is given by the control law as:

$$u = E^{-1}\left\{v - \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}\right\} \tag{3.71}$$

where E^{-1} is the characteristic, as named by Kravaris and Soroush [77] or decoupling matrix as named by Slotine and Li [14] of the system, and is given by:

$$E = \begin{bmatrix} \beta_1 & 0 & \beta_2 \\ 0 & \beta_3 & \beta_4 \\ \beta_5 & \beta_6 & \beta_7 \end{bmatrix} \tag{3.72}$$

which has been checked, is nonsingular.

In a similar way to Section 3.3 for **Design 1**, the equations:(3.73), (3.74) and (3.75) are detailed here again to show the tracking closed loop design. Here the linearized closed loop system for each channel is given by:

$$\ddot{y}_i = v_i \tag{3.73}$$

where (v) is the new linearized system input and is given by:

$$v = \ddot{y}_d - k_1\dot{e} - k_2 e \tag{3.74}$$

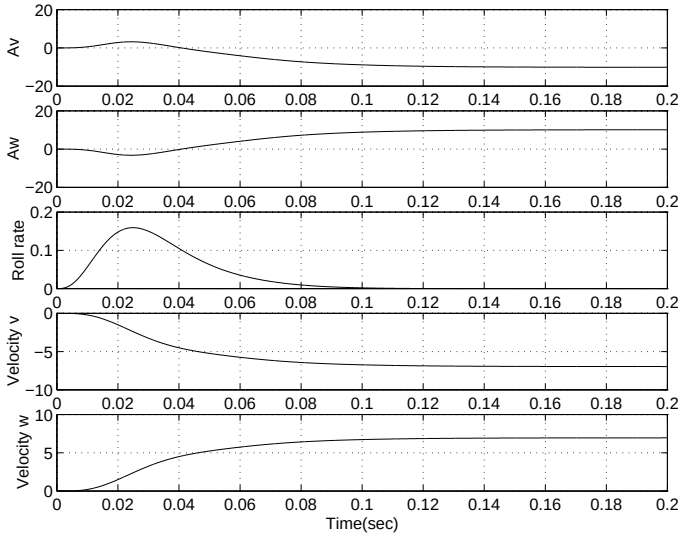
and where $e = y - y_d$. The closed-loop system for each channel is thus characterised by the following second order error dynamics:

$$\ddot{e} + k_1\dot{e} + k_2 e = 0 \tag{3.75}$$

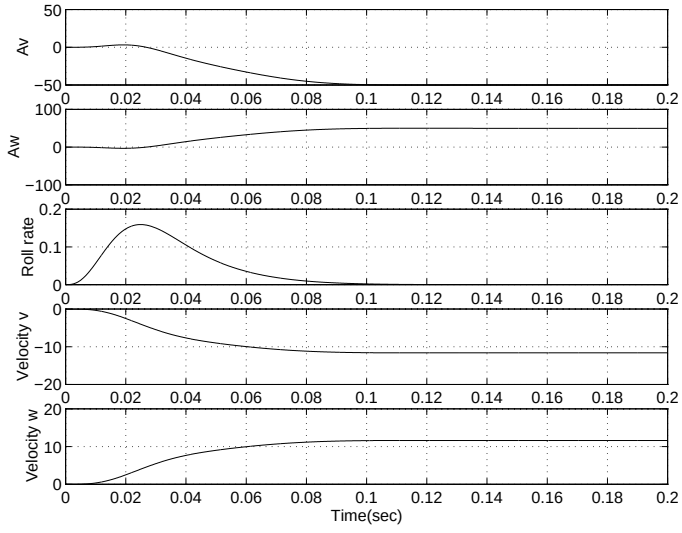
where k_1 and k_2 are chosen such that all roots of $s^2 + k_1 s + k_2 = 0$ are Hurwitz in the open left-half plane, which ensures $\lim_{t\to\infty} e(t) = 0$, as detailed by Wang [76], hence the tracking control problem for the non-linear MIMO system has been solved. The stability of the linearized system has been guaranteed since no zero dynamics has been involved.

The trajectory control design has got the same structure as shown in fig.3.2 for **Design 1** for the SISO system, but with an additional output (p) for the roll channel. Again a fast linear actuator with natural frequency of 250 rad/sec has been included in the non-linear system. The desired lateral acceleration a_d for each channel is achieved by using the non-linear equation $a_d = f(v)$. Therefore the trajectory controller performs by using the desired acceleration as a function of the lateral velocity demand. The error dynamics are constructed using the a_d signal and the feedback of the actual states - velocity, rate and acceleration. The error coefficients in (3.75) for the trajectory controller are chosen to satisfy Hurwitz polynomial.

The results for 1g (10 m/sec^2) and 5g (50 m/sec^2) lateral acceleration demands are shown in fig. 3.7. Fully decoupling has been achieved in yaw, pitch and roll channels, as detailed by Tsourdos et al [81]. Both figures show desired tracking performance as the predicted and the actual performance are very close, with almost no steady state error. The non-linear relationship between side-slip (or vertical) velocity and lateral acceleration for both (yaw and pitch) channels can also be seen.



Accelerations, velocities and roll rate for 1g demand



Accelerations, velocities and roll rate for 5g demand

Figure 3.7: Results for **Design 1** in MIMO for $a_d = 10\ m/sec^2$ and $a_d = 50\ m/sec^2$

3.4.2 Design 2: Polar coordinates

The aim of this section is to track the missile lateral acceleration demand in both the pitch and yaw plane as well as the roll rate in the roll plane, using the missile aileron, rudder and elevator, hence yielding a system with 3 inputs and 3 controlled outputs. The tracking and the non-linear controllers are designed by defining lateral velocities (v) and (w) as outputs as they produce higher relative degree than directly controlling lateral acceleration, which has a relative degree of zero. Lateral velocity is directly related to the lateral acceleration, as in steady state a constant incidence angle is associated with a constant lateral acceleration. The basic system is fifth order, with an integrator in front of the roll channel yielding a sixth order system.

The missile system is transformed in Polar coordinates, with the flight direction given by $z = \sqrt{v^2 + w^2}$ and the angle of orientation given by $\lambda = arctan\frac{v}{w}$. These transformations are used to simplify in a significant manner the heavy computational load required by the nonlinear control law derivation, (see equations (3.64) and (3.66).

The equations of motion described in (2.43) in Chapter 2 are used to derive the state-space form of the non-linear system in a compact parametric format, as:

$$\begin{aligned}
\dot{z} &= a_1 z + a_2 z^2 + a_3 r sin(\lambda) - a_3 q sin(\lambda) \\
&\quad + (a_4 z + a_5)(sin(\lambda)\zeta + cos(\lambda)\eta) \\
\dot{r} &= b_1 z^3 sin(\lambda) + b_2 z^2 sin(\lambda) + b_3 z sin(\lambda) + b_4 z r + b_5 r \\
&\quad + (b_6 z + b_7)\zeta - (b_9 + b_8 z + b_1 0 z^2)\xi \\
\dot{q} &= -b_1 z^3 cos(\lambda) - b_2 z^2 cos(\lambda) - b_3 z cos(\lambda) + b_4 z q + b_5 q \\
&\quad - (b_6 z + b_7)\eta + (b_9 + b_8 z + b_1 0 z^2)\xi \\
\dot{p} &= c_1 p + c_2 \xi + (c_3 + c_4 z)(\zeta + \eta) \\
\dot{\lambda} &= -a_3 z^{-1}(q sin(\lambda) + r cos(\lambda)) \\
&\quad + z^{-1}(a_4 z + a_5)(sin(\lambda)\eta - cos(\lambda)\zeta)
\end{aligned} \tag{3.76}$$

The nonlinear system written in a standard form is:

$$\begin{aligned}
\dot{x} &= f(x) + g(x)u \\
y &= h = \begin{bmatrix} v \\ w \\ p \end{bmatrix} = \begin{bmatrix} z sin(\lambda) \\ z cos(\lambda) \\ p \end{bmatrix}
\end{aligned} \tag{3.77}$$

and Input/Output Linearization technique can be applied to it. In order to retain the system order with no zero dynamics, an approximate Input/Output Linearization technique is applied to the missile model. It is based on an approximation method involving the modification of the function g, as presented by Hauser et al [74]. Using the approximation technique, terms are discarded in order to retain an approximate system with an equivalent order and relative degree. In other words

the g vector field is modified. This is achieved by neglecting the terms $\psi_1(x,\zeta)$ and $\psi_2(x,\eta)$ shown in the following equations.

For the yaw plane let $\mu_1 = \phi_1 = h_1(x)$. Then:

$$\begin{aligned} \dot{\mu}_1 &= \mu_2 + \psi_1(x,\zeta) \\ \dot{\mu}_2 &= \alpha_1 + \beta_1\zeta + \beta_2\xi = v_1(x,\zeta,\xi) \end{aligned} \tag{3.78}$$

where:

$$\begin{aligned} \alpha_1(x) &= (a_1 cos(\lambda) + 2a_2 z cos(\lambda))(a_1 z + a_2 z^2 + a_3 r sin(\lambda) - a_3 q sin(\lambda)) \\ &\quad +(a_1 z cos(\lambda) + a_2 z^2 cos(\lambda))(-a_3 z^{-1}(q sin(\lambda) + r cos(\lambda))) \\ &\quad +a3(b_1 z^3 sin(\lambda) + b_2 z^2 sin(\lambda) + b_3 z sin(\lambda) + b_4 z r + b_5 r) \\ \beta_1(x) &= a_3(b_6 z + b_7) - (b_9 + b_8 z + b_{10} z^2) \\ \beta_2(x) &= a_3(b_9 + b_8 z + b_{10} z^2) \end{aligned} \tag{3.79}$$

Hence the output $h_1(x)$ possesses a relative degree r_1 of 2.

For the pitch plane let $\mu_3 = \phi_3 = h_2(x)$. Then:

$$\begin{aligned} \dot{\mu}_3 &= \mu_4 + \psi_2(x,\eta) \\ \dot{\mu}_4 &= \alpha_2 + \beta_3 u_2 + \beta_4 u_3 = v_2(x,\eta,\xi) \end{aligned} \tag{3.80}$$

where:

$$\begin{aligned} \alpha_2(x) &= (a_1 sin(\lambda) + 2a_2 z sin(\lambda))(a_1 z + a_2 z^2 + a_3 r sin(\lambda) - a_3 q sin(\lambda)) \\ &\quad -(a_1 z sin(\lambda) + a_2 z^2 sin(\lambda))(-a_3 z^{-1}(q sin(\lambda) + r cos(\lambda))) \\ &\quad -a3(-b_1 z^3 cos(\lambda) - b_2 z^2 cos(\lambda) - b_3 z cos(\lambda) + b_4 z q + b_5 q) \\ \beta_3(x) &= a_3(b_6 z + b_7) - (b_9 + b_8 z + b_{10} z^2) \\ \beta_4(x) &= a_3(b_9 + b_8 z + b_{10} z^2) \end{aligned} \tag{3.81}$$

The output $h_2(x)$ also possesses a relative degree r_2 of 2.

Finally, for the roll plane, for the linearization process (i.e. the design of the non-linear controller), we take as output the roll rate p, but place an integrator in front of the roll channel to equalize the channel orders.

Let $\mu_5 = \phi_5 = h_3(x)$, where $h_3(x)$ is the roll angle. Then:

$$\begin{aligned} \dot{\mu}_5 &= \mu_6 \\ \dot{\mu}_6 &= \alpha_3 + \beta_5\zeta + \beta_6\eta + \beta_7\xi = v_3(x,\zeta,\eta,\xi) \end{aligned} \tag{3.82}$$

where:

$$\begin{aligned} \alpha_3(x) &= c_1 \\ \beta_5(x) &= c_3 + c_4 z \\ \beta_6(x) &= c_3 + c_4 z \\ \beta_7(x) &= c_2 \end{aligned} \tag{3.83}$$

Hence the output $h_3(x)$ possesses a relative degree r_3 of 2. The total relative degree of the system is equal with the sum of the r_1, r_2, and r_3 is now 6, and has the same order as the original system and hence there are no internal dynamics. Since the total relative degree is equal with the order of the system, fully linearization of the non-linear system has been achieved.

The effect of neglecting the terms (ψ_1) and (ψ_2) in the previous equations is to eliminate a non-linear zero in the system within the model description, and which is not taken into account in the non-linear control design. This will not affect the performance of the control design in a significant manner as the zero can be approximated by:

$$z \approx -\frac{(a_4 z + a_5)}{(2a_3 b_6 z + a_3 b_7)} \tag{3.84}$$

Equations (3.78), (3.80) and (3.82) represent a direct relationship between the outputs (h_i) and the inputs (u_i). The required static state feedback for decoupled closed loop Input-Output behaviour is given by Slotine and Li [14] as:

$$u = E^{-1} \left\{ v - \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \right\} \tag{3.85}$$

where E^{-1} is the characteristic or the decoupling matrix of the system, and here is determined by:

$$E = \begin{bmatrix} \beta_1 & 0 & \beta_2 \\ 0 & \beta_3 & \beta_4 \\ \beta_5 & \beta_6 & \beta_7 \end{bmatrix} \tag{3.86}$$

which is nonsingular. The determinant of the decoupling matrix is:

$$p(z) = det(E) = p_0 + p_1 z + p_2 z^2 + p_3 z^3 + p_4 z^4 + p_5 z^5$$

All the roots are complex. There is no value of interest for z which could make $p(z)$ (i.e. the determinant of the decoupling matrix) equal to zero.

The linearized closed loop system for each channel is given by:

$$\ddot{y}_i = v_i \tag{3.87}$$

where (v) is the new linearized system input and for tracking problem can be chosen to be:

$$v = \ddot{y}_d - k_1 \dot{e} - k_2 e \tag{3.88}$$

where $e = y - y_d$. The closed-loop system is thus characterised by:

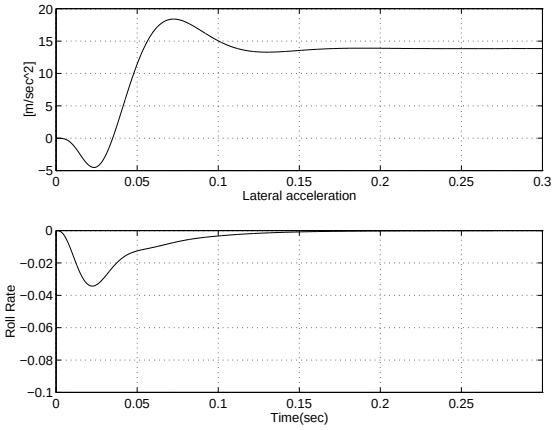
$$\ddot{e} + k_1\dot{e} + k_2 e = 0 \tag{3.89}$$

where k_1 and k_2 are chosen such that all roots of $s^2 + k_1 s + k_2 = 0$ are in the open left-half plane, which ensures $\lim_{t\to\infty} e(t) = 0$.

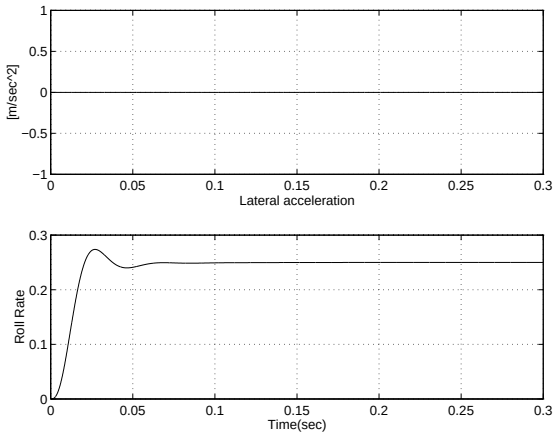
Like in the first **Design 1** for the SISO system, a fast linear actuator with natural frequency of 250 rad/sec has been included in the non-linear system. The desired acceleration a_d has been achieved by using the non-linear equation $a_z = f(v, w)$, but in a Polar sense. The desired acceleration is a function of magnitude (z) of the lateral velocities: $a_z = a_1 z + a_2 z^2 = a_1(\sqrt{v^2 + w^2}) + a_2(v^2 + w^2)$ and it is used in the feedback to construct the error dynamics.

The error coefficients in (3.89) are chosen to satisfy a Hurwitz polynomial. For the acceleration channel, $k_1 = 2\zeta w_n$ and $k_2 = w_n^2$ are chosen with $w_n = 60$ rad/sec and $\zeta = 0.7$, for the roll channel with $w_n = 80$ rad/sec and the same damping factor, ζ. This speed of response is significantly faster than the open loop response and so should exercise the dynamics of the non-linear missile. The tracking control problem for the non-linear system has been solved using the control law in equation (3.85). Since the equation (3.89) has the same order as each channel of the non-linear system, there is no part of the system dynamics which is rendered "unobservable" in the approximate Input/Output Linearization. Since there are no zero dynamics in the linearized system, the stability of the linearized system has been guaranteed and the tracking problem has been solved. Desired tracking performance for lateral accelerations and roll angle of the missile has been achieved by using a non-linear control law that has been derived by selecting lateral velocities and roll rate as the linearization outputs. This has been detailed by Tsourdos et al [82].

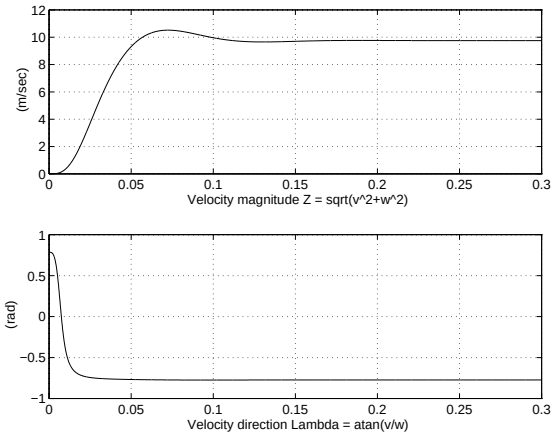
Finally, simulation results are shown in fig. 3.8 that exercise the final design and show that the linearization and the controller design are satisfactory. When there is no lateral acceleration demand, shown in fig. 3.8c, a constant roll rate demand, resulted in zero velocity magnitude which is a good indication for fully decoupled system. Also a constant roll rate demand (the ξ input on the roll channel) had no effect on the yaw and pitch channels.



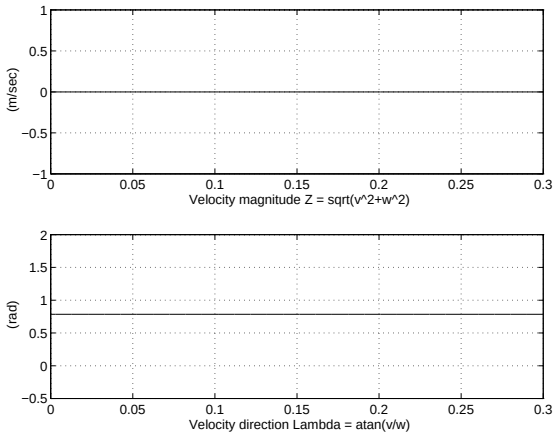
a) Lateral accelerations $a_d = 14$ and roll rate $p_d = 0$



c) Lateral accelerations $a_d = 0$ and roll rate $p_d = 0.25$



b) Magnitude and direction of velocity for $a_d = 14$ and $p_d = 0$



d) Magnitude and direction of velocity for $a_d = 0$ and $p_d = 0.25$

Figure 3.8: Results for **Design 2** in MIMO for $a_d = 14\ m/sec^2$ and $p_d = 0\ rad/sec$

3.5 Conclusions

There are three ways to increase the relative degree of a non-linear system. These are either to propose a new output that is an approximation of the desired one, to neglect sufficiently small terms during the differentiation process or finally to design a pre-compensator for the system.

All four designs (SISO and MIMO), presented in Chapter 3, have used a combination of the first two. By neglecting small terms associated with the fin deflection which modifies the g vector and by defining outputs for the linearization procedure which are related with the controlled outputs, an approximate Feedback Linearization technique has been successfully applied. The design has resulted in a linear equivalent system with no internal or zero dynamics ("no unobservable" states during the linearization), and with a design of a trajectory control which gives small tracking errors for both lateral velocities and accelerations. The simulation results have shown desired tracking performance for a large range of 1g up to 10g lateral acceleration demands (for SISO and MIMO systems) and roll control (for the MIMO system).

When the augmented acceleration was chosen as an output for the linearization process, in **Design 2** for the SISO system, the nonlinear control law involved more complex mathematics and more nonlinear terms than the nonlinear control law in **Design 1**. On the other hand the relationship (augmented acceleration - lateral acceleration) is linear, so differences in closed loop performance for higher demands are small, and only in the steady state error. Also, the augmented lateral acceleration is used in **Design 2** for the SISO system, provided that the direct acceleration produced by the fin is small compared to the augmented acceleration. It also shown that a neglected zero during the linearization process was minimum phase.

Two ways of manoeuvring the missile motion have been proposed by **Design 1** and **Design 2** for the MIMO system. Although the Horton model has been designed for Cartesian control, Polar control is also possible to be designed because it can significantly reduce the computational load of the nonlinear control design, which can be important (less risky and less expensive - computationally speaking).

Finally, full decoupling for the highly non-linear missile system has been achieved. All four Designs (SISO and MIMO) have involved increasing the speed of responses of the system sufficiently and the responses for both small and large demands have shown to be invariant. Other techniques have been researched by White [71] that involve a quasi-linear approach, or involve pre-compensation to look at techniques that can be applied to the lateral acceleration directly. This involves dealing with a non-minimum phase system that yields unstable zero dynamics with direct linearization methods.

Chapter 4

Robust Fuzzy Autopilot Design

It has been shown in the previous chapter that by applying Feedback Linearization the desired tracking performance can be obtained by assuming an exact knowledge of aerodynamic coefficients and missile configuration parameters (i.e., reference area, Mach number, mass, moment of inertia) in the entire flight envelope. In practice however, this assumption is not valid. Also, if there are either parameter variations from the nominal case or external disturbances, the Feedback Linearization cannot guarantee desired performance, neither is robustness provided.

Conversely, fuzzy logic appears promising when dealing with vague and imprecise information such as uncertain measurement values, parameter variations and noise. For these reasons, a robust non-linear trajectory controller based on fuzzy logic has been applied in the outer loop in order to provide robustness for the feedback linearizable system. An evolutionary algorithm optimisation approach is then applied off-line to determine the membership function distribution and the rule base structure of the fuzzy controller. The design uses a genetic algorithm optimisation approach using a multiple model description of the airframe aerodynamics and meets objectives related to closed loop performance such as: steady state error, overshoot, rise and settling time.

The aim of Chapter 4 is to track the missile side-slip velocity demand in the presence of uncertainties in the aerodynamic coefficients. The required demands are considered for both pitch and yaw planes, using the missile rudder and elevator as control surfaces hence only lateral motion is considered, yielding two uncoupled systems with one input and one controlled output each. Multiple demand tracking is also addressed here.

4.1 Hybrid Fuzzy Nonlinear Control

"*Everything is a matter of degree and you do not realize it till you have tried to make it precise*". Bertrand Russell

4.1.1 Fuzzy Logic philosophy

"Fuzziness means *multi-valence*. It means infinite shades of grey between black and white. Fuzzy things resemble fuzzy non-things "*A resembles not A*" and have vague boundaries with their opposites, their non-things. The more *a thing* resembles its opposite the fuzzier it is", by Kosko [83].

Fig. 4.1.1 stands for the world of opposites. The maths language creates boundaries between black and white. Reason smoothes them out as it works with grey. Borders are inexact and things coexist with non-things. Fuzzy logic is reasoning with fuzzy sets. Fuzzy logic deals with ambiguous events or situations. However, ambiguity does not mean there is no sort of certainty in the events or situations. For example probability did not alter or even challenge the black-white *picture of the world. It just showed how to gamble in it.*[83]

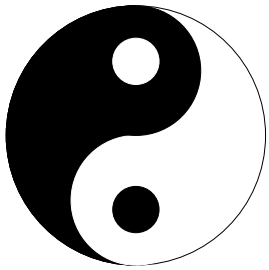


Figure 4.1.1 The Yin-Yang symbol

Fuzzy Logic is a mathematical discipline developed by Zadeh [32] based on fuzzy set theory which allows for degrees of truth and falseness. Fuzzy control is based on fuzzy logic and provides a means of converting a linguistic control strategy based on expert knowledge into an automatic control strategy, as detailed by Lee [33]. Fuzzy logic maps a set of inputs called antecedents to a set of control command outputs called consequents, which actuate devices to translate the system to the desired state. Because of the multi-valued nature of fuzzy logic, the values of the system states can be categorically described by linguistic variables which maintain the intuitive knowledge for the system. For example, rates may be described as positive fast or negative slow and control actions classified as negative large or positive medium. The major advantage of fuzzy logic over conventional control algorithms is that systems can be controlled, based on the designer's experience (input and output observations), not on the theoretical methods, which implies that there is no need to rely on precise models. Fuzzy inferencing provides the means of systematically synthesizing various fuzzy rules to produce decision actions so that complex non-linear systems can be controlled. In addition, the ability to control a system in

an *uncertain* environment is an important feature, which is the main reason to be used here for solving our problem. Fuzzy logic has been used in control for many years. Engineers have successfully applied fuzzy systems in many commercial areas. Fuzzy systems "intelligently" automate subways, focus cameras, tune colour television, control automobile transmissions, defrost refrigerators, control air conditioners, automate washing machines and vacuum sweepers, guide robot arm manipulators, control traffic lights, elevators and cement mixers. Most of these applications originated in Japan, and have been sold and applied throughout the world. Some details are given by Kosko [83], Bonivento et al [84], Palm and Driankov [85], Mcneill and Freiberger [86].

Linear control techniques are mainly useful for linear systems. Since we are dealing with a nonlinear plant, conventional techniques will not be appropriate to use here. Many processes controlled by human operators in industry cannot be automated using conventional, linear control techniques, since the performance of these controllers is often inferior to that of the operators.

Conversely, knowledge-based control techniques try to formalise the domain-specific knowledge, and use reasoning mechanisms for determining the control action from the knowledge stored in the system and from the available measurements, as given by Palm and Driankov [85]. These control systems try to enhance the performance, reliability and robustness of the current control system. Fuzzy Logic Controllers (FLCs) are rule-based control systems where fuzzy sets are used for specifying qualitative values of the controller inputs and outputs. The experts knowledge contains linguistic terms such as *negative (Neg), zero (Z), positive (Poz)* of the error variable and can be represented by fuzzy sets (see fig. 4.1).

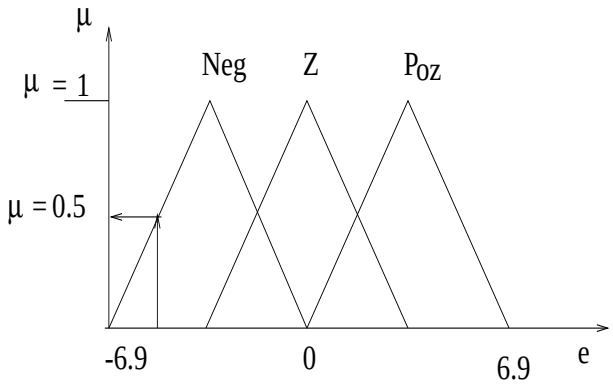


Figure 4.1: Membership functions defined for the error variable

The membership functions shown in fig. 4.1 provide a smooth interface from the linguistic knowledge to the numerical process variable.

Using fuzzy sets and fuzzy operations it is possible to design a fuzzy reasoning system which can act as a controller, as illustrated in fig.4.2. The control strategy is stored in the form of **if-then** rules in the rule base. The rules represent an approximate static mapping from inputs (e.g. errors) to outputs (control actions) (see Fuzzy Logic Toolbox [87]) and for example can be formulated as follows:

If error is *negative medium* **then** control action is *positive small*.

The first part of the rule, called the antecedent, specifies the conditions under which the rule holds, while the second part, called the consequent, describes the corresponding control action. Both the antecedent and the consequent contain linguistic terms (*large,small,near zero etc.*) that reflect the experts knowledge of the process. The antecedent condition is defined as a combination of several individual conditions, using a connective, such as the logical AND, OR operations. The reader may refer to [87] for further understanding.

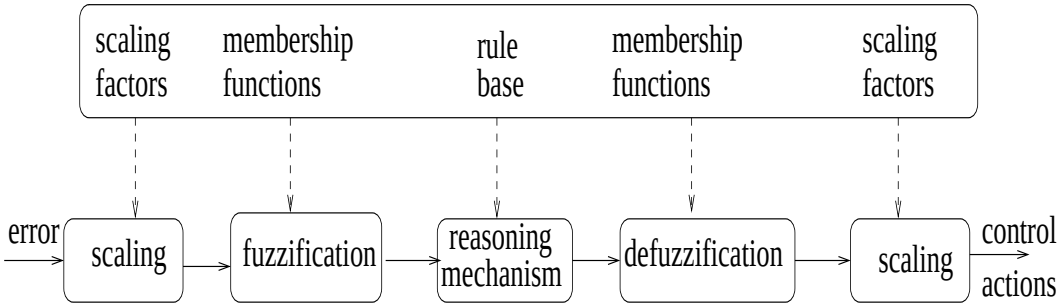


Figure 4.2: Block-schematic representation of a fuzzy logic controller

The fuzzification module determines the membership degree of the inputs to the antecedent fuzzy sets. The reasoning mechanism combines this information with the rule base and determines the fuzzy output of the rule base system. In order to obtain a crisp signal, the fuzzy output is defuzzified using several techniques to produce a single continuous variable.

Fuzzy sets and Membership functions

A fuzzy set is defined as a set with degree of membership associated with each member. It is a set of ordered pairs which associate each value of the variable to its grade of membership in the set. The grades of membership are represented by the membership function μ_A. Consider a universal, crisp set U, called the universe of discourse and a fuzzy set A. The membership function μ_A maps the elements $x \in U$ into real numbers in $[0, 1]$:

$$\mu_A(x) : U \longrightarrow [0, 1]$$

which gives a measure of the grade of membership of x which belongs to U in the fuzzy set A. The position and shape (triangular or bell shaped) of μ_A depend on the particular application. For a PD controller there is no difference between different shapes, as given by Hamm [88]. Fig. 4.1 shows an error of 1.75 which belongs 50% to the set of Z and 50% to the set of Poz. The μ degrees of the fuzzy sets Z and Poz are both 0.5, which is an orthogonal condition and it has been considered in our work. It has been shown by Lotfi [89] that membership functions have a dominant effect on the reasoning process rather than the number of rules or the inference mechanism.

Fuzzy set operations

Fuzzy set operations are performed by logical connectives such as:

AND $\mu_A(x) = min(\mu_A(x), \mu_B(x)) = \mu_A \wedge \mu_B$,

OR $\mu_A(x) = max(\mu_A(x), \mu_B(x)) = \mu_A \vee \mu_B$,

NOT $\mu_A(x) = 1 - \mu_A(x)$.

In our work, the minimum operator is used for conjunction and the maximum operator for disjunction. The Mamdani method is used for our fuzzy inference system, i.e. the min operator rule is adopted for the logic AND operator. For example, the value W^l of the antecedents (A_1^l and A_2^l) of the l^{th} rule (A_1^l, A_2^l, B^l) is calculated as:

$$W^l = min(\mu_{A1}, \mu_{A2}) = \mu_{A1}(S_e e) \wedge \mu_{A2}(S_{de}\dot{e})$$

which is the degree of fulfillment of the l^{th} rule, where $\mu_{A1}(S_e e)$, $\mu_{A2}(S_{de}\dot{e})$ are the membership grades of the scaled variables in fuzzy sets A_1, A_2 and S_e, S_{de} are the scaling factors for the input variables with $\wedge$, the min operator. The most common method for determining the output value for each control in the vector u is by calculating the centroid of where its membership function values are acting along the output control's universe of discourse. There are many possible ways to defuzzify an output. The centre of area can be used for defuzzification and the output is given by:

$$y^o = \frac{\sum W^l u_i^l}{\sum W^l}$$

where u^l is the center of the l^{th} rule's consequent fuzzy set B^l, i.e. $\mu_B^l(u_i^l) = 1$. The crisp perturbation control is given by $u = S_u y^o$, where S_u is the scaling factor for the control output u. Each rule is weighted by the degree to which the antecedent of the rule is fulfilled. The final control decision is obtained as the weighted average of all the contributed conclusions.

Adjustment of membership functions and rules for a fuzzy controller is examined in detail by Hamm and Splettstoser [88]. A detailed procedure for selecting the type and the number of μ_A for each domain has been considered. They have found that

up to five μ_A is easier to design and optimise. If there are more than two inputs or if each input has many terms, then the number of rules can increase dramatically.

A very useful idea of applying PD fuzzy control for vehicle tracking has been investigated by Chiu et al [40]. The derivative control rules are given much smaller influence than the proportional control rules to avoid over-damping, in the same way that the derivative gain is typically smaller than the proportional gain in conventional linear control. The derivative control actions are predicated upon the condition that the rate error is near zero. The resultant behaviour is that the controller would not impose damping until the vehicle approaches the commanded roll rate. Their control strategy has pushed the vehicle toward the commanded roll rate at maximum acceleration, and applied damping to stabilize the vehicle only during close tracking.

Fuzzy logic has been successfully applied in combination with other techniques. An Input/Output Linearization with an adaptive fuzzy outer loop has been applied to the depth control of a nonlinear underwater vehicle by Trebi-Ollennu and White [90]. The adaptive fuzzy systems are Sugeno type and have been used to approximate the uncertainties caused by forward speed variations in order to improve the robustness properties. This control scheme has enhanced the closed loop performance by reducing the output tracking errors and by adding "intelligence" to the conventional Input/Output controllers.

A hybrid approach, integrating Feedback Linearization and FLC (FL/FLC), has been proposed by Lin and Gau [91] for improving the transient performance and robustness of a highly nonlinear and open loop unstable magnetic bearing system. The disturbance rejection capability of FL/FLC was much better than only the FL approach. Rotor speed trajectory and gap deviation regulation have been considered. The nine output variables of the system were transformed to nine linear decoupled subsystems with no internal dynamics. For each of these systems, 7 μ_A input-output variables were used to produce a 49 rule base structure of the FLC. However, the FLC parameters were tuned by using extensive computer simulations (e.g. the trial and error method) which can be very computationally expensive.

A very good control design approach has been investigated by Kwan et al [92] for a pitch autopilot for a simple missile model. They have used on-line tuning of a fuzzy CMAC neural network to improve the robustness of Feedback Linearization. The fuzzy logic has been used to produce a systematic way of adjusting the neural network weights on-line. No off-line training phase was needed which is an interesting achievement. However, an increased complexity of the control system is associated with such a design.

An interesting approach has been proposed by Leland [93] for using Feedback Linearization to design controllers for systems with fuzzy uncertainties. Instead of considering bounded uncertainty, they have used a fuzzy uncertain model. The Feedback Linearization has provided asymptotic stability for the controller.

4.1.2 Fuzzy trajectory controller for the feedback linearized system

A Feedback linearized system with fixed gain trajectory controller has been designed in Chapter 3. The nominal model of aerodynamic coefficients has been considered. In order to design an appropriate fuzzy logic controller which can deal with the non-linear parametric uncertainties of the missile model, we have to reproduce the dynamic behaviour of the Input/Output linearizable controller by replacing the fixed gain trajectory controller with a FLC type. At the start, the model has been kept with the exact knowledge of the aerodynamic coefficients and the missile configuration parameters, so initial fuzzy rules have been derived. One input-one output (i.e. error-control action) FLC with only five rules has been derived. The contribution of the fuzzy logic trajectory controller has significantly improved the transient response. Almost no steady state error and smaller overshoot have been achieved, conversely to the design with the fixed gain trajectory controller. Each variable of the FLC has five membership functions symmetrically placed within the Universe of Discourse. A two input-one output FLC has also been designed taking into account the derivative action of the error. This resulted in steady state error. The trial and error mechanism has been used with many iterations before an appropriate rule base structure has been achieved which is time demanding and not very practical. Hence it has been replaced by evolutionary optimisation using a genetic algorithm for better adaptation.

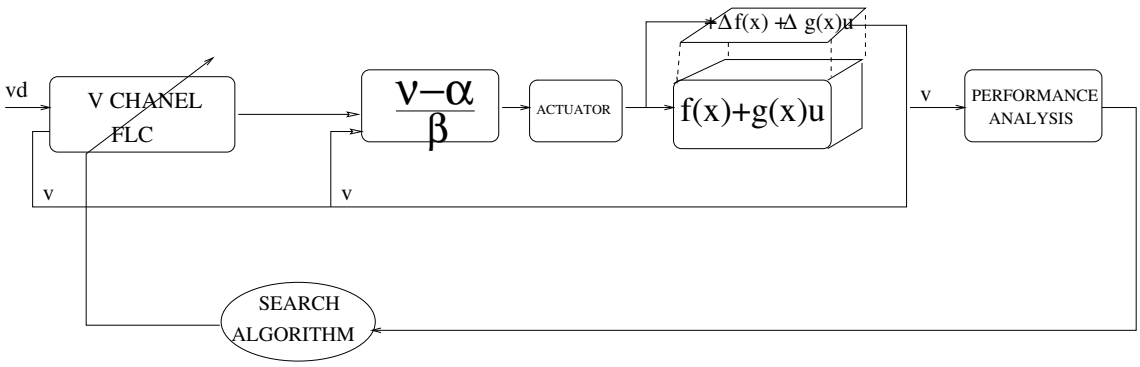


Figure 4.3: Fuzzy-Feedback Linearized Autopilot Design

The autopilot design is shown in fig. 4.3. The missile dynamics are represented by a multi-modelling format:

$$\begin{aligned} \dot{x} &= f(x) + \triangle f(x) + (g(x) + \triangle g(x))u \\ y &= h(x) \end{aligned} \tag{4.1}$$

where $\triangle f(x)$ and $\triangle g(x)$ are considered as uncertainties caused by the aerodynamic coefficients $(C_{yv}, C_{y_\zeta}, C_{nr}, X_{cp})$. In this chapter the models are randomly generated

polynomials within a large range of 0° to 45° roll angle. Fast 250 $rads/sec$ second order linear actuator representing rudder is included within the missile dynamics. The non-linear control law $u = \frac{\nu - \alpha}{\beta}$, is derived by the feedback linearization technique, as detailed in equation 3.47, Section 3.3.1 of Chapter 3. The selected output for the linearization process is the side-slip velocity. A fuzzy logic trajectory controller is used in the outer loop for the side-slip velocity, V CHANEL. The trajectory controller is designed, based on fuzzy inference engines, as two inputs - one output system with four membership functions for each variable. An optimisation algorithm is used to generate the fuzzy control parameters (i.e. membership functions and rule base structure), while the non-linear controller $u = \frac{\nu - \alpha}{\beta}$ remains fixed. The obtained fuzzy controller is tested on five trials (i.e. randomly generated models). Then a performance analysis is done off-line for each autopilot simulation. Four closed loop performance criteria are considered (i.e. steady state error, settling time, rise time and overshoot). The maximum objective value of the five trials is returned to the optimisation algorithm for evaluation of the tested fuzzy controller. The optimisation process repeats for large number of iterations until satisfactory closed loop performance of the autopilot system is obtained.

4.2 Optimisation of the Fuzzy Logic Controller

One of the major drawbacks of fuzzy logic controllers is that the membership functions are chosen arbitrarily which implies a need of using a "trial and error" design philosophy to improve the closed-loop system's behaviour, which may not always be possible. An evolutionary optimisation technique is suggested and described in the next section as a possible way to tune the FLC parameters. A surrogate additive function which transfers the vectorised multi-objective problem into a scalar optimisation problem is used here.

4.2.1 Evolutionary Algorithm

The membership functions and rule base structure of a fuzzy controller can be defined by trial and error. However, there is a need for a suitable learning medium in order to increase the robustness of the FLC. The choice of learning method is dictated by the nature of the task domain and the available information. One possible way would be the use of Neural Networks (NNs), as detailed by Linkens and Nyongesa [31]. They depend highly on the availability of sufficient data representing the input-output mapping, but in a situation where such data cannot be obtained an alternative approach is necessary. One such approach is to test hypothetical trial solutions of the system and generate better solutions on the basis of the performances using evolutionary techniques. Genetic algorithms (GAs), which are modelled on natural evolutionary strategies, based on *Darwinian principle* of survival of the fittest in biological reproduction, as described by Goldberg [94], are one possible methodology that can be used as a learning and optimisation technique under such conditions. They are capable of finding global solutions when

employed in noisy search spaces, whereas NNs can only provide fine tuned adjustments using local search. The operators GA use, to direct them through the search space, have features for self repair, self guidance and reproduction which are found in natural genetics of biological systems. Whenever robust performance is required nature does it better. GAs are theoretically and empirically proven to provide robust search in complex spaces. They have been successfully applied to a variety of problems such as function optimisation, control, identification, self adaptive and learning systems. The reasons for a large number of applications are because GAs are computationally simple and powerful in their search engines. Also they are not limited by restrictive assumptions concerning continuity, existence of derivatives or unimodality. Other optimisation techniques are shown by Rao [95], such as Calculous based (A), Enumerative (B) and Random (C). Some are local in scope (A) and use point by point search (A,B and C), hence converging to a local optima. Such methods depend strongly upon the restrictive requirements mentioned above and are suitable for a very limited problem domain. Conversely GAs consider many points from the search space simultaneously (a population of strings climbing many peaks in parallel) which preserve the probability of converging to global optima. Also they only need the objective function values associated with each individuals to asses the quality of the solution. Unlike many methods, GAs use probabilistic transition rules to guide their search. They use random choice as a tool to guide the search toward regions with likely improvement and have problem-independent characteristics of the search scheme, which enables black-box treatment of the GA code. That is the GA supplies the parameters to the optimisation problem and in return, the software provides the fitness function which is then utilized by the GA to evolve the next generation. Goldberg has given a very good example in [94] of the black box optimisation problem with on-off switches illustrating the idea of a coding and a payoff measure.

How GA's work

The basic cycle for GAs is illustrated in fig. 4.2.1. It starts by randomly generating a population of individuals (strings) which are then evaluated by some fitness function. Then selection takes over to reproduce new individuals by using GA operators to create a new generation of possible solutions. Each string represents one possible solution to the problem.

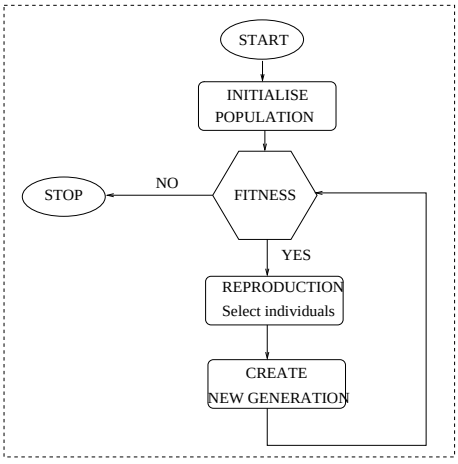


Figure 4.2.1 Simple GA structure

GAs work iteration by iteration, generating and testing a population of strings. This population by population approach is similar to a natural population of biolog-

ical organism, where each generation successively evolves into the next generation by being born and raised until it is ready to reproduce. Optimal strings are found through population reproduction via selection, crossover and mutation. Selection is based on stochastic universal sampling and is the process where an old string is carried through into a new population depending on its performance index (fitness function) value. So strings with above average fitness values get larger numbers of copies in the next generation. This strategy, in which good strings get more copies in the next generation, emphasizes the survival of the fittest concept of GA. A crossover phase then follows. Crossover exchanges information between the selected strings paired at random (i.e. between two search points). The mutation operator is an occasional random alteration of a string position for binary genes (based on probability of mutation). For real genes, it mutates each variable from the population with a given probability. The mutation operator helps to avoid local minimum, which is very important.

In summary, the search algorithm has inherent parallelism which enables rapid search of the high-performance regions of complex domains such as a fuzzy logic control structure. GAs have been recognised to be a powerful tool for learning the control rules and tuning their membership functions: Bonivento et al [84], Bica et al [96]. An important point to be mentioned here is that a good solution depends on setting the objective function correctly. However a major drawback of the technique is that GAs are computationally inefficient as many trials are necessary until reasonable good solutions are found. But with the new high speed technology such as UNIX stations, high performance computers (e.g. Cranfield University SGI CRAY ORIGIN 2000 supercomputer), GAs are able to produce fast solutions. As a conclusion we can highly recommend that this technique can be useful for generating fuzzy control parameters of a non-linear missile.

4.2.2 GA tuning the FLC parameters

The steps for tuning the FLC are as follows:

First, the scaling factors (SF) for inputs and outputs of the FLC are determined based on observation of the error, derivative of error and output responses of the fixed gain trajectory controller for the closed loop system with the nominal model. The domains of $(-SF$ to $+SF)$ are the most important parameter of the μ_A tuning, as given by Hamm and Splettstoser [88]. The effect of the domain of a fuzzy variable is exactly the same as that of the gain factors of a non fuzzy controller. Changing the error domain affects rise time and overshoot about three times as much as changing the domain of the derivative of error [88]. However the SFs are not included in the optimisation procedure in our work, they are not part of the chromosome structure.

Second, the membership functions have been shown to be more important to tune, rather than the rule base parameters, as detailed by Driankov et al [35]. A modified term in a term set affects one row, column or diagonal in the rule table, while a mod-

ified rule only affects a table cell. In our work, the membership functions for each universe of discourse have been chosen standard and uniformly spreaded. Initially, they are uniformly positioned triangles overlapping at a 50% level over the normalised universe of discourse. Since the controller is defined by a nonlinear control surface in $(e, \delta e, u)$ space, three term sets for each variable $(e, \delta e, u)$ are designed. At the start, the μ_A distribution is symmetric, and after the optimisation the μ_A distribution is asymmetric. In other words by changing the distribution of these terms within the control variables domain, the design algorithm has been adjusting the gains of the trajectory fuzzy outer loop.

The existing iterative approaches for choosing the membership functions μ_A are manual trial and error process and lack learning capability and autonomy. The automatic generation of fuzzy rules and membership functions can be approached by using evolutionary algorithms and categorised into four types: learning μ_A with fixed fuzzy rules, as Bonivento et al [84]; learning fuzzy rules with fixed μ_A; learning fuzzy rules and μ_A in stages, first evolving good fuzzy rule sets using fixed μ_A, then tuning μ_A using the derived fuzzy rule sets; learning fuzzy rules and μ_A simultaneously as Hong et al [97], Liska and Melsheimer [98]. Each chromosome in [97] consist of an intermediary fuzzy rule set and its associated μ_A. This allows the GA operators to integrate multiple fuzzy rule sets and their μ_A at the same time. This is the way we have chosen to generate the FLC parameters of the trajectory controller. Further tuning near the optimum can also be achieved by using a conjugate gradient method [98]. GAs have also been applied to FLC design by Ng and Li [99] for searching poorly understood irregular and complex spaces. Forty nine bits have been used to form the rule base structure where a single bit represents each control action.

The proposed framework of our work maintains a population of fuzzy rule sets with their membership functions and uses the evolutionary algorithm to automatically derive the resulting fuzzy knowledge base. A hybrid real valued-binary chromosome is used to define each individual fuzzy system. The real valued parameters are defined as being the $[\triangle a, \triangle b, \triangle c]$ values shown in fig.4.2.2, which lie in range $(0, 1]$. Triangular shapes are used for the membership functions. By varying $\triangle a, \triangle b, \triangle c$, the centre of each μ_A are varying which changes the shapes of the membership functions.

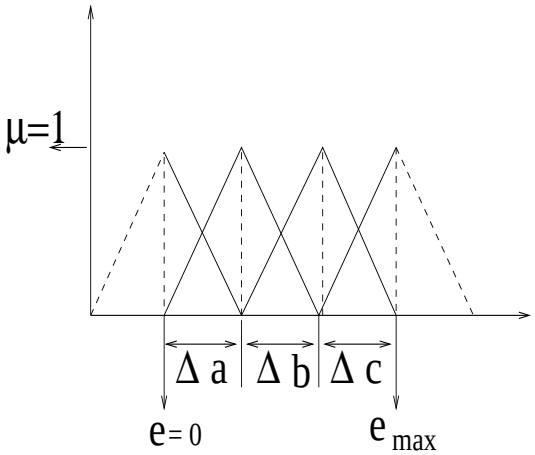


Figure 4.2.2 Membership functions

Five membership functions are used for each input and output, for better closed loop performance. Next the rule base structure is defined. The binary component

shown in fig. 4.4 encodes the set of rules used in the system. The membership functions μ_k of the output O_k for each rule is either on or off (0/1) and corresponds to the form:

$$\text{if } A_i \textit{ is negative small} \text{ AND } B_j \textit{ is zero} \text{ then } O_k \textit{ is negative small} \qquad (4.2)$$

where A_i denotes membership function i of input A (i.e. error), B_j denotes membership function j of input B (i.e. derivative of error), and O_k denotes membership function k of the output O (i.e. control action). In that way the number of μ_i for each output variable involved in each rule is allowed to change dynamically during the GA search. This process allows a full set of rules to be developed for the fuzzy system, but maintains a fixed length chromosome. This leads to a chromosome with 12 real valued genes for two inputs and one output and with 125 binary genes for the rule base. For simplicity fig. 4.4 shows only four membership functions of real-binary coding of the FLC. The length of the chromosome is $N.r = r^2.r = r^3$, where N is the number of rules and r is the number of membership functions. The simulations were carried on Unix workstation with a processor speed of 300 MHz. When using multi-objective optimisation and real-binary coding for the rule base structure, approximately 12 hours were needed for the GA to optimise the control parameters if only one demand was required. If each chromosome is evaluated on three trials (i.e. on three different demands), then 36 hours computational time is needed. By using real-binary coding of the chromosome structure, the increase of number of membership functions leads to significant increase on the size of the rule base structure which is very inefficient computationally speaking. However, produces control surfaces which are more robust on parametric uncertainties. Also when 6 membership functions are used, the length of the chromosome is $r^3 = 216$ bits long. The maximum number of generations used to stop the evolution process is not enough to tune the rules and performance requirements are not met.

In order to decrease processing time, the chromosome structure was modified to real-integer coding, as shown in fig. 4.5. This reduces the length of the chromosome by a factor r, where r is the number of membership functions. In this case each rule can fire only one membership function at the time. Zero is used when a rule is not fired. For evaluations of a chromosome on one trial only (i.e. one set of model coefficients and one required demand), the processing time decreased from 12 down to 5 hours.

The fuzzy system uses product for the member function 'AND'. The 'OR' function is not required as the rules are all expressed as 'AND' terms. The implication method chooses the minimum value and crops the output member functions. The aggregation method chooses the maximum values of the set of member functions. A centroid approach is used to defuzzify the output.

The evolutionary algorithm follows the usual format of ranking, selection, crossover, mutation and evaluation but with the real and binary parts of the chromosomes being processed separately. The number of offsprings that are generated is the same

as the number of parents, hence a total replacement policy is used. To evaluate the performance of each chromosome, a fitness function has been defined such that to assess the closed loop behaviour of the autopilot system. Hence four objectives such as: rise time, steady state error, overshoot and settling time have been used. Three of these objectives: overshoot, rise and settling time have been treated as penalties in order to meet the specified requirements, i.e., if the parameters are within a required range, the penalty is zero and the penalty increases when a threshold is exceeded. A multi-objective approach simplified to a scalar optimisation is considered in this chapter by combining the four closed loop performance criteria in one function $O = O_1 + O_2 + O_3 + O_4$, with O_1 used for steady state error, O_2 for overshoot, O_3 for rise time and O_4 for settling time. However, in Chapter 5, these criteria are treated separately, hence a multi-objective optimisation problem is also considered.

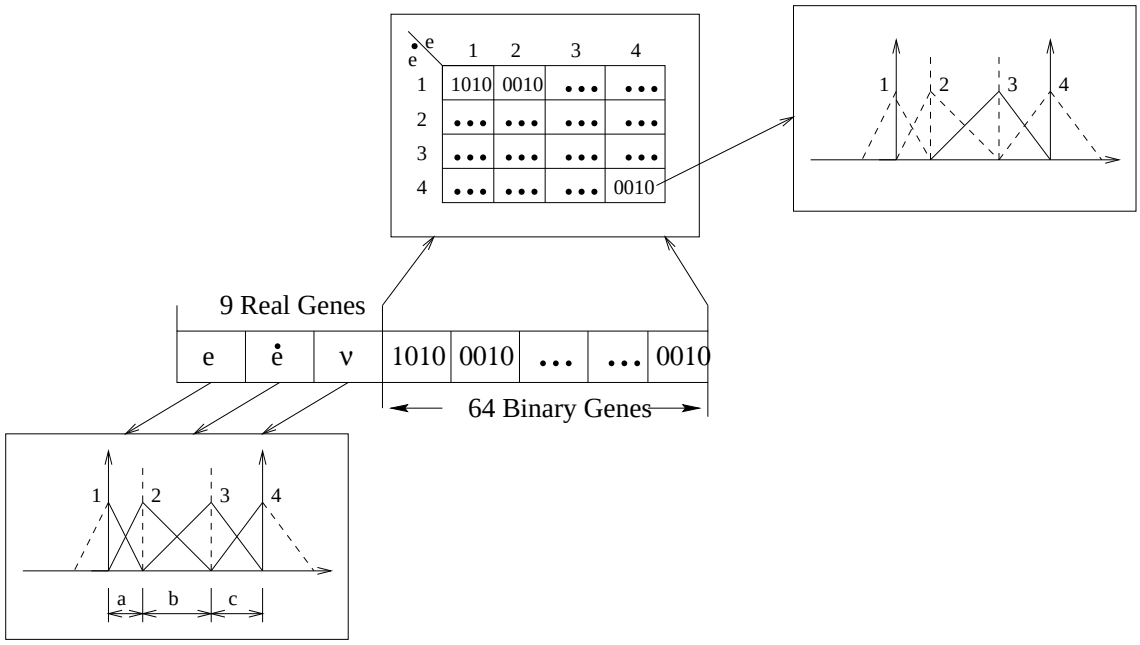


Figure 4.4: FLC chromosome structure with real-binary coding

The efficiency of GAs can be affected by population size. A small population is most likely to result in insufficient coverage of the problem space. Large populations have the advantage of preventing premature convergence to local optima instead of global optima. Large populations can however increase computational time, hence a compromise is usually required, as noticed by Trebi-Ollennu [24]. The choice of population size is a problem dependent. In our work, the evolutionary algorithm was run with a population size of 20 individuals and for 300 generations. These values were suggested by an expert and were found to be sufficient for the problem.

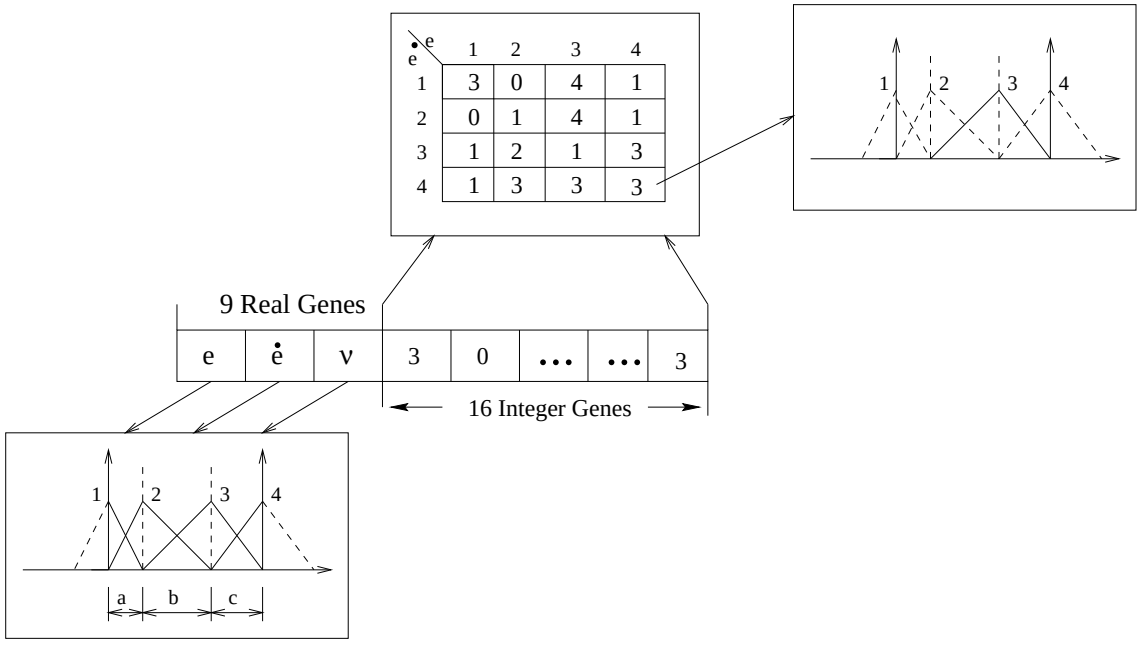


Figure 4.5: FLC chromosome structure with real-integer coding

The fuzzy control parameters were tuned on a large set of randomly generated models of aerodynamic coefficients. These models are simulated within the polynomials of 0° to 45° roll angle. The polynomials are described in tables 2.1 and 2.2 of Chapter 2. Each individual (i.e. an alternative trajectory controller) has been evaluated on five trials, i.e. randomly generated missile models ($\dot{x} = f_i + \triangle f_i + (g_i + \triangle g_i)u$) where $i = 1, \ldots, 5$ and $\triangle f_i, \triangle g_i$ are non-linear functions of the aerodynamic coefficients $(C_{yv}, C_{y\zeta}, X_{cp}, C_{nr})$. In that case five successive evaluations of the same chromosome information returned five sets of objectives. The maximum objective value of the five trials (i.e. steady state error if multi-objective optimisation is considered) has been returned to the GA for evaluation of the chromosome. After all the individuals have been ranked, crossover and mutation operators are processed separately for the real and the binary part of the chromosome. The number of offsprings that are generated are the same as the number of parents, hence a total replacement policy has been used. The results shown in fig. 4.8 of Chapter 4 are obtained for the entire flight range of 0° to 45° roll angle. For the scalar optimisation problem, the algorithm has achieved convergence in approximate three hours computational time on a 300 MHz Unix workstation. During the optimisation 6000 fuzzy logic controllers were evaluated. However when this algorithm is applied to multi-objective optimisation problem, see fig. 4.6, two problems arise.

First, the GA process has taken approximately 12 hours which was computationally inefficient. For each generation, Pareto solutions that are identified are added

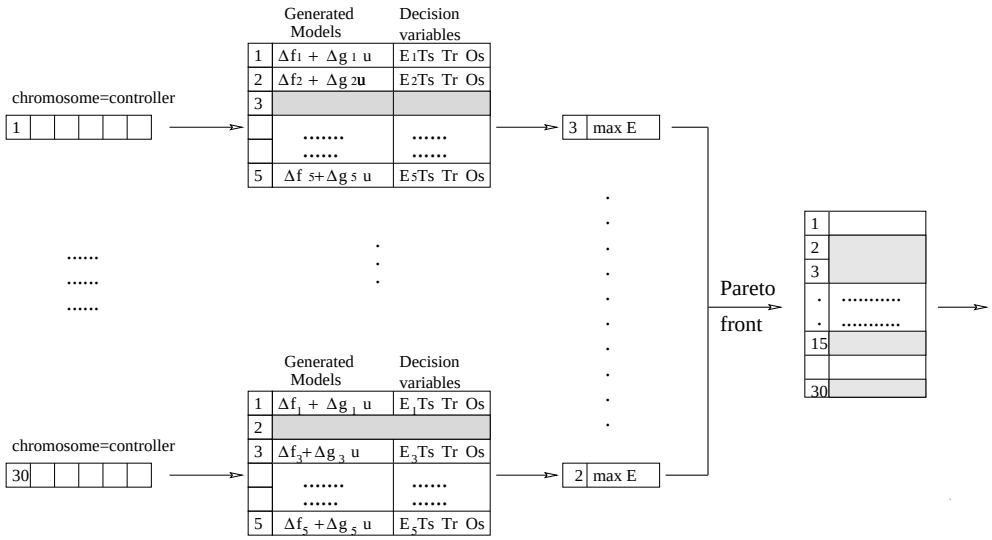


Figure 4.6: GA Multi-objective optimisation

to the existing Pareto solution set. This produces a population with more individuals for the GA to evaluate at each generation. Because 80% of the population were non-dominated solutions, they were on the Pareto front, hence at each next generation, the number of individuals to evaluate was progressively expanding. Also when a multi-objective optimisation was considered, the ranking process has taken longer because each controller was evaluated based on four decision variables (Er_i, Ts_i, Tr_i, OS_i shown in fig. 4.6), which defines the closed loop performance criteria such as steady state error, settling time, rise time and overshoot.

Second, we cannot maintain robustness because the Pareto front was noisy, hence was never consistent. Also we cannot afford to test on many random models to cover the parameter set to sufficiently maintain good solutions from one generation to the next one, because we have not exercised enough models to be statistically consistent. There were many good solutions within each generation which were local but were lost because in the next generation they were tested on a completely new randomly generated models. Since the good solutions were lost it was not possible to breed from them, and hence maintain a robust control surface towards model uncertainties which may arise within such large range of aerodynamic coefficients of 0° to 45° roll angle.

For solving such a noisy problem non-dominated sorting may not be the best way of ranking the individuals. Some other techniques such as MOGA, MOPSEA would provide better performance, as detailed by Hughes [100].

The computation efficiency of the GA algorithm can be improved if a coevolutionary approach is possible to apply. For example Pena-Reyes and Sipper [101] have introduced the fuzzy cooperative coevolution to a real world problem such as breast cancer diagnosis. In their framework the two coevolving species were defined

respectively as membership functions and rules where the fitness of the individuals (membership functions) depend on their ability to colaborate with individuals from the other species (fuzzy rules). Further understandings of their algorithm is under investigation.

Stability issue of the FLC

A stability analysis of the nonlinear fuzzy controller in a closed-loop configuration with the equivalent feedback linearized system is very difficult. The amount of noise coming from the aerodynamic coefficients have caused unpredictable parametric uncertainties since we cannot measure them, neither we do know how many aerodynamic forces or moments will be distributed, hence impossible to analytically analyse. The available analytical methods from nonlinear system theory such as Lyapunov or Popov criterion require an accurate description of the process and the stability proofs can generally only be applied under very special conditions and valid only for simplified models. The resulting controllers are usually conservative because of the conservative nature of the stability criteria. Therefore the analysis of fuzzy controllers in practice are mostly examined by simulation studies.

4.2.3 Results for the scalar optimisation problem

Fig. 4.7 shows the fuzzy surface of the trajectory controller generated by the evolutionary algorithm. This has been developed with randomly generated models exercising the full range of aerodynamic coefficients from 0° to 45° roll angles. The polynomial models for 0° and 45° are defined in tables 2.1 and 2.2, in Section 2.8.1 of Chapter 2.

Model variations at roll angle 45° have caused large steady state error to the acceleration and the velocity responses, hence by using fixed gain trajectory controller tracking performance has not been achieved, as shown in fig. 4.8a. On the other hand the performance of the fuzzy controller has been verified by 200 random trials and the results have been summarised in fig. 4.8b, where the solid line shows the responses for the model at 0° roll angle, and the dashed line is for the model at 45° roll angle.

The desired acceleration a_d is achieved by using the non-linear equation $a_d = f(v)$ which is shown in more details in Chapter 2. Therefore the desired acceleration is a function of the lateral velocity demand. The error dynamics are constructed using the a_d signal and the feedback of the actual states - velocity, rate, and acceleration. The results for lateral acceleration demand 10 m/sec^2 are shown in fig. 4.8. The lateral acceleration is controlled through side-slip velocity and the closed loop performance criteria are defined for the side-slip velocity. As a result, the steady state error on lateral acceleration has not been corrected by the fuzzy trajectory controller when the model at roll angle 45° was used (see the dashed line of fig. 4.8b).

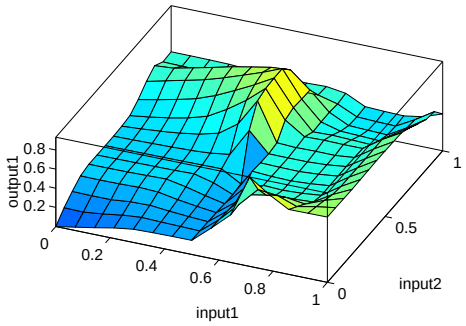
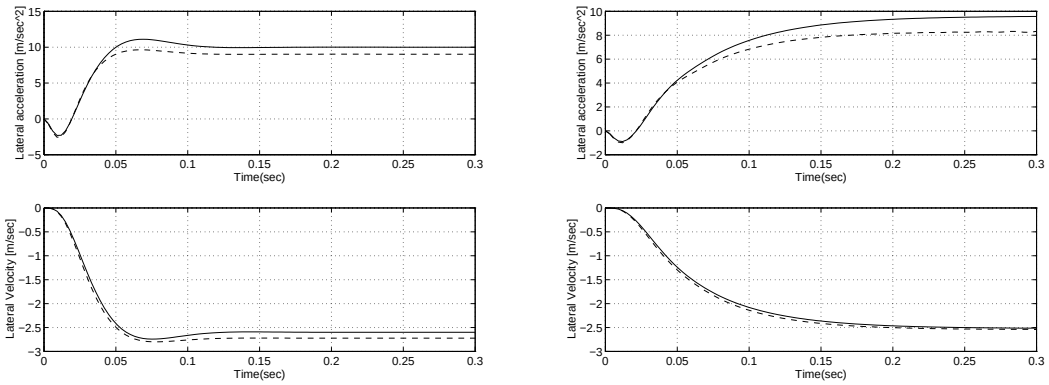


Figure 4.7: Surface of two input, one output fuzzy controller

However, for both models, at roll angle 0° and 45°, the fuzzy trajectory controller has achieved satisfactory tracking performance for side-slip velocity response with almost no steady state error and no overshoot, shown by Blumel et al [102].

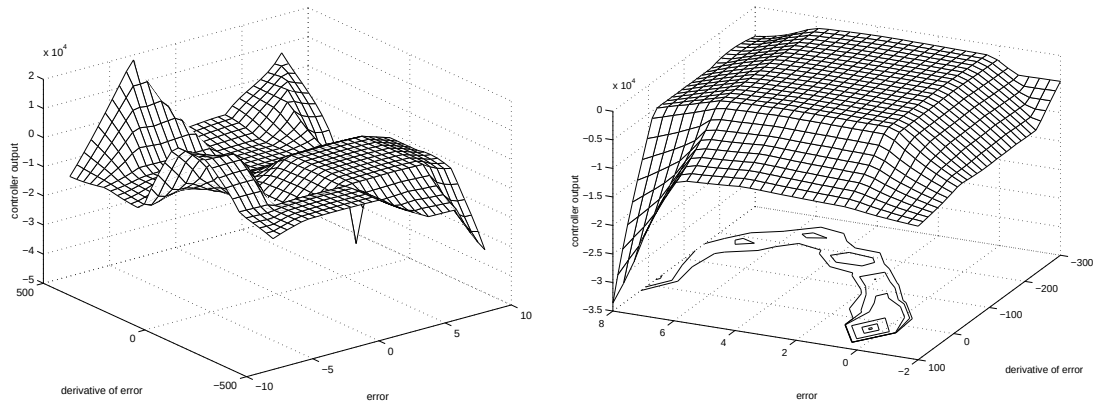


a)Fixed gain trajectory controller b)Fuzzy gain trajectory controller

Figure 4.8: Results for 10 m/sec^2 lateral acceleration demand
solid line - model parameters at 0° roll angle
dashed line - model parameters at 45° roll angle

Conversely to fig. 4.7, the fuzzy control surface shown in fig. 4.9 has been developed with the model exercising the nominal aerodynamic coefficients only. Fig. 4.9a shows the full fuzzy surface of the trajectory controller generated by the evolutionary algorithm. Fig. 4.9b shows the section of the surface that has been used, which is only a small area. These results are obtained by using four membership functions for the fuzzy logic controller, which were not enough to achieve good closed loop performance. The contour of 4.9b shows the usage of the different regions (i.e. the fired rules of the full control surface). It is clear that only a small proportion is

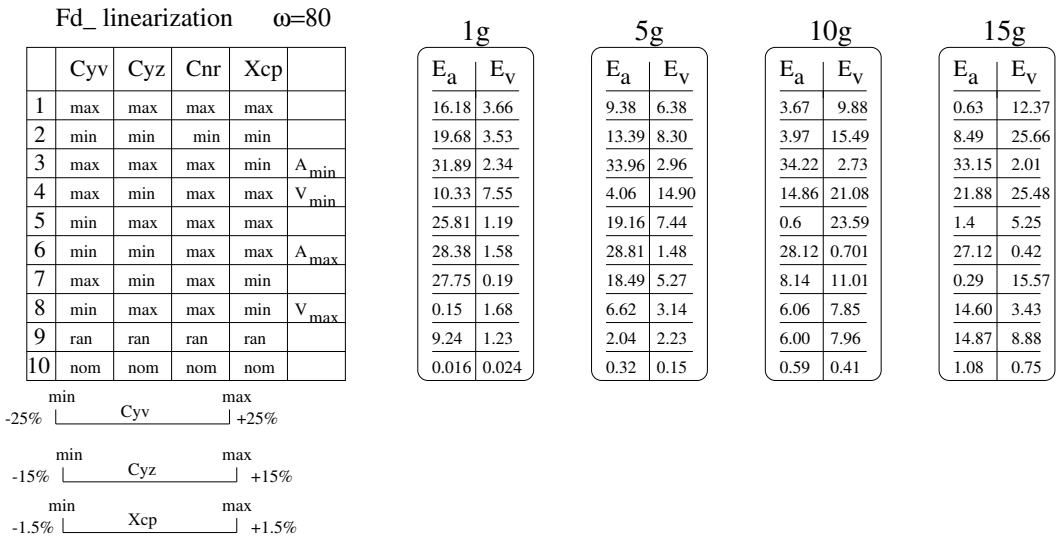
actually used and therefore 'tuned' by the evolutionary algorithm. The most fired rule, 70%, is when both, the error and the derivative of the error, are zero which is the steady state area of the response.



a)FLC surface b)FLC used area

Figure 4.9: Fuzzy control surface found by GA

Initially a set of models was generated by randomly selecting a roll angle between 0° and 45° and calculating the coefficients for the randomly selected value. Variations in the coefficients $C_{yv}, C_{y\zeta}, X_{cp}, C_{nr}$ were also randomly generated at the same time. This produced a large set of models which proved time consuming and so a vertex set of models was determined, as shown in fig. 4.10. The minimum and maximum ranges of the aerodynamic coefficients were chosen to give approximately 10% change in steady state performance for a $1g$ demand and 25% change for a $15g$ demand. This range was judged to be realistic based on the error analysis in the Horton report [69]. When applying higher demands up to $15g$, some of the variations in coefficients at their vertex points cause big steady state errors on side-slip velocities but small on lateral accelerations and some of the variations cause small steady state errors on side-slip velocities but big on lateral accelerations, as shown in fig. 4.10. A decision was made to control the side-slip velocity rather than the acceleration in order to simplify the problem. The extra complexity of acceleration control would slow down the optimisation process.



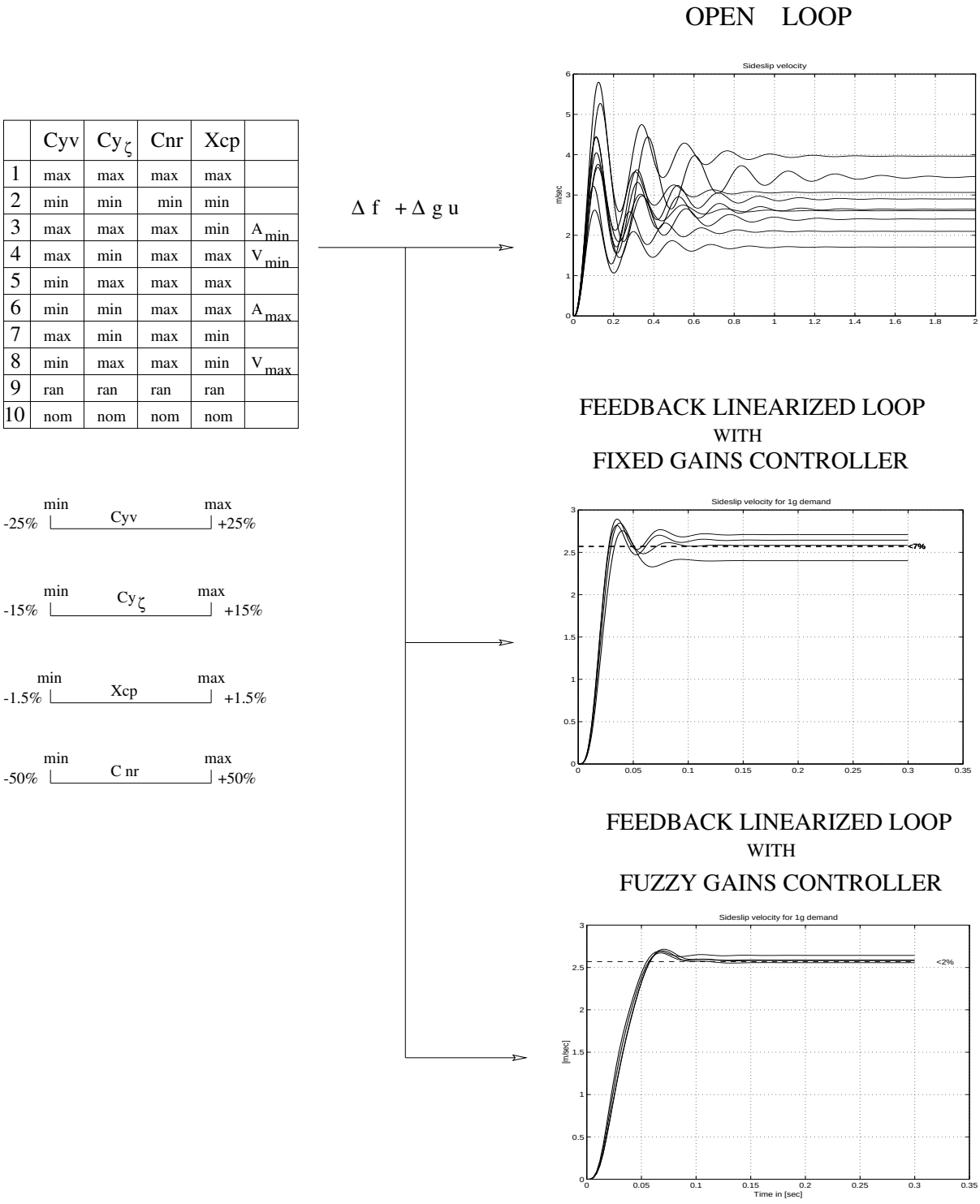
	Cyv	Cyz	Cnr	Xcp	
1	max	max	max	max	
2	min	min	min	min	
3	max	max	max	min	A_{min}
4	max	min	max	max	V_{min}
5	min	max	max	max	
6	min	min	max	max	A_{max}
7	max	min	max	min	
8	min	max	max	min	V_{max}
9	ran	ran	ran	ran	
10	nom	nom	nom	nom	

1g		5g		10g		15g	
E_a	E_v	E_a	E_v	E_a	E_v	E_a	E_v
16.18	3.66	9.38	6.38	3.67	9.88	0.63	12.37
19.68	3.53	13.39	8.30	3.97	15.49	8.49	25.66
31.89	2.34	33.96	2.96	34.22	2.73	33.15	2.01
10.33	7.55	4.06	14.90	14.86	21.08	21.88	25.48
25.81	1.19	19.16	7.44	0.6	23.59	1.4	5.25
28.38	1.58	28.81	1.48	28.12	0.701	27.12	0.42
27.75	0.19	18.49	5.27	8.14	11.01	0.29	15.57
0.15	1.68	6.62	3.14	6.06	7.85	14.60	3.43
9.24	1.23	2.04	2.23	6.00	7.96	14.87	8.88
0.016	0.024	0.32	0.15	0.59	0.41	1.08	0.75

Figure 4.10: Fixed gain closed loop performance

For the worst vertex points models (i.e. models: 3, 4, 6, 8 for which velocity or acceleration responses have their extreme, minimum A_{min}, V_{min} or maximum A_{max}, V_{max}, values), the feedback linearized loop with fixed gain trajectory controller has not been able to provide robust performance, hence tracking is achieved with $\pm 7\%$ on steady state error, as shown in fig. 4.11.

The fuzzy gain trajectory controller has been tuned for nominal aerodynamic coefficients, for a side-slip velocity demand of 2.57 m/sec corresponding to 1g lateral acceleration. Then, the FLC has been tested on parameter variations on the aerodynamic coefficients, $C_{yz}, C_{yv}, X_{cp}, C_{nr}$, for the worst vertex points models. Robust performance within 2% on steady state error has been achieved. For the above mentioned uncertain multi-model airframe dynamics, the fuzzy gain trajectory controller has improved the robustness by 5%, as shown in fig. 4.11.

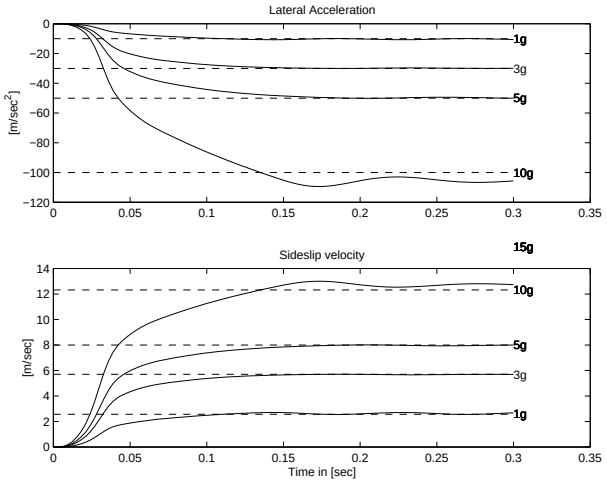


	Cyv	Cyζ	Cnr	Xcp	
1	max	max	max	max	
2	min	min	min	min	
3	max	max	max	min	A_{min}
4	max	min	max	max	V_{min}
5	min	max	max	max	
6	min	min	max	max	A_{max}
7	max	min	max	min	
8	min	max	max	min	V_{max}
9	ran	ran	ran	ran	
10	nom	nom	nom	nom	

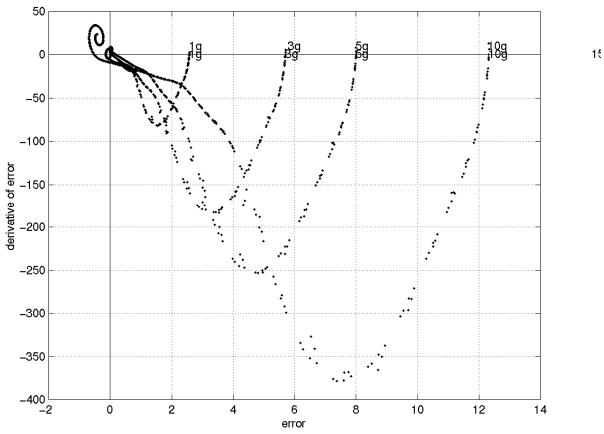
Figure 4.11: The autopilot test on the vertex points models

4.3 Fuzzy Gain Scheduling

The FLC has been tuned for 5g lateral acceleration demand only. The responses are very slow on rise time but good on steady state error, as shown in fig. 4.12. However, for demands higher than 8g, in this example 10g demand is demonstrated, the FLC has not been able to control the velocity to the required demand. This is obvious, because the range of the scaling input-output domain has been changed which has automatically altered the rule based structure. Therefore, a change of the FLC scaling domain is required for any other demand different than the tuned one. An interpolation for a large set of demands (i.e. $1g, \ldots, 15g$) and their FLC's input-output scaling factors have been proposed in the next section.



FLC tunned only for 5g



Phase Portrait

Figure 4.12: Multiple demands

4.3.1 Polynomial fit of the multiple demands for FLCs scaling factors

Input-Output scaling

The membership functions defining the fuzzy values of controller inputs and controller outputs have been defined off-line, on a common normalized domain. This means that the actual physical values of the controller's inputs and outputs are mapped onto the same predetermined normalized domain. This mapping is called normalization and it is done by the so-called normalization factors. Input scaling is the multiplication of a physical, crisp controller input with a normalization factor

so that it is mapped onto the normalized domain. Output scaling is the multiplication of a normalized controller output with a denormalization factor so that it is mapped back onto the physical domain of the controller outputs. Hence, fuzzification, rule firing and defuzzification can be designed independently of the actual physical domains of the controller inputs and controller outputs. The scaling factors, which describe input normalization and output denormalization, play a roll similar to that of the gain coefficients in a conventional controller. In other words, they are of utmost importance with respect to the controller performance and stability related issues, i.e. they are the source of possible instabilities, oscillation problems and deteriorated damping effects as noted by Palm [85]. In Bonissone's chapter of Fantuzzi's book [84], the scaling factors of the FLC have been tuned by GAs. Some time scaling factors are used to fine tune the performance of the system in a similar way to the tuning of a PID controller. In [103] the firing of the rules in a fuzzy controller has been shown by Chen with different values of the scaling factors. The adjustment of the factors is equivalent to the re-construction of the membership functions in a rule-base, and should be done carefully if the linguistic meaning of the rule-base has to be preserved. It is inappropriate to tune the input scaling factors if the rule-base structure is constructed by experts. Fine tuning can be better achieved by tuning the membership functions only, so that the linguistic meaning of the rule-base is preserved.

Bearing in mind those valuable findings we could suggest in future investigations to include the optimisation of the FLC's scaling factors in our work in the presence of uncertainties. However for now, the three scaling factors (error, derivative of error and output) for each required lateral acceleration demand $1g, 2g, \ldots, 15g$ have been determined via simulations based on the results obtained with fixed gain trajectory controller for the nominal model. Then a polynomial fitting has been used to interpolate between the required demands for side-slip velocities in order to obtain the scaling factors of the FLC's inputs and outputs for each demand. As a result, smooth transition of the scaling factors has been achieved when a different demand was required within the above mentioned range.

There are two possible ways of applying polynomial fitting: One, is to use the linear relationship between the required side-slip velocity demands and their scaling factors; Two, is to use the non-linear relationship between the required acceleration demands and the scaling factors for the velocity inputs and outputs of the fuzzy controller.

By applying the linear relationship type of polynomial fitting, it has been found that the first scaling factor for the error is a 1^{st} order polynomial and the polynomial curve is shown in fig. 4.13.

$$SC_{v-er} = f(v_d) = v_d \tag{4.3}$$

where v_d represents the required side-slip velocity demand for the required lateral acceleration respectively. The scaling factor for the derivative of error is of a 3^{rd}

order polynomial:

$$SC_{v-erd} = f(v_d) = p_3 v_d^3 + p_2 v_d^2 + p_1 v_d + p_0 \tag{4.4}$$

and the output scaling factor is a 1^{st} order polynomial.

$$SC_{out} = f(v_d) = q_1 v_d + q_0 \tag{4.5}$$

where $p_0, \ldots, p_3$ and q_0, q_1 are the polynomial fit coefficients for each scaling factors respectively.

The second way is to interpolate between the required lateral acceleration demands and their velocity scaling factors respectively. The non-linear relationship (velocity-acceleration) can be seen in fig. 4.14. Again the fuzzy logic engine has been scaled between $(0, 1)$. In order to achieve the correct scaling factors for the inputs and output of the fuzzy trajectory controller we have interpolated the data between a number of required demands for lateral accelerations (i.e.$1g, 2g, \ldots, 15g$) and their corresponding scaling factors for the error and derivative of error of side-slip velocity respectively.

For the first scaling factor a 4^{th} order polynomial fit has been obtained and the polynomial curve is shown in fig. 4.14.

$$SC_{v-er} = f(a_d) = b_4 a_d^4 + b_3 a_d^3 + b_2 a_d^2 + b_1 a_d + b_0 \tag{4.6}$$

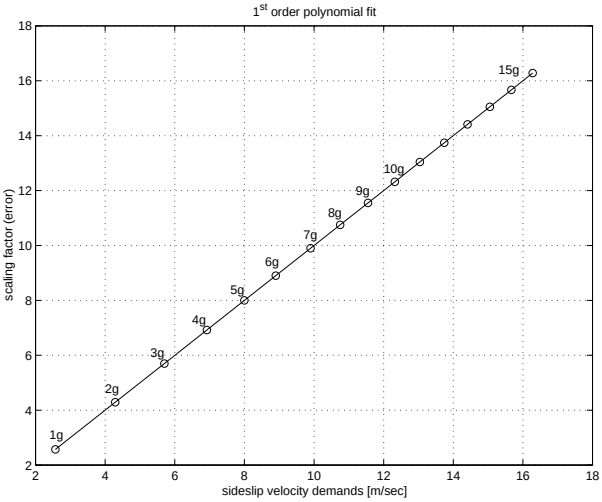
where a_d represents the required lateral acceleration demand and SC_{v-er} the corresponding scaling factor respectively. The scaling factor for the derivative of error is a 3^{rd} order polynomial.

$$SC_{v-erd} = f(a_d) = c_3 a_d^3 + c_2 a_d^2 + c_1 a_d + c_0 \tag{4.7}$$

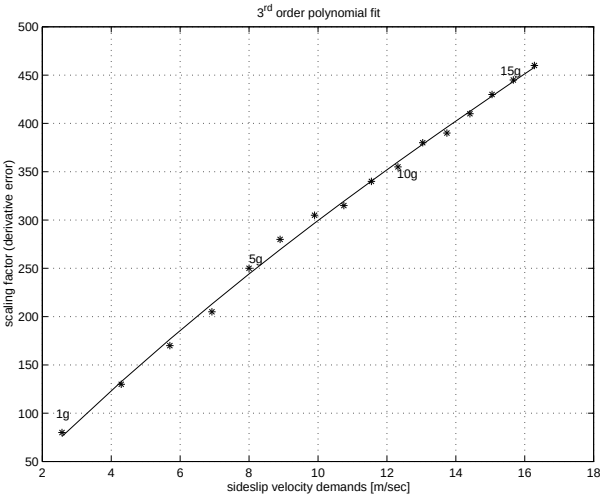
and the scaling factor for the output is a 4^{th} order polynomial.

$$SC_{out} = f(a_d) = d_4 a_d^4 + d_3 a_d^3 + d_2 a_d^2 + d_1 a_d + d_0 \tag{4.8}$$

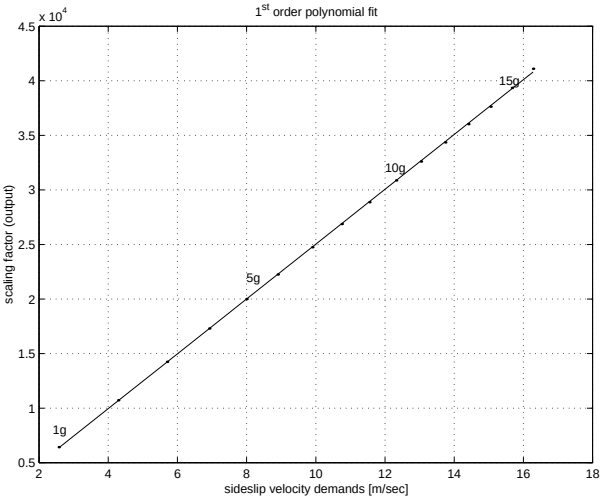
where $b_0, \ldots, b_4$ and $c_0, \ldots, c_3$ and $d_0, \ldots, d_4$ are the polynomial fit coefficients for each scaling factors respectively.



$$SC_{v-er} = f(v_d) = v_d$$

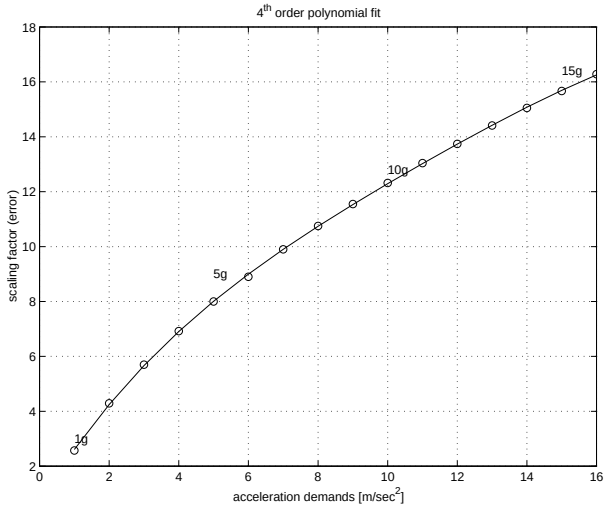


$$SC_{v-erd} = f(v_d) = p_3 v_d^3 + p_2 v_d^2 + p_1 v_d + p_0$$

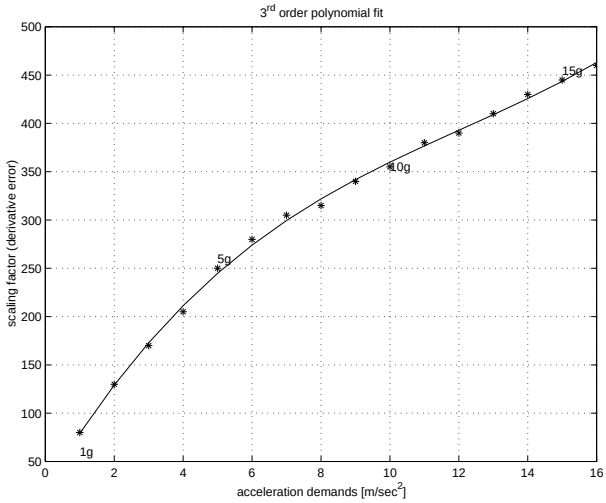


$$SC_{out} = f(v_d) = q_1 v_d + q_0$$

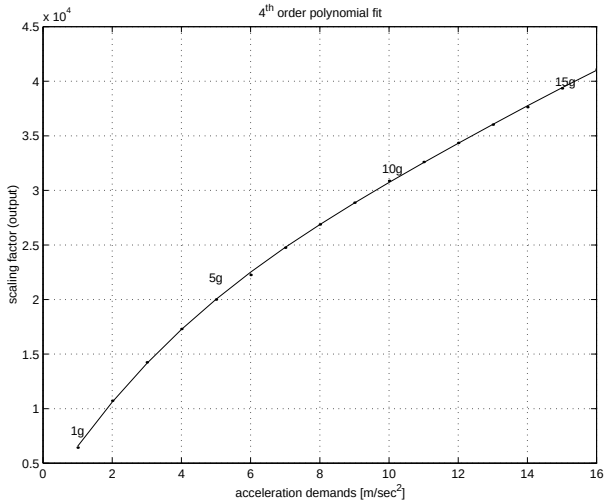
Figure 4.13: Polynomial fit curves for multiple demands v_d



$$SC_{v-er} = f(a_d) = b_4 a_d^4 + b_3 a_d^3 + b_2 a_d^2 + b_1 a_d + b_0$$



$$SC_{v-erd} = f(a_d) = c_3 a_d^3 + c_2 a_d^2 + c_1 a_d + c_0$$



$$SC_{out} = f(a_d) = d_4 a_d^4 + d_3 a_d^3 + d_2 a_d^2 + d_1 a_d + d_0$$

Figure 4.14: Polynomial fit curves for multiple demands a_d

4.3.2 Results for a large range of lateral acceleration demands

In a situation when a change of a demand is required, smooth transition and gradual interpolation between the fuzzy control surfaces has been automatically achieved. The FLC has been simultaneously tuned for two different demands, in this case $5g$ and $15g$. The resulting rule base structure and membership function's shapes have been achieved by the scaling factors determined through the polynomial fitting. The linear type relationship to determine the polynomials for each scaling factor has been used. The more points we use, the better fit we get. The purpose of such a tuning process is to improve the system performance with the intention to maintain the linguistic meaning of the fuzzy controller, which has been validated for each required demand.

The FLC control surface is shown on the left side of fig. 4.15. It has been tested for a variety of required demands in this case $1g, \ldots, 15g$. It can be seen that for each demand, the FLC scaling factors have changed automatically and desired tracking has been achieved. However the linguistic meaning of the rule base structure has been altered and variations in some rules can be seen. The abscissa of the right column figures have presented the phase portrait for the side-slip velocity errors and their derivatives. Indirect lateral acceleration control has also been achieved by requiring different side-slip velocity demands (for example $1g, 5g, 10g, 15g$), as shown in fig. 4.16.

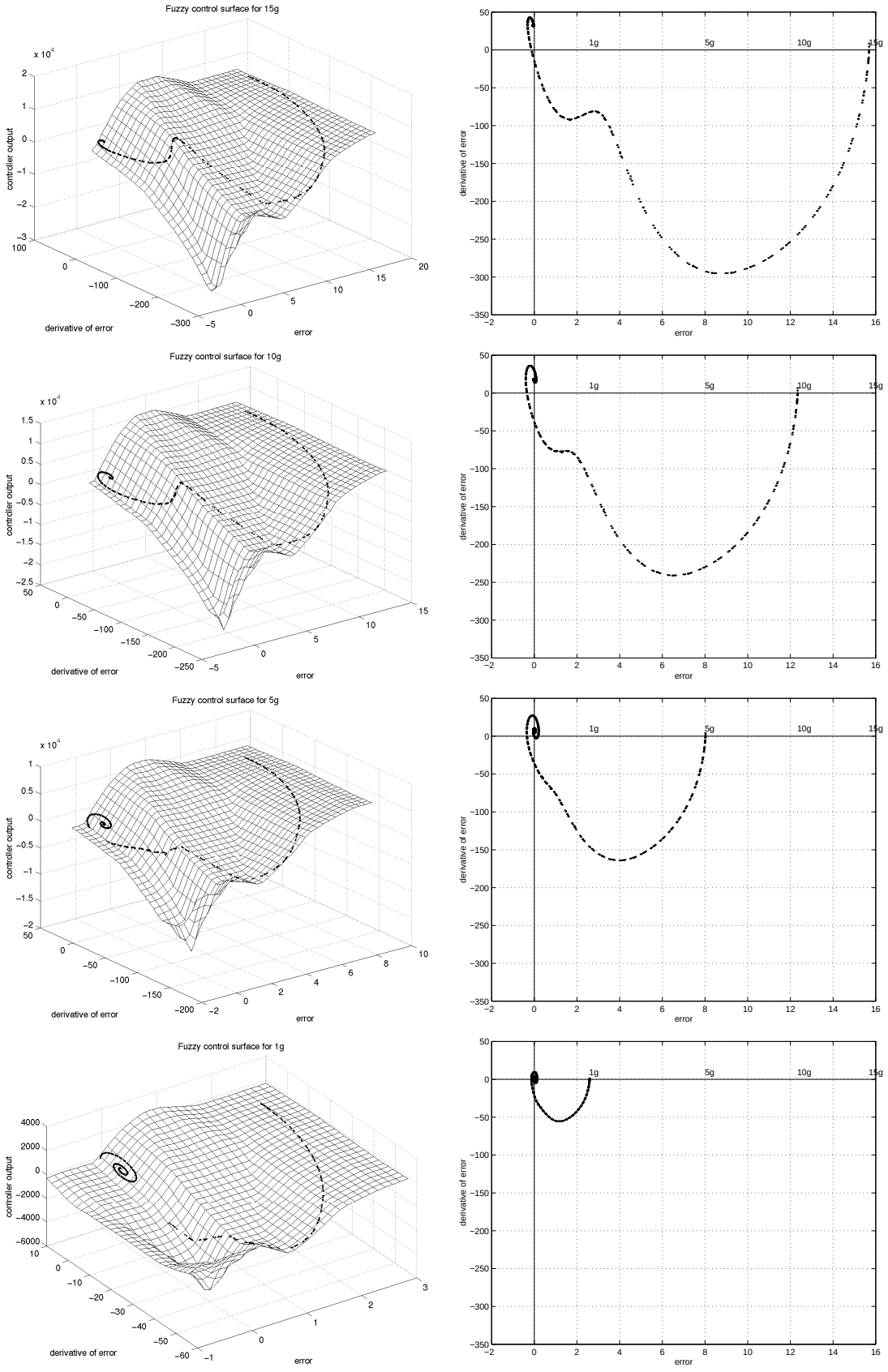
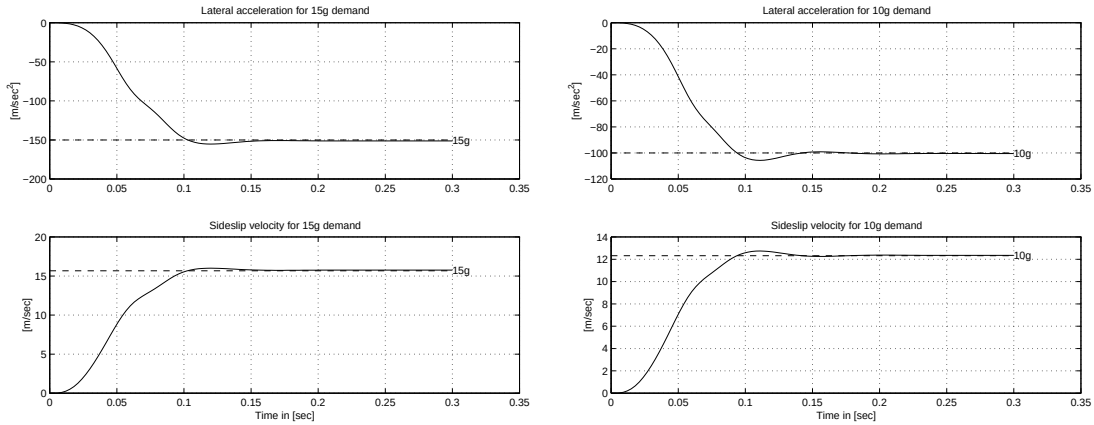
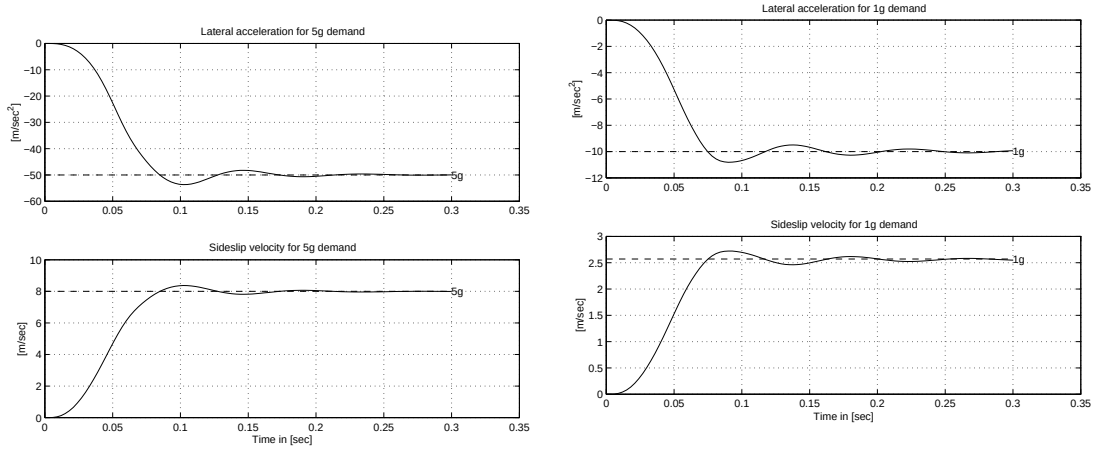


Figure 4.15: The best tuned FLC for 15g and 5g simultaneously



a)15g demand b)10g demand



c)5g demand d)1g demand

Figure 4.16: Responses for different demands

Fixed Gain Trajectory Controller

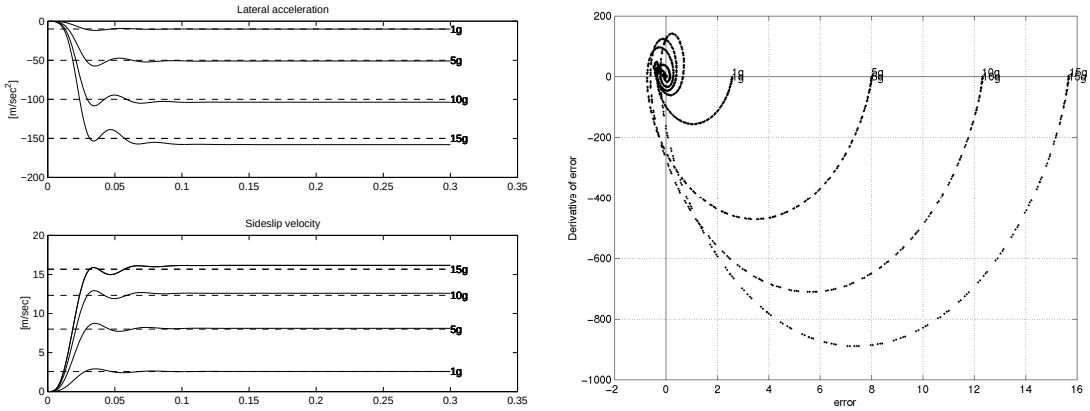


Figure 4.17: Fixed gain trajectory controller for multiple demands 15g, 10g, 5g, 1g

In order to compare the performance of the fixed gain and the fuzzy gain trajectory controllers, we have demonstrated in fig. 4.17 and fig. 4.18 the side-slip velocity and lateral acceleration responses for a set of different demands ($1g$,$5g$,$10g$,$15g$). The fuzzy trajectory controller has been found to be superior to the fixed gain one. The quality of the responses has been improoved on steady state error and overshoot. The feedback linearizable system has been modified by neglecting the g term in the system when feeding back the acceleration. This rendered a significant steady state error which has not been corrected by the fixed gain controller especially when higher demands were required.

Fuzzy Logic Trajectory Controller

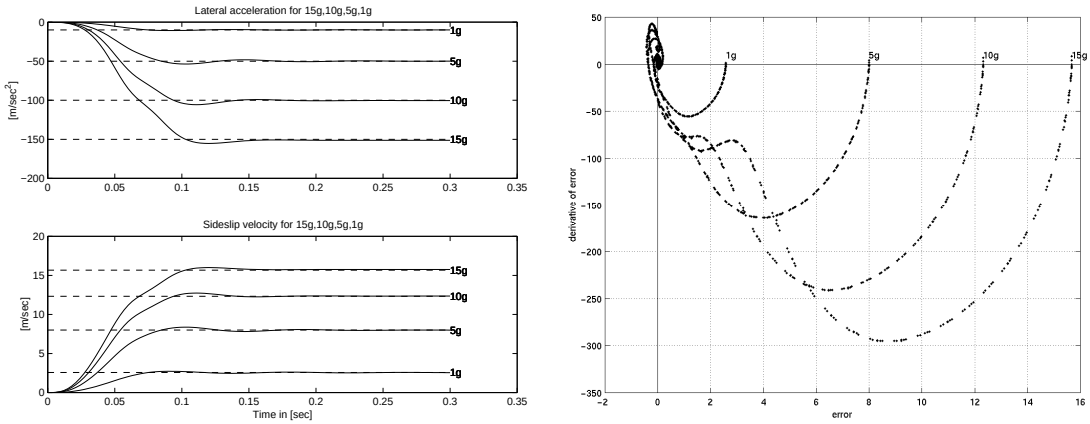


Figure 4.18: Fuzzy Trajectory Controller for multiple demands 15g, 10g, 5g, 1g

4.4 Conclusions

A fuzzy nonlinear trajectory controller has been proposed within the outer loop to improve the robustness of the Feedback Linearization with respect to parametric uncertainties caused by changes of the aerodynamic coefficients due to different flight conditions. The autopilot design has been found to be robust on mass changes.

The fuzzy logic-feedback linearized control design has been found to be more effective for improving the transient and steady state performances than the fixed gain-feedback linearized one. The ability of the FLC to improve the closed loop performance while managing uncertainties has been shown.

The fuzzy inferencing procedure can provide the means of systematically synthesizing various fuzzy rules to produce decision actions so that a complex non-linear missile system can be controlled. Fuzzy reasoning builds the understanding of *imprecision* into the process, hence provide the ability to control the system in an *uncertain* environment and derive smooth control action for uncertain system behaviour which is one of the most important characteristics of an intelligent control system.

The FLC is of nonlinear nature, hence can be designed to capture the nonlinear dynamics of a system. By increasing the number of the membership functions, the FLC can achieve better closed loop performance, but for the loss of computational time, because the number of rules increase significantly. When a large number of rules must be captured and stored, the FLC implementation can be expensive (computationally speaking).

The trial and error mechanism for tuning FLC parameters has been replaced by evolutionary algorithm optimisation using GAs for better adaptation and robustness. The hybrid control strategy has been validated via extensive computer simulations.

A fuzzy logic scheduled controller for missile autopilot design has been examined. The fuzzy logic input output scaling factors have been determined by using polynomial fitting for a large range (1g up to 15g) of multiple acceleration demands.

This chapter has shown that fuzzy controllers can be used for solving engineering problems allowing the designer to investigate the properties of the system. These systems are reliable over wide variations in plant dynamics and offer control designers a more elegant solution for such a complex autopilot design system. Also hardware chip and board level solutions are available, as stated by Mcneill and Freiberger [86], hence fuzzy control systems can be prototyped.

The next chapter will address the optimisation of the fuzzy trajectory control parameters from multi-objective point a view.

Chapter 5

Multi-objective optimisation using GA

Many problems involve simultaneous optimisation of multiple objectives. In principle, multi-objective optimisation is very different from single objective optimisation. In single objective optimisation, one attempts to obtain the best design or decision, which is usually a global minimum or global maximum, whichever is the accepted definition of optimum. In the case of multiple objectives one solution that is best with respect to all objectives may not exist. These solutions are known as non-dominated. Since none of the solutions in the non-dominated set is absolutely better than any other, any one can be an acceptable solution. The choice of one solution over the others requires problem knowledge and problem related factors.

One of the main requirements for an autopilot design is to yield a response as fast as possible with the minimum of overshoot so that any command is attained quickly and is of the required magnitude. For low g demands only a slight overshoot of short duration is usually acceptable, since overshoot can compensate for loss of acceleration during the initial transient. For high g demands, overshoot is usually unacceptable since the airframe structural load limit may be exceeded as stated by Lin [104]. In order for the autopilot to yield an accurate and fast response it is very important to assess the quality of lateral acceleration response, which is quantified in terms of rise time, settling time, maximum percentage overshoot with almost no steady state error. This means that while tuning the trajectory control parameters, the optimisation process should consider these four criteria simultaneously, hence the single optimisation problem has become one of multi-objective optimisation, which provides the designer with multiple solutions. Then, question can be asked: Is the engineer more interested in fast rise time responses or is a slow rise time with no overshoot satisfactory? The four criteria are conflicting in nature and a compromise solution must be used. It is interesting to mentioned here that in most multi-objective optimisation cases it is not clear what kind of preferences should be specified for each objective, whereas in this particular case the missile engineer is interested in achieving closed loop performance values within specified ranges in order that the missile can respond as fast as possible to guidance commands under

all flight conditions. The determination of these ranges has been proposed by the author in two different ways:

1. using reference points (ideal, maximum and minimum values for each objective), as discussed in Section 5.4

2. handling the objectives as penalties based on fuzzy logic membership functions, as discussed in Section 5.5.

Both ways incorporate preference information into the genetic algorithm optimisation process to direct the search towards feasible areas which satisfy specific values of the objectives. A Pareto based approach using non-dominated sorting is used to produce optimal solutions.

The aim of this chapter is to produce multiple solutions (alternative fuzzy trajectory controllers) which allow the designer to select the best and to investigate the properties of the system.

5.1 Multi-objective optimisation problem

Multi-objective optimisation (also called multi-criteria optimisation or vector optimisation) has been defined by Ocyczka [105] as:

the problem of finding a vector of decision variables which satisfies constraints and optimises a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term 'optimise' means finding such a solution which would give the values of all the objective functions acceptable to the designer.

It can be stated as follows: Find the vector

$$x^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$$

which will satisfy the m inequality constraints:

$$g_i(x) \geq 0, i = 1, 2, \ldots, m$$

the p equality constraints

$$h_i(x) = 0, i = 1, 2, \ldots, p$$

and optimises the vector function

$$f(x) = [f_1(x), f_2(x), \ldots, f_k(x)]^T$$

where $x = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables. The problem is to determine the particular set of decision variables which yields the optimum values

of all the objective functions. The constraints define the *feasible region* F and any point x in F defines a *feasible solution*. The vector function $f(x)$ is a function which maps the set F in the set X which represents all possible values of the objective functions. The k components of the vector $f(x)$ represent the non-commensurable criteria[1] which must be considered. The constraints $g_i(x)$ and $h_i(x)$ represent the restriction imposed on the decision variables. The vector x^* denote the optimal solutions (normally there will be more than one).

The meaning of *optimum* is not well defined in this context, since it is very rare to get x^* such that for all $i = 1, 2, \ldots, k$

$$\wedge_{x \in X}(f_i(x^*) \leq f_i(x))$$

In that case, x^* would be a desirable solution. However, normally the case in which all the $f_i(x)$ have a minimum in F at a common point x^* does not occur in practice and in that case certain criteria need to be established to determine what would be considered as an 'optimal' solution.

5.1.1 Ideal vector

The vector f^* is an ideal vector (the demanded level vector) in the objective space which contains reference values for each criteria. The values f_j^*, $j \in 1, \ldots, m$ can be specified by the decision maker or can be determined by solving each single optimisation problem separately:

$$f^* = [minf_1(x), minf_2(x), \ldots, minf_m(x)]$$

Generally the vector f^* is not attainable.

5.1.2 Pareto Optimum

The concept of *Pareto optimum* was formulated by the economist Vilfredo Pareto in the 19^{th} century. A point $x^* \in F$ is *Pareto optimal* if for every $x \in F$ either

$$\wedge_{i \in I}(f_i(x) = f_i(x^*))$$

or there is at least one $i \in I$ such that

$$f_i(x) > f_i(x^*)$$

This definition says that x^* is Pareto optimal if there exists no feasible vector x which would decrease some criterion without causing a simultaneous increase in at least one other criterion. The Pareto optimum almost always gives not a single solution, but rather a set of solutions called non-inferior or non-dominated solutions.

[1] Non-commensurable means that the values of the objective functions are expressed in different units

5.1.3 Pareto front

The minima in the Pareto sense are in the locus of the tangent points of the objective functions. The region of points is called *Pareto front*. It is not easy to find an analytical expression of the line or surface that contains these points, and the normal procedure is to compute the points F^k and their corresponding $f(F^k)$. A point $x^* \in F$ is a weakly non-dominated solution if there is no $x \in F$ such that $f_i(x) < f_i(x^*)$, for $i = 1, \ldots, n$. A point $x^* \in F$ is a strongly non-dominated solution if there is no $x \in F$ such that $f_i(x) \leq f_i(x^*)$ for $i = 1, \ldots, n$, and for at least one value of i, $f_i(x) < f_i(x^*)$. Thus, if x^* is strongly non-dominated, it is also weakly non-dominated, but the converse is not necessarily true.

5.2 Review of multi-objective GAs-based approaches

The motivation to use an evolutionary technique such as Genetic Algorithms (GAs) for multi-objective optimisation problems is because GAs are very useful for finding global solutions when applied to multi-modal noisy search spaces. GAs work with a population of points as it is natural to use them to capture a number of solutions simultaneously and are powerful in their search for improvement. Hwang et al [106] have provided an extensive survey of multiple objective decision making approaches. Fonseca and Fleming [107], followed later on by Coello [108] has detailed many of them in the context of genetic algorithms optimisation. Only a brief discussion will be given here in order to give some idea of the many possible ways of tackling a multi-objective optimisation problem.

Coello [108] has distinguished between three different groups: aggregating ('naive'), non - aggregating (none Pareto based) and Pareto based approaches. The first group (weighted approach, goal programming, goal attainment and constraint method) work on the principles of combining all the objectives into a single one. There are some obvious problems such as providing some accurate scalar information on the range of the objectives to avoid having one of them dominate the others. This implies that the behaviour of each of the objective functions should be known but in real world applications this could be a very expensive process (computationally speaking) and is not always possible. However, this is the simplest approach and one of the most efficient procedures, because no further interaction with the decision maker is required. Also, these approaches are applicable in cases when it is necessary to assign more importance to certain objectives by using **weights**. Most researchers like Begg et al [109], Gen et al [110] use a simple linear combination of the objectives and then generate the trade-off surface by varying the weights. The approach has the disadvantage of missing the concave portions of the trade-off surface as detailed by Ritzel et al [111]. In addition, if the decision maker (DM) has to assign targets or goals (**goal programming**) that have to be achieved for each objective, the objective function will try to minimize the absolute deviations

from the targets to the objectives. This idea started from Zeleny [112] is called 'the ideal displacement' and later on was used by Weistroffer and Narula [113] as the reference point approach, which has been further expanded to the optimistic or pessimistic approach by Weistroffer [114]. In our research we have used this idea to specify goals for each objective. The difference is that it is combined with non-dominated GA sorting, which is a Pareto based approach, to provide the GAs with preferable directions in which to search for desirable solutions. Details are given in Section 5.4. Some applications of goal programming combined with GA are published in the literature by Sandgren [115], Wienke et al [116]. On the other hand Wilson and Macleod [117] elicited some problems associated with **the goal attainment method**. The main weakness is that, if there are two candidate solutions which are the same in one objective function value, but different in the other, they will still have the same goal attainment value for their two objectives, which means for the GAs that none will be better than the others. Another technique is **the constraint method**, which is based on minimization of one (the most preferred) objective function and considering the other objectives as constraints bounded by some allowable levels ξ_i. Hence, a single objective minimization is carried out for the most relevant objective function subject to additional constraints on the other objective functions. The constrained levels are then altered to generate the entire Pareto optima set. This approach was suggested by Ritzel et al [111] as a simple and naive way of solving multiple optimisation problems using genetic algorithms. The idea is to code the GA in such a way that all the objectives except for one are kept constant(constrained to a single value) and the remaining objective is the fitness function for the GA. Thus, through a process of running the GA numerous times with different values of the constrained objectives, a trade-off surface can be developed. The obvious drawback is that it is time consuming and also tends to find weakly non-dominated solutions.

The other big group within multi-objective optimisation is the non-aggregating approaches that are not Pareto based. The vector evaluated genetic algorithm - (**VEGA**) differs from simple genetic algorithm only in the way in which selection is performed. At each generation, a number of sub-populations is generated by performing proportional selection according to each objective function in turn. For a problem with k objectives, k sub-populations of size $\frac{N}{k}$ are generated, assuming a total population size of N. These sub-populations are shuffled together to obtain a new population of size N on which the GA applies the crossover and mutation operator in the usual way. Shaffer [118] found that the solutions are non-dominated in a local sense, because their non-dominance is limited to the current population. An individual who is not dominated in the generation, may become dominated by an individual who emerges in a later generation. This approach is easy to implement but Richardson et al [119] notes that the shuffling and merging of all the sub-populations corresponds to averaging the fitness components associated with each of the objectives. The resulting expected fitness corresponds to a linear combination of the objectives, where the weights are dependent on the distribution of the population at each generation. Certain points in concave regions will not be found

through this optimisation procedure in which a linear combination of objectives is used, regardless of the set of weights. In the **Lexicographic ordering** technique, the objectives are ranked in order of importance by the designer. The optimum solution is then obtained by minimizing the objective functions, starting with the most important one and proceeding according to the assigned order of importance of the objectives. The use of tournament selection makes an important difference with respect to VEGA, because the pair-wise comparisons of tournament selection will make scaling information negligible (Fonseca and Fleming [107]), which means that this approach may be able to see as convex a concave trade-off surface. The idea of **Weighted Min-Max approach** has been taken from game theory which deals with solving conflicting situations. Knowing the extremes, obtained by solving the optimisation problem for each criterion separately, the desirable solution is the one which gives the smallest values of the relative increments of all the objective functions. Hajela and Lin [120] included the weights for each objective in the chromosome and promoted their diversity in the population through fitness sharing, hence providing the ability to simultaneously generate a family of Pareto-optimal designs corresponding to different weighting coefficients in a single run of the GA. A single number used in the chromosomic string represented not the weight itself but a combination of weights and the sharing was applied to those combinations. Also, a mating restriction mechanism was imposed to avoid members within a radius σ_{mat} to cross, hence keeping only feasible solutions at all generations. This approach may create a very high selection pressure for certain combinations of weights. However, the use of a sharing factor may avoid premature convergence, but it is difficult to design. On the other hand the use of mating restrictions and feasibility checks during the entire evolution process is a constraint-handling approach and may not work in concave search surfaces.

Finally, Pareto based approaches are reviewed. The basic idea is to find the set of strings in the population that are Pareto non-dominated by the rest of the population. These strings are then assigned the highest rank and eliminated from further contention. Another set of Pareto non-dominated strings are determined from the remaining population and are assigned the next highest rank. This process continues until the population is suitably ranked. A niching mechanism such as sharing, as given by Goldberg and Richardson [121], can allow the GA to maintain individuals all along the non-dominated frontier. The performance of Pareto ranking technique is highly dependent on an appropriate selection of σ_{share} value. The main strength is that it is less susceptible to the shape or continuity of the Pareto front. Goldberg [94] first suggested the use of non-domination ranking and selection to move a population toward the Pareto front. **MOGA** has been described by Fonseca in Zalzala and Fleming book [122]. The rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. All non-dominated individuals are assigned rank 1, while dominated ones are penalized according to the population density of the corresponding region of the trade-off surface. To avoid premature convergence a niche-formation method is used to distribute the population over the Pareto optimal region, but instead of performing

sharing on the parameter values as Deb [123], they have used sharing on the objective function values, which means that two different vectors with the same objective function values can not exist simultaneously in the population under this scheme. This could be undesirable because the user may be interested in this kind of solution. Fonseca and Fleming [107] have proposed the use of a utility function combined with MOGA to produce a method for the progressive articulation of preferences. In this case it is possible to evolve only a certain region of the trade-off surface by combining Pareto dominance with partial preference information in the form of a goal vector. The idea is to have a feedback loop between the DM and the GA so that certain solutions (from the Pareto set) are given more preference than others. **Non-dominated sorting** (NSGA) is based on several layers of classification of the individuals, as given by Srinivas and Deb [124]. Before the selection is performed the population is ranked on the basis of domination. All non-dominated individuals are classified into one category (with a dummy fitness value), which is proportional to the population size to provide an equal reproductive potential for these individuals. To maintain the diversity of the population, these classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of non-dominated individuals is considered. The process continues until all individuals in the population are classified. A stochastic remainder proportionate selection is used. Since individuals in the first front have the maximum fitness value, they always get more copies than the rest of the population. This allows to search for non-dominated regions and results in quick convergence of the population toward such regions. Sharing, on its part, helps to distribute it over this region. The efficiency of NSGA lies in the way in which multiple objectives are reduced to a dummy fitness function using a non-dominated sorting procedure. With their approach any number of objectives can be solved and both maximization and minimization problems can be handled. In this case, sharing is done on the parameter values instead of the objective values (like MOGA does), to ensure better distribution of individuals, and to let multiple equivalent solutions exist. **Niched Pareto GA** is a tournament selection scheme based on Pareto dominance. Instead of limiting the comparison to two individuals, a number of other individuals in the population is used to help determine dominance. When both competitors are either dominated or non-dominated, the result of the tournament is decided through fitness sharing, as given by Goldberg and Richardson [121]. This approach does not apply Pareto selection to the entire population, but only to a segment of it at each run, hence the technique is very fast and produces good non-dominated fronts that can be kept for a large number of generations.

Concluding remarks

In summary, if it is necessary to assign more importance to certain objectives, an aggregating approach is the one to use, as it can change the importance of the objectives easily, in contrast with the ranking techniques (Pareto based approaches). However, the closed loop performance criteria of autopilot responses are all important, hence it is not appropriate to apply aggregating techniques. Also, non-Pareto

evolutionary algorithms are often sensitive to the non-convexity of Pareto optimal sets, which is not the case for Pareto based algorithms, as given by Fonseca and Fleming [125]. However, there is no such thing as the best method of applying Pareto-optimality, although the use of Pareto-based ranking seems to be gaining some popularity in new research. These methods allow information from the whole of the population to be incorporated into the search capabilities of the GA. Zitzler and Thiele [126] have compared four Pareto based approaches quantitatively, among which the non-dominated sorting genetic algorithm have shown best performance. It has been chosen to populate the Pareto front of optimal solutions. The sharing mechanism is done in the parameters values (the chromosome structure) instead of the objective values (as in MOGA). The former ensures better distribution of individuals within the non-dominated front. In addition, preferable ranges for each closed loop performance criteria are required by the designer engineer before the start of the optimisation procedure. That is why the Reference Point approach has been suggested in combination with the non-dominated sorting approach, in order to incorporate preference information into the GA to guide the search to the particular Pareto region that is of interest to the DM.

5.3 GA strategy for finding non-dominated solutions

The evolutionary algorithm follows the usual format of ranking, selection, crossover, mutation and evaluation, but with the real (membership functions) and discrete (rule-base structure) parts of the chromosomes being processed separately. Then, a multi-objective approach is used to identify multiple solutions. The mechanism of the non-dominated sorting Pareto based approach has already been explained in the previous Section 5.2. The non-dominated ranking is detailed by Deb [123]. All solutions in the population are compared for domination on all objectives and the ones that are not marked 'dominated' are non-dominated solutions. All these non-dominated solutions are assumed to constitute the first non-dominated front in the population. These solutions are temporarily ignored from the population and the procedure is applied again. The resulting non-dominated solutions are assumed to constitute the second non-dominated front. This procedure is continued until all population members are assigned a front. The ranking operation helps to prevent premature convergence of the genetic algorithm.

Since all solutions in a particular non-dominated front are equally important, all are assigned the same *fitness* value. We begin with solutions of the first non-dominated front. A *dummy* fitness value (equal to 1) is assigned to each non-dominated solution of the first front. However, in order to maintain diversity among solutions, a sharing mechanism is applied to these individuals, reducing their assigned value if they have near neighbours (on a chromosome level). The sharing process ensures that a spread of solutions is obtained across the Pareto front. The minimum value assigned to the first front solutions is identified and then reduced by 1%. This re-

duced value is then used as the dummy value for the second front solutions and the sharing procedure is performed among the solutions of the second non-domination front. This process is continued until all population members are assigned a shared fitness value. The conventional ranking and selection processes are then applied as normal to the objective obtained by the non-dominated ranking and the sharing operation. After all solutions are assigned a fitness, a *selection operator* based on the stochastic universal sampling principle is used to select good individuals from the population for breeding, where a solution is selected as a parent in proportion to its fitness value. With such an operator, solutions of the first non-dominated front have a higher probability of being a parent than solutions of other fronts. This process allows the algorithm to search for non-dominated regions, which will finally lead to the Pareto-optimal front. This results in quick convergence of the population toward non-dominated regions and the sharing procedure helps to distribute it over this region. Thus, the selection operator helps to emphasize better solutions in the population and reproduce them, but does not help to create new solutions, a matter which is performed by *Crossover and Mutation operators*. Before producing new individuals, the concept of generation gap was employed. The generation gap (GGAP) represents the percentage of the population to be replaced during each generation. For each new generation (N*GGAP) individuals of the current population are selected to be replaced in the next generation, where N is the number of individuals in the population. In this work a generation gap of 20% is used. Crossover utilizes probabilistic decisions to exchange systematic information among two randomly selected individuals from the mating pool to produce new individuals. The process involves picking uniformly, at random, a crossover point along the two individuals. This is followed by exchanging all characters either to the right or left of this point. Therefore, two new individuals are generated. On the other hand, mutation generates new individuals by modifying one or more of the gene values of an individual offspring after crossover. Values for those operators are mentioned later when the genetic strategy is given. The new individuals are then concatenated into the current population to generate the new population for the next generation. And the process is repeated until a maximum number of generations is reached.

It is important to mentioned here that a different evolutionary strategy than total replacement is used and detailed further on. When the objectives are combined in one scalar function as in Chapter 4, the same number of offsprings are generated as parents and a total replacement policy is used which takes approximately three hours of computational time. When using this strategy for the multi-objective optimisation problem, it takes at least four times longer, because the GAs are dealing now with four objectives and the ranking mechanism is working by comparing each objective for each possible solution (individual). In addition, many duplicate solutions are generated during FLC tuning for the nominal model and also the non-dominated solutions from each population are concatenated with the next one, rendered in a larger number of evaluations of the control parameters. In order to prevent the expansion of the population, a different GA strategy from that in Chapter 4 is proposed here, see fig. 5.1. A population of 100 individuals is main-

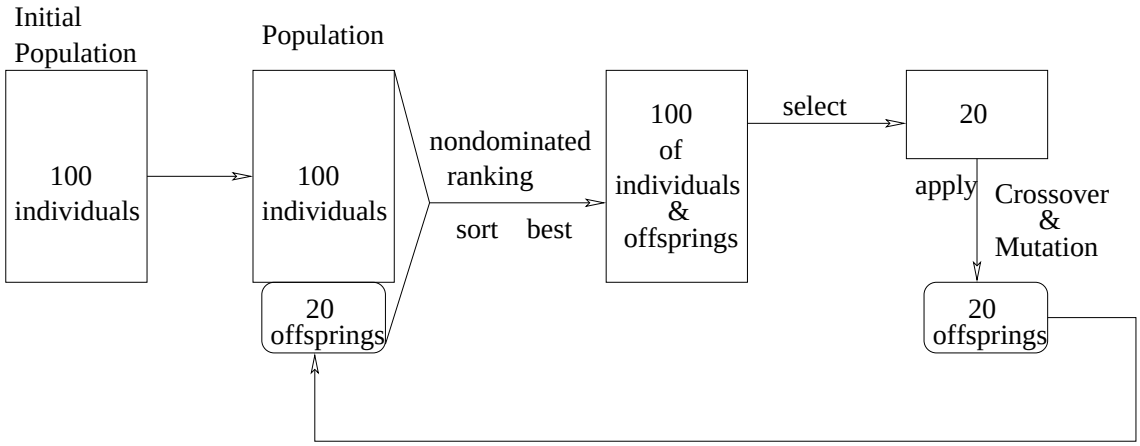


Figure 5.1: Genetic algorithm structure

tained by the algorithm. In each generation, 20 individuals are selected for breeding. Crossover is performed at a rate of 0.9, with intermediate crossover being used for the real values and uniform multi-point crossover for the binary part. A mutation rate of 2/137, with 137 being the length of the chromosome and a selective pressure of 1.7 is used. The high crossover value and the low selective pressure is used to slow convergence and to help prevent a local optimum being exploited. The 20 new individuals are evaluated and then concatenated with the old population, forming a set of 120 individuals. Non-dominated ranking is then applied to this set and the best 100 are taken for the next generation. In this application, much of the feasible space of the controller is not used (see the results section). The genes responsible for these areas will settle to some semi-random state. That is why some solutions with very similar control surfaces may have very different chromosomes. This feature disturbs the sharing process, so a fixed value of $\sigma_{share} = 0.5$ was used, as varying σ_{share} has little effect for this problem.

The main strengths of the non-dominated sorting approach is that it can handle any number of objectives independently and takes account of non-domination during the ranking process. In the next section unique method is proposed to incorporate preference information into the evolutionary multi-objective algorithm by using the optimistic reference point approach to direct the GAs search towards specified areas for optimal solutions.

5.4 Optimistic Reference point approach

As shown by Hwang et al [106], preference articulation can be given by assigning weight coefficients, priorities, or goal values which indicate desired levels of performance in each objective dimension. The way goals are interpreted may vary. The goals may represent minimum levels of performance to be attained, Utopian

performance levels to be approximated, or ideal performance levels to be matched as closely as possible. Goals are usually easier to set than weights and priorities, because they relate more closely to the final solution of the problem.

Depending on how the multiple objectives have been considered, they will affect the evolutionary algorithm behaviour in terms of convergence and searching through feasible regions for acceptable solutions. One way to explore this problem is to define the closed loop performance criteria as four objectives using the Reference Point approach (Weistroffer [114]), which is a kind of preference information for the GAs. Fonseca and Fleming [125, 127] have demonstrated the need for some degree of preference articulation in Pareto based evolutionary optimisation by using a goal attainment method. In their work, they have also achieved interactive optimisation with the DM. In this case, if the DM finds the candidate solutions unacceptable, DM can refine the preferences in order to stimulate the GA to move in to a different region of the non-dominated set. In our case, this kind of active interaction has not been necessary as we shall see.

In the optimistic reference point approach, for example, the DM initially specifies optimistic objective function values (not achievable simultaneously) as the desired values. A solution is found by minimizing the under achievements of the objective function values with respect to the specified desired values. The optimistic approach can be viewed as the special case of the reference point approach in which all reference values consistently exceed the objective function values at all the intermediate solutions. The Reference Point approach has been applied to a scalar optimisation problem using a surrogate aggregating function, as given by Weistroffer [114], Stoyanov et al [128]. In our case, the objectives are treated separately and, by specifying the desirable ranges for each, GAs have achieved simultaneous convergence on all objectives without having the opportunity to stack in a local area on one of the objectives. If references are not specified, it may well be possible for a genetic drift to appear.

Generally, the objective criteria are not comparable and the numerical values may differ considerably. A procedure for normalization must be used to convert the criteria $y_j(\mathbf{x})$ into a dimensionless function $\eta_j(\mathbf{x})$ for which usually $\eta_j(\mathbf{x}) \in [\mathbf{0}, \mathbf{1}]$. The optimistic reference point approach given by Weistroffer [114], and followed by Narula and Weistroffer [113] uses a function of losses to represent the losses from the ideal values $y_j^\star$ for the objectives given by:

$$\eta_j(x) = \frac{y_j^\star - y_j(\mathbf{x})}{y_j^\star}, j \in [1, \ldots, m]. \tag{5.1}$$

If the ideal values $y_j^\star$ are very small numbers or $y_j^\star \longrightarrow 0$, the following alternative form can be used:

$$\eta_j(x) = \frac{y_j^\star - y_j(\mathbf{x})}{y_{jmax} - y_{jmin}}, j \in [1, \ldots, m]. \tag{5.2}$$

where y_{jmax} and y_{jmin} are respectively the maximum and the minimum values of the criterion $y_j(x)$ in $x \in X$, which define the set of feasible solutions.

This approach is called *optimistic*, because $y_j^\star$ are the most desired values for each objective. The form of equation (5.2) is applied to all four closed loop performance criteria described in the next section.

5.4.1 Closed loop performance criteria

Rise time (t_r), steady state error, overshoot $y(t_p)$ and settling time (t_s) are the important criteria with which to judge the quality of a unit step response. They are shown in fig. 5.2 and are used as objectives for the optimisation process. The aim is to minimize each within a specified range, as required by the missile engineer.

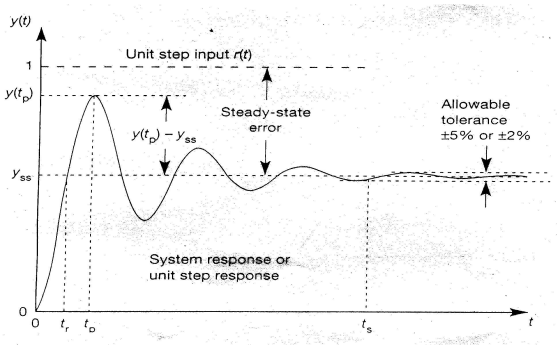


Figure 5.2: Closed loop criteria

- Steady state error in %

$$e_r = \frac{\mid y_{demand} - y_{final} \mid}{\mid y_{demand} \mid} * 100$$

- Evaluation of percentage overshoot $\%OS$

The percentage overshoot, $\%OS$ is given by:

$$\%OS = \frac{y_{max} - y_{final}}{y_{final}} * 100$$

where $\%OS$ is the amount that the response overshoots the steady-state or final value, expressed as a percentage of the steady-state value. y presents the lateral acceleration or the side-slip velocity, depending which one is controlled. y_{max} is the value at the time, the response reaches its first maximum peak.

- Rise Time t_r

Rise time is defined as the time for the response to go from 10% to 90% of its final value.

$$t_r = time(y_{final} * 0.9 - y_{final} * 0.1)$$

- Settling Time t_s

Settling time (t_s) is the amount of time required for the transient damped oscillations to stay within $\pm 2\%$.

$$t_s = time((y > y_{final} * 1.02) \text{ or } (y < y_{final} * 0.98))$$

Rise time, settling time and peak time yield information about the speed of the transient response. This information can help the designer to determine whether or not the speed and the nature of the response degrade the performance of the system. The objective values are all expressed in different units and a normalization procedure is necessary for further use, which is explained in the next section.

5.4.2 Function of losses - preference information

The closed loop performance criteria are chosen as the following:

- Steady state error:

$$\eta_{1j}(x) = \frac{Er_j^\star - Er_j(\mathbf{x})}{Er_{jmax} - Er_{jmin}}, j \in [1, \ldots, m]. \tag{5.3}$$

- Overshoot:

$$\eta_{2j}(x) = \frac{Os_j^\star - Os_j(\mathbf{x})}{Os_{jmax} - Os_{jmin}}, j \in [1, \ldots, m]. \tag{5.4}$$

- Rise time:

$$\eta_{3j}(x) = \frac{Tr_j^\star - Tr_j(\mathbf{x})}{Tr_{jmax} - Tr_{jmin}}, j \in [1, \ldots, m]. \tag{5.5}$$

- Settling time:

$$\eta_{4j}(x) = \frac{Ts_j^\star - Ts_j(\mathbf{x})}{Ts_{jmax} - Ts_{jmin}}, j \in [1, \ldots, m]. \tag{5.6}$$

where m are the number of evaluated individuals.

Table 5.1 shows the reference points used in the objective calculations. The most desired values $y_j^\star$ for each objectives are defined to satisfy the missile control engineer requirements for the Horton model.

	Ideal point	Maximum point	Minimum point
Steady State Error	$Er_j^\star = 0.0[\%]$	$Er_{jmax} = 2.0[\%]$	$Er_{jmin} = 0.0[\%]$
Settling time	$Ts_j^\star = 0.15[sec]$	Ts_{jmax}=0.25[sec]	$Ts_{jmin} = 0.1[sec]$
Rising time	$Tr_j^\star = 0.08[sec]$	$Tr_{jmax} = 0.14[sec]$	$Tr_{jmin} = 0.07[sec]$
Overshoot	$Os_j^\star = 4.5[\%]$	$Os_{jmax} = 25.0[\%]$	$Os_{jmin} = 2.0[\%]$

Table 5.1: Closed loop performance criteria

5.4.3 Decision Making

Bearing in mind that there will be more than one solution, the influence of the decision maker is of utmost importance. Pareto optimality is not the only step towards solving a multi-objective optimisation problem. The choice of a suitable compromise solution from all non-inferior alternatives is also important. It is not only problem dependent, it depends also on the subjective preferences of a DM. Hence, the final solution to the problem is the result of both the optimisation process and the decision process. Depending on how those two are combined in the search for compromise solutions, the following groups have been identified by Hwang et al [106]:

- no articulation of preferences is needed from the DM
- *a priori* articulation of preference - expressed before the search is run
- *interactive* articulation of preferences - the preferences are expressed and can be altered as the search is running
- *a posteriori* articulation of preference-expressed after the search is run, the DM chooses from a set of possible solutions provided at the end of the run.

A priori methods clearly allow a degree of certainty by fixing the targeted outcome in advance of the optimisation run.

Interactive methods allow the user both to react to changing situations in the application problem, and to interact with the optimisation process by updating objectives or goals as the optimisation is conducted.

A posteriori methods have the advantage of allowing no possible solution to be eliminated prematurely in the optimisation process by preserving all potential outcomes. This may be seen as a disadvantage if there are a large number of solutions as the user may be presented with an excessive number from which to make a choice.

Ultimately, the preferred method of DM is likely to be influenced by the problem requirements. In fact, multi-criteria decision making (MCDM) is a field in which we are all well practiced in our personal lives, as we make decisions which involve multiple conflicting criteria daily, without the support of a formal approach. The very nature of multiple criteria problems is that there is much information of a com-

plex and conflicting nature, often reflecting differing viewpoints and often changing with time. One of the principal aims of the MCDM approaches is to help decision makers organize and synthesize such information in a way which leads them to feel comfortable about making a decision, as stated by Zeleny [112]:

" The decision unfolds through a process of learning, understanding, information processing, assessing and defining the problem and its circumstances".

The multi attribute decision problem can be expressed in matrix format as:

$$D = \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{matrix} \begin{bmatrix} x_{11}, x_{12}, \ldots, x_{1n} \\ x_{21}, x_{22}, \ldots, x_{2n} \\ \vdots \\ x_{m1}, x_{m2}, \ldots, x_{mn} \end{bmatrix} \tag{5.7}$$

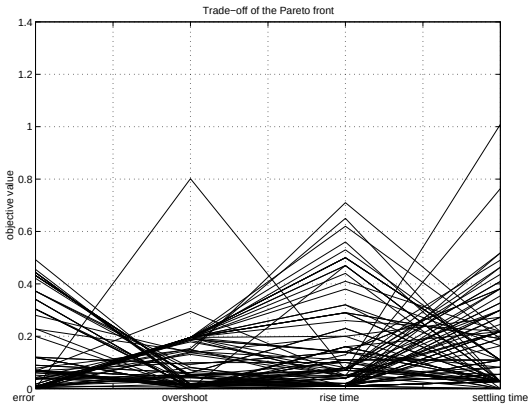
Where $A_i, i = 1, \ldots, m$ are possible courses of action (referred to as alternatives); $x_{ij}, i = 1, \ldots, m, j = 1, \ldots, n$ are attributes with which alternative performance are measured; i is the performance (or rating) of alternative A_i with respect to attribute j. With respect to our problem the matrix is:

$$D = \begin{matrix} Chrom_1 \\ Chrom_2 \\ \vdots \\ Chrom_m \end{matrix} \begin{bmatrix} Er_1,\ Ts_1,\ Tr_1,\ OS_1 \\ Er_2,\ Ts_2,\ Tr_2,\ OS_2 \\ \vdots \\ Er_m,\ Ts_m,\ Tr_m,\ OS_m \end{bmatrix} \tag{5.8}$$

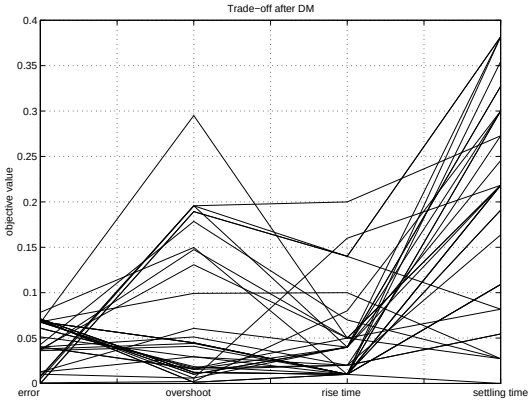
where $Er_i,\ Ts_i,\ Tr_i,\ OS_i$ are the closed loop performance criteria and $Chrom_i$ stands for chromosomes which define the trajectory controllers and represent the compromise individuals at each generation.

In a typical run, about 95% of the solutions are non-dominated and of highly competitive nature. Final decision is made based on the maximum acceptable level for each objective, which corresponds to the pre-specified maximum desired values of the reference point approach. Only solutions which are below the maximum desired value on each objective are considered. Fig. 5.3 shows the trade-off plots for the closed loop performance criteria in the final population (last generation). Most of the solutions are non-dominated and the one shown in fig. 5.3b, within minimum and maximum range of each criteria, as specified by the DM has been considered as acceptable. The objective values $\eta_j(x)$ in fig. 5.3 are normalized using equations (5.3, 5.4, 5.5 and 5.6). Each continuous line in fig. 5.3 represents one set of the four closed loop criteria for one alternative solution (the optimised fuzzy controller parameters). A strong conflict can be seen between overshoot and rise time, which is expected, with not so much conflict between rise and settling times. The magnitude of the objective values are scaled, hence they do not represent their physical values. Also, on this plot we cannot see the conflict between settling time and overshoot or between settling time and steady state error. That is why a detailed trade off surface for the individuals in a population is illustrated in figures 5.4, 5.5

and 5.6. The intention here is to show how solutions evolve within generations, so trade off dynamics can also be seen. Before the convergence is achieved, there are some unacceptable solutions clustered in areas of higher objective values, as shown in fig. 5.4, that die out as the evolution progresses. After convergence (after approximately 30 generations, as shown in fig. 5.7) most of the objective values are within the pre-specified range, which is an indication that solutions have converged towards the desired feasible area (see the scale in fig. 5.5). There are some solutions strongly dominated on one objective which obviously are not taken under further considerations by the DM. Each star represents the objective value of an individual within a current population. Also, the non-dominated solutions are almost identical at the 151^{th} generation and at the 250^{th} generation when the optimisation process is complete. The fact that there are no major differences suggest that an earlier stopping mechanism could be used in this case.



a) Before decision making



b) After decision making

Figure 5.3: Trade-off surfaces

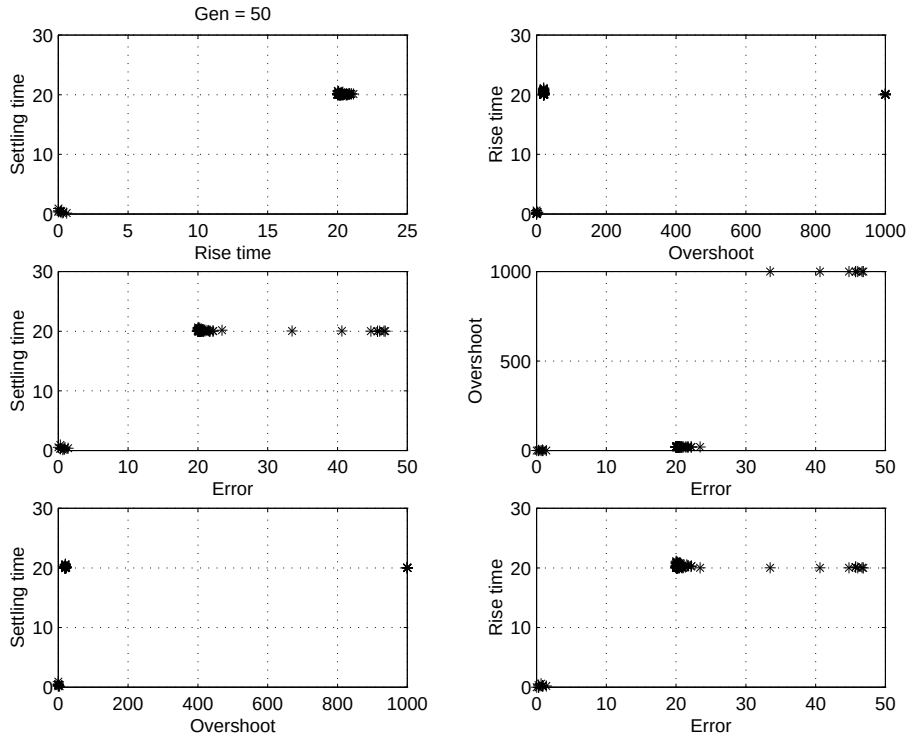


Figure 5.4: Trade-off between each objective at generation 50

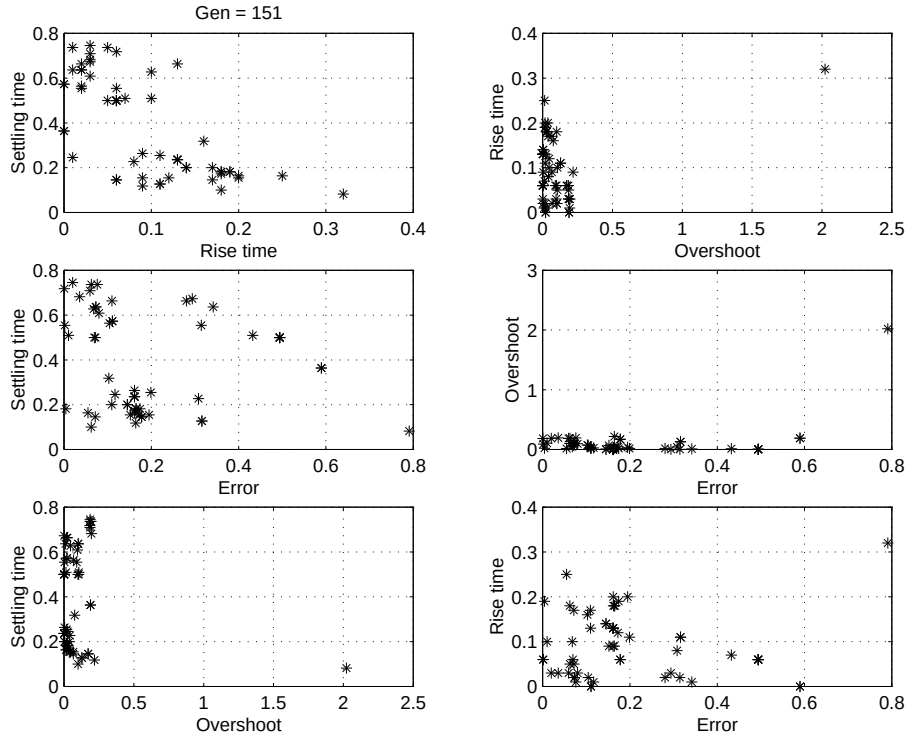


Figure 5.5: Trade-off between each objectives at generation 151

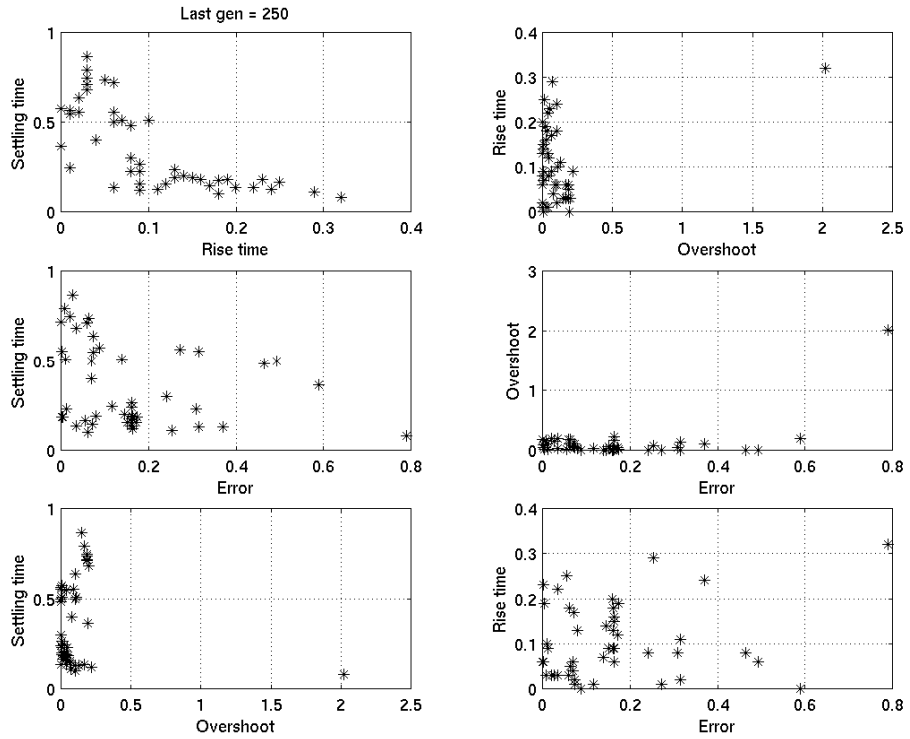


Figure 5.6: Trade-off between each objectives at last generation 250

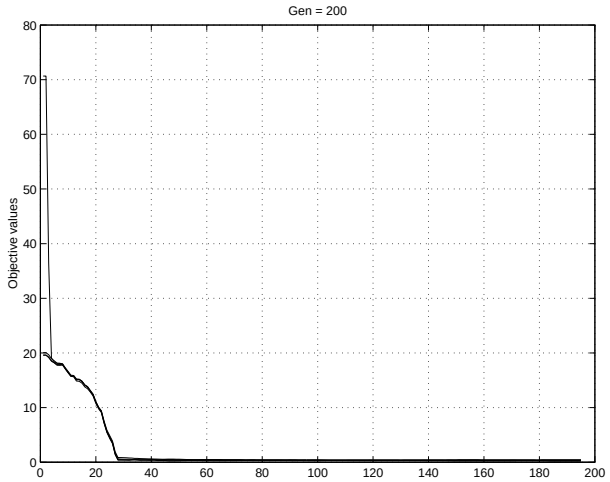


Figure 5.7: Genetic algorithm convergence

5.4.4 Results using the reference approach

1. Side-slip velocity control

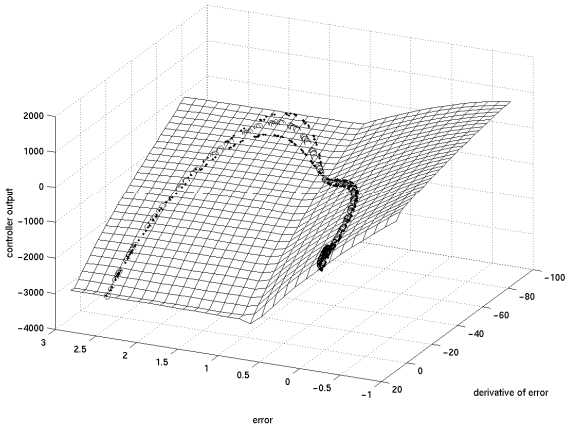
The nonlinear control law of the autopilot design for the SISO system has been considered. It is defined as **Design 1** in Section 3.3 of Chapter 3. A fuzzy logic trajectory controller has been added in an outer loop and multi-objective optimisation using a genetic algorithm has been used to determine the fuzzy logic control parameters, as given by Blumel et al [129], [130].

Fig. 5.8 shows the control surface section used and the corresponding side slip velocity responses for two alternative solutions from the non-dominated set. These results are obtained using five membership functions. Solution (b) is one of the best selected according to all four closed loop performance criteria: steady state error within (2%), (5%) on overshoot, a fast rise time ($0.05s$) and very close to ideal settling time($0.12s$). Solution (d) has an acceptable steady state error($< 2\%$), almost no overshoot, but is very slow on rise time and on settling time. From a practical point of view, the first one would be preferred by a missile control engineer. Both solutions have similar control surfaces showing a 'winged' structure (see the left side of fig. 5.8). By looking at the phase portrait pattern, further information for the membership functions and rules used can be extracted. The circles present the nominal case of aerodynamic coefficients and the dotted lines are for the uncertain case. Solution (f) is very bad on steady state error, and hence, is not acceptable.

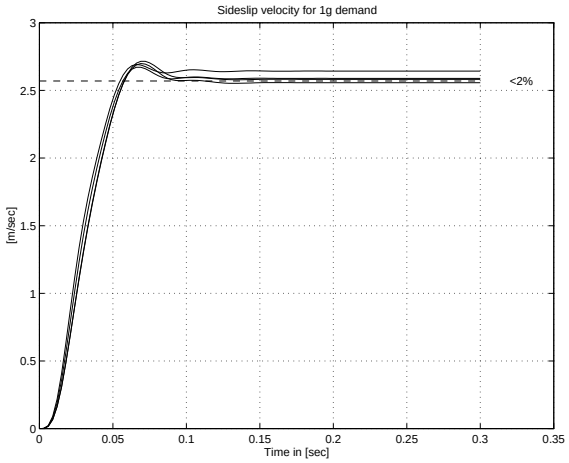
The fuzzy logic controller is tuned for the nominal case of the aerodynamic coefficients, a demand of 2.57 m/s corresponding to 1g lateral acceleration, and is tested for parameter variations within the ranges specified in Section 2.8 of Chapter 2. Two particular combinations of model variation have been used:

1. $[\ C_{yv_{min}} \quad C_{y\zeta_{min}} \quad C_{nr_{max}} \quad X_{cp_{max}}\ ]$
2. $[\ C_{yv_{max}} \quad C_{y\zeta_{min}} \quad C_{nr_{max}} \quad X_{cp_{min}}\ ]$

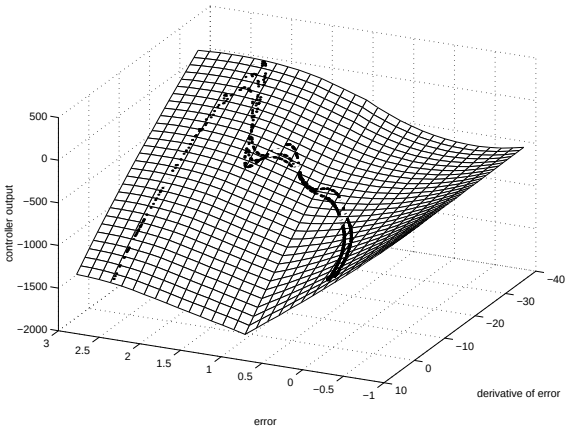
Robust performance within 2% relative steady state error is achieved. If we look at the side-slip velocity response produced with the fixed gain trajectory controller shown in fig. 4.11 in Chapter 4, we can see that for the same combinations of the multi-model airframe dynamics of aerodynamic coefficient uncertainties, an error of about 7% is achieved. On the other hand, in this case the fuzzy trajectory controller has improved the robustness against these uncertainties by 5%.



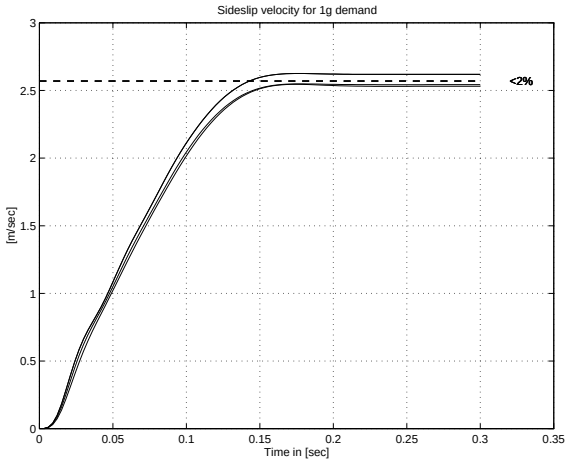
a) Fuzzy control surface



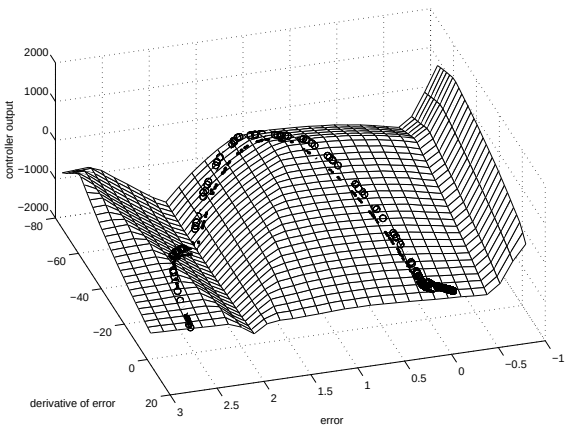
b) Side-slip velocity response



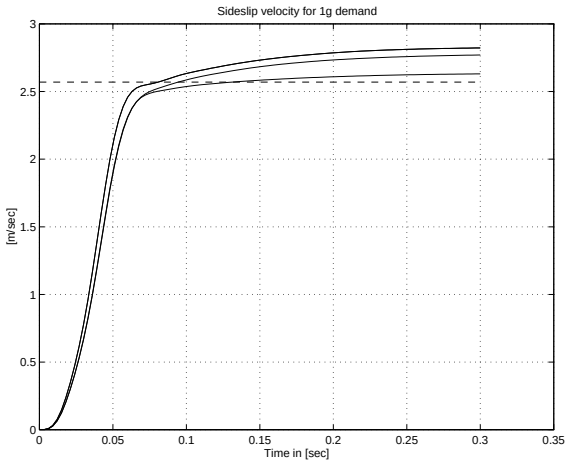
c) Fuzzy control surface



d) Slow on rise time



e) Fuzzy control surface



f) Unacceptable solution

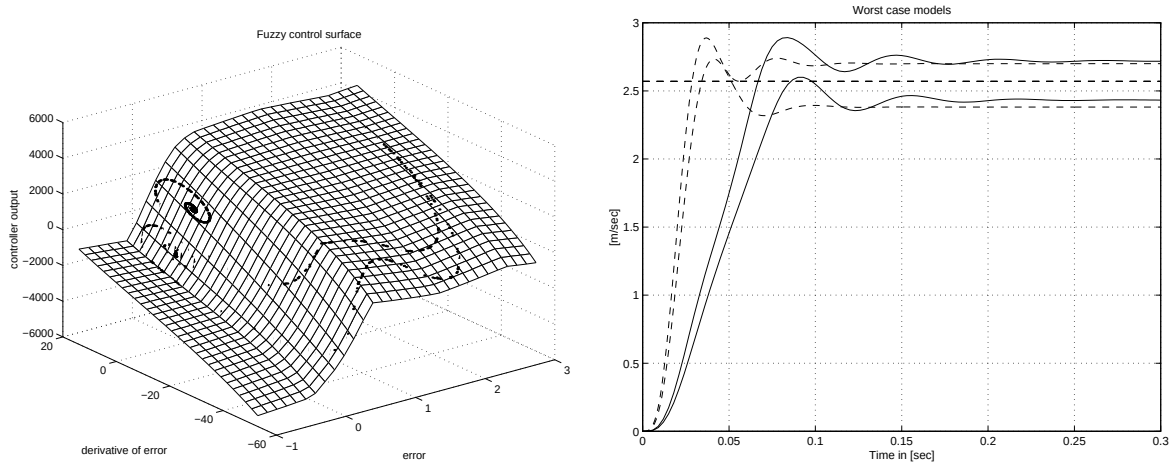
Figure 5.8: Three alternative Pareto solutions with velocity control

In this chapter, the fuzzy control parameters were mainly tuned for the nominal model coefficients because the process was time demanding when multi-objective optimisation was considered. One reason for this is because of the use of binary coding. The chromosome structure with the binary coding of the rule base system allowes robust properties of the fuzzy controller to be used but the optimisation algorithm takes 12 hours on a 300 MHz Unix workstation. Also, in order to evaluate each chromosome in the population, each controller is tested on all vertex points models.

Two changes have been made to decrease processing time. First, a generation gap was introduced to prevent a change in number of individuals to be evaluated at each generation. Second, the chromosome structure was modified to real-integer coding which reduced the length of the chromosome by a factor r, where r is the number of membership functions. For evaluations of chromosomes on one trial only (i.e. 1 set of model coefficients and 1 required demand), the processing time decreased from 12 down to 5 hours using this approach. Four membership functions and four objectives have been considered.

2. FLC tuning on vertex points models

The fuzzy trajectory controller has been tuned for a set of worst case vertex models, 4 and 8 (i.e. V_{min} and V_{max} values on side-slip velocities at steady state, see fig. 4.10 in Chapter 4). Therefore the controller is robust against any parametric uncertainties which may appear within the range defined by the vertex models. The exact model within the vertex models is determined by the flight condition and will also be a function of the aerodynamic coefficients $(C_{yv}, C_{y\zeta}, X_{cp}, C_{nr})$ within their uncertainty ranges. Most of the results shown in the thesis were obtained based on nominal model simulations to a step input. As a result, the fuzzy logic control surface has been exercised very little and hence the robust properties of the resulting controller were not as good as the ones obtained based on extensive simulations on all vertex models. There is still a problem associated with using a step input, however. The difference from fig. 5.8 of Chapter 5 is that the fuzzy gain surface shown in fig.5.9 has been tuned for a set of vertex points models, hence can maintain robustness for any uncertainties which may arise within the ranges of these models. Multiple solutions were also obtained but not shown. Initially the controller was tuned on 5 trials (i.e. 4 vertex points models and the nominal one) for 250 generations. The optimisation algorithm has taken approximately 27 hours on a PC 300 MHz. A real-integer chromosome structure has been used. Most of the Pareto solutions at the last 5 generations were with a st.st. error bigger than 10%. For evaluations of 22500 fuzzy controllers(i.e. $NIND.GGAP.trials.MAXGEN$), 250 generations were not enough to obtain satisfactory closed loop performance criteria. The fuzzy controller was then tuned on 3 trials (two worst case models and nominal one), but still the optimal objective values were not satisfactory. Finally, the GA were seeded with a set of Pareto solutions of the last 5 generations from the previous run. Then for another 200 generations the optimisation was carried on two



a)Fuzzy gain controller b)Velocity responses

Figure 5.9: Results

trials (i.e. the worst case models). The mutation rate was reduced from $(1/NVAR)$ to $(0.5/NVAR)$ to ensure the survival of good solutions. In fig. 5.9b, dashed line shows the velocity responses when the fixed gain controller is tested on worst case models and the solid line shows the velocity responses when the fuzzy gain optimal controller is tuned and tested on the same models. The responses produced with the fuzzy gain controller are very close to the bounds of responses produced with the fixed gain controller. Also, when the fuzzy gain controller was tested on all vertex points models, the performance criteria on steady state error remained within the bounds obtained by the worst case models, hence the FLC has maintained robust properties of the closed loop system.

3. Lateral acceleration control

The difference here is that the lateral acceleration control and the nonlinear control law detailed as **Design 2** in Section 3.3 of Chapter 3 is used. Again, only a single plane for the yaw or pitch channel has been considered. As a result of multi-objective optimisation, multiple solutions are obtained from which the designer can choose the one which best satisfies his requirements. In fig. 5.10 we show a set of lateral acceleration responses with a variety of closed loop performance criteria. Some are unacceptable, with high overshoot values, are very slow on rise time or settling time, but some are very good with almost no steady state error and no overshoot. Fig. 5.11 shows the fuzzy gain surfaces for three of them and the corresponding acceleration responses of these. (b) is the best on steady state error, and (d), which is within 6% error from the demand, is probably not acceptable, although it has no overshoot and has a satisfactory rise time. Finally, solution (f), which is too slow on rise time and settling time but within 3% on steady state error, may not be considered as an acceptable solution by the designer. The dashed line represents the augmented acceleration which shows an almost identical closed loop performance as that for lateral acceleration. The only difference is in the non-minimum phase effect which can be seen in the solid line for the lateral acceleration. However, it should be remembered that here the augmented acceleration is used to design the nonlinear control law, but the actual lateral acceleration is used as the controlled output. The fuzzy gain surfaces for the acceleration control are more nonlinear compared with the surfaces for the side-slip velocity control.

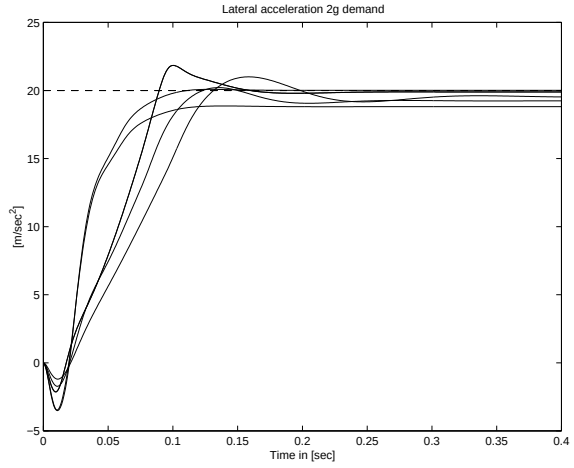
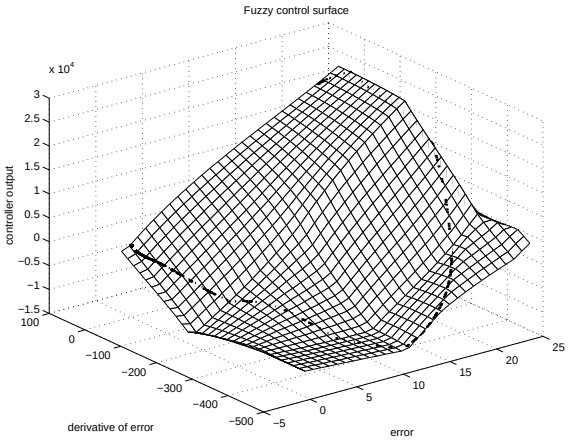
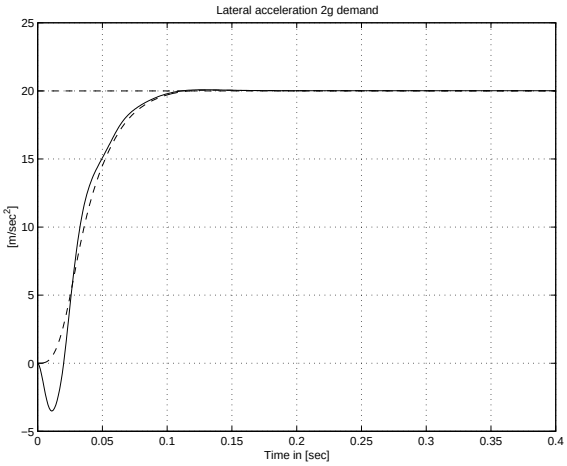


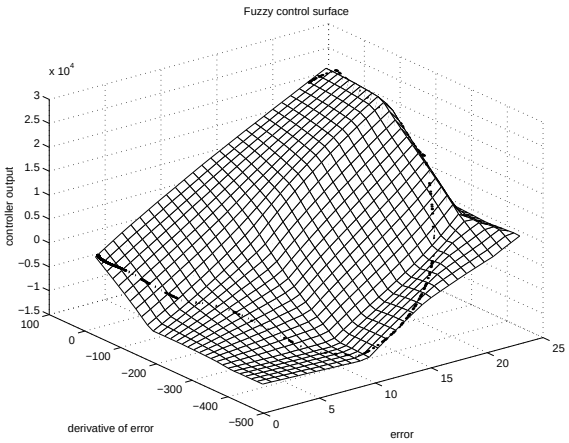
Figure 5.10: A set of lateral acceleration responses



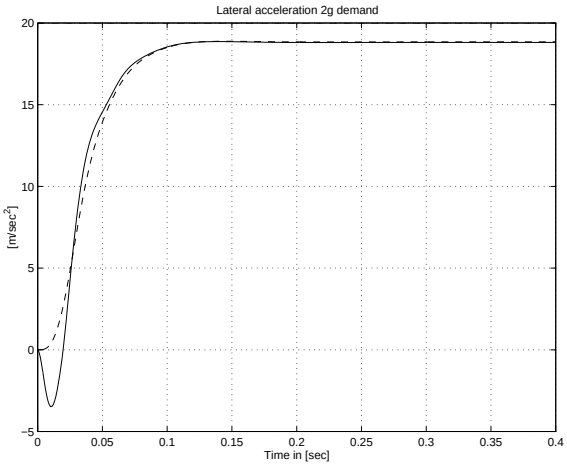
a) Fuzzy control surface



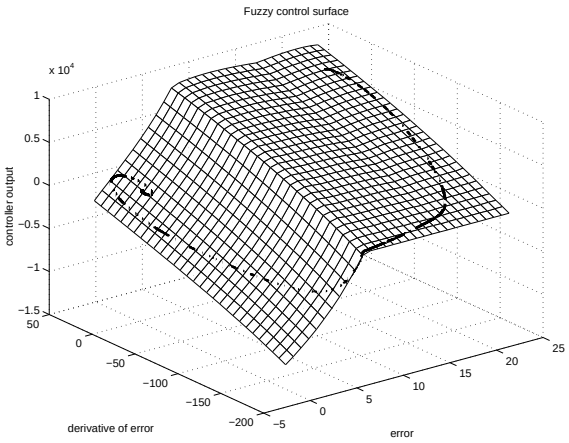
b) Best on steady state error



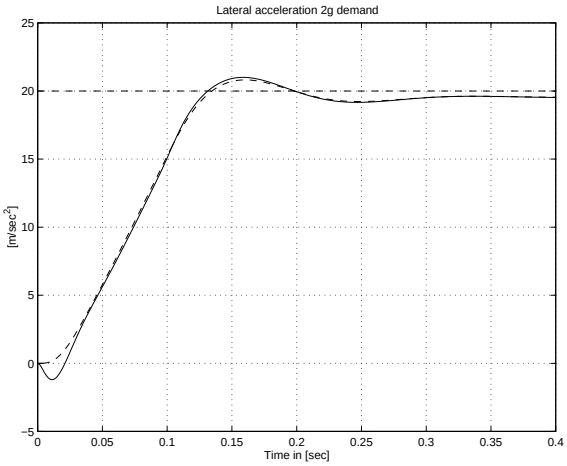
c) Fuzzy control surface



d) $< 6\%$ steady state error



e) Fuzzy control surface



f) Acceleration response

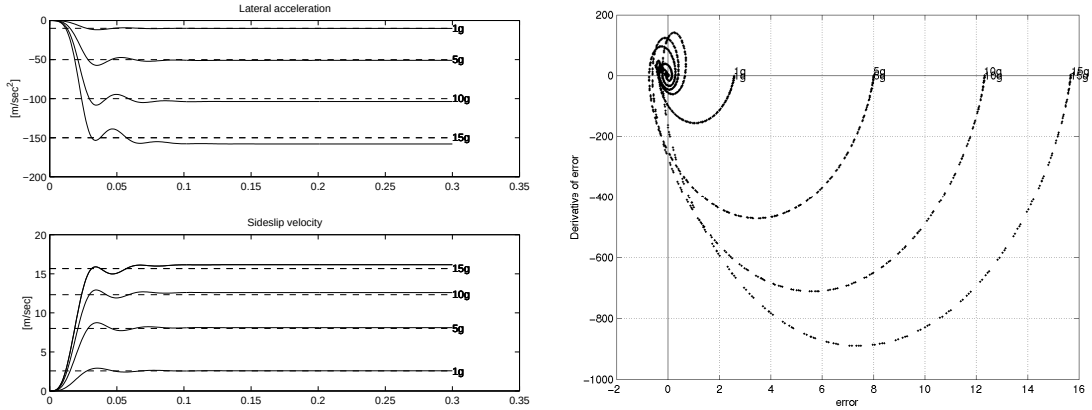
Figure 5.11: Three alternative Pareto solutions with acceleration control

5.4.5 Results for Fuzzy logic scheduled controllers multiple demands case - 1g, 5g, 10g, 15g

Multiple solutions are obtained when a large range of different velocity demands are required. The optimisation process is examined using two trial 5g and 15g demands and individuals with bigger objective values are evaluated for further optimisation. It may be noticed that the values of the criteria are very similar for all four demands, however a little offset on the rise time can be observed on each response as, for every required demand, the system changes, which then alters the fuzzy logic control properties.

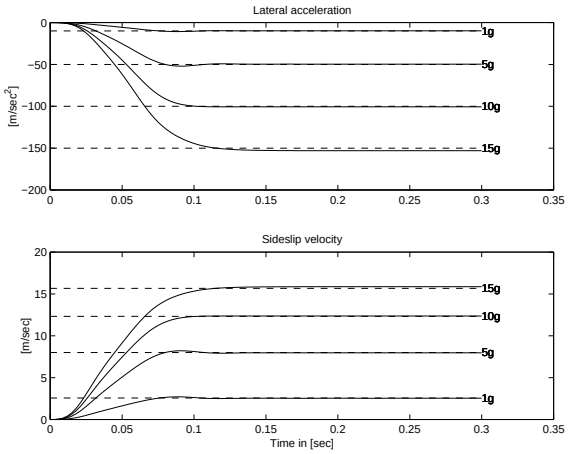
In fig. 5.12 a fixed gain trajectory controller is illustrated for all four demands. The presence of steady state error is due to the neglected fin control surface term in the system. Conversely, all fuzzy trajectory controllers (see fig. 5.13) have been able to provide less fast solutions but with almost no oscillations and very little steady state error. The closed loop criteria here have been considered such that the reference point approach has been used to determine optimal solutions for the side-slip velocity control of the fuzzy autopilot design.

Various solutions have been presented to show the powerful interpolative mechanism of the fuzzy scheduled controllers when multiple demands are required. The scaling factors of the fuzzy controller have been determined by using polynomial fitting for each demand. The optimisation procedure is able to find multiple solutions (alternative fuzzy controllers) in terms of closed loop performance criteria and is able to tune the control parameters (membership functions and rule base structure) simultaneously for multiple demands.

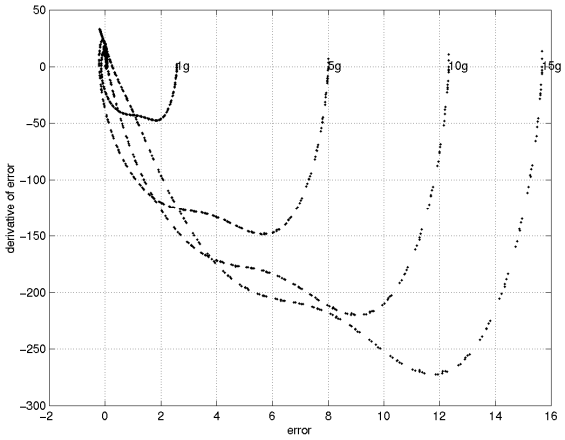


Fixed gain controller Phase portrait

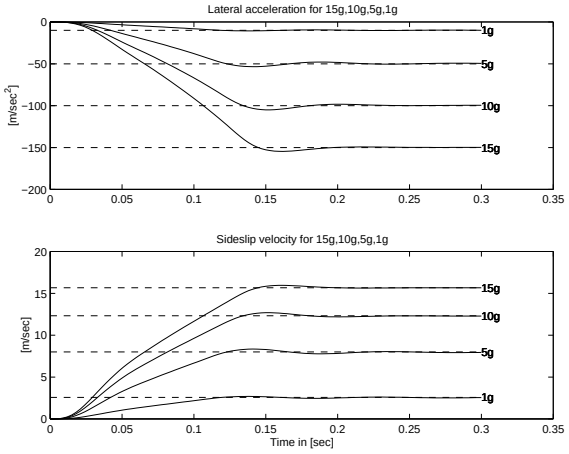
Figure 5.12: Fixed gain trajectory controller



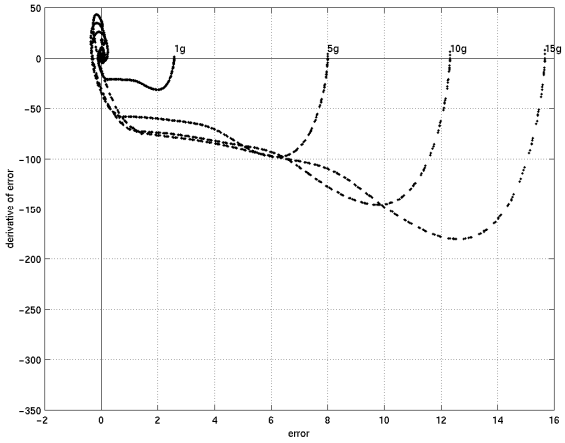
Almost no overshoot



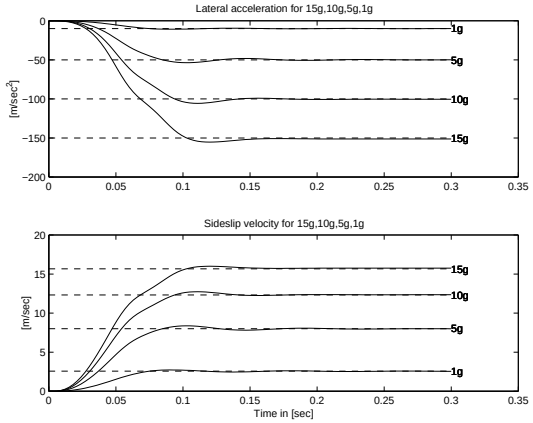
Phase portrait



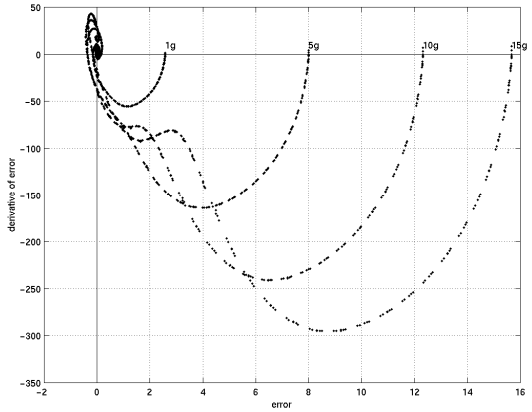
Too slow on rise time



Phase portrait



Best on steady state error



Phase portrait

Figure 5.13: Alternative solutions for multiple demands

5.5 The objectives defined as fuzzy constraints-penalties

Another way of looking at multi-objective optimisation is to specify the objectives as constraints, or in this case as penalties using the fuzzy logic set theory to produce the membership functions. This idea has been proposed previously by Trebi-Ollennu and White [18] and has been further investigated in our research and applied to missile fuzzy control parameters. In [18] a multi-objective fuzzy genetic algorithm optimisation for selecting free control parameters was used. The difference from our work is that Trebi-Ollennu has addressed the multi-objective problem as a scalar optimisation problem and has generated multiple solutions by varying the weights in order to address the relative importance of the fuzzy objectives. We have used this idea of representing the objectives as fuzzy constraints, but have allowed the optimisation procedure to find optimal solutions by using non-dominated sorting, hence ranking each solution based on independent objective values.

Using fuzzy logic theory, the objectives can be presented as penalties with the following membership functions: μ_{OS} in fig. 5.5.1, μ_{T_r} in fig. 5.5.2 and μ_{T_s} in fig. 5.5.3.

- Overshoot penalty

$$\mu_{OS} = \begin{cases} \frac{OS}{10} & OS <= 10 \\ 1 + \frac{OS-10}{10} * 3 & OS > 10 \end{cases}$$

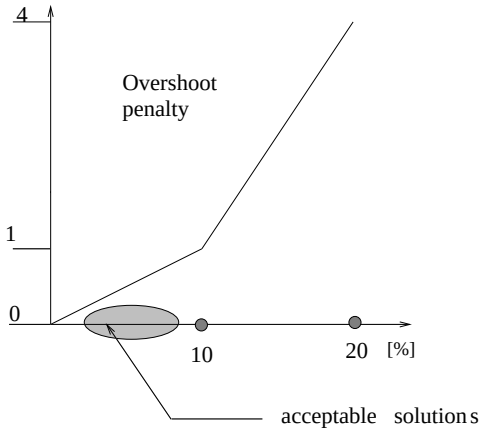


Figure 5.5.1 Overshoot penalty

- Rise time penalty

$$\mu_{T_r} = \begin{cases} \frac{0.05-T_r}{0.05} & T_r <= 0.05 \\ 0 & 0.05 < T_r < 0.1 \\ \frac{T_r-0.1}{0.1} & 0.1 < T_r < 0.2 \\ 1 & T_r >= 0.2 \end{cases}$$

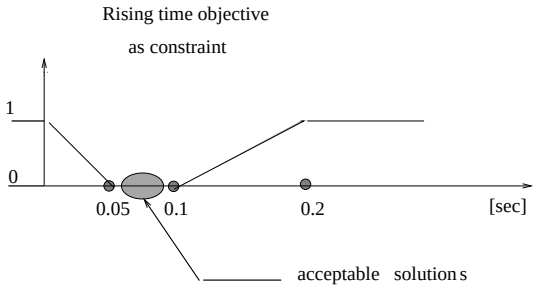


Figure 5.5.2 Rise time penalty

• Settling time penalty

$$\mu_{T_s} = \begin{cases} 1 & T_s <= 0.05 \\ \frac{0.1-T_s}{0.05} & 0.05 < T_s < 0.1 \\ 0 & 0.1 < T_s < 0.2 \\ \frac{T_s-0.2}{0.1} & 0.2 < T_s < 0.3 \\ 1 & T_s >= 0.3 \end{cases}$$

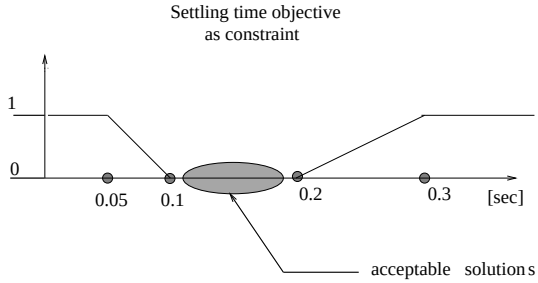


Figure 5.5.3 Settling time penalty

The steady state error is defined as discussed in Section 5.4.1. The specified range for acceptable solutions has been determined by the design engineer and is indicated by the grey area.

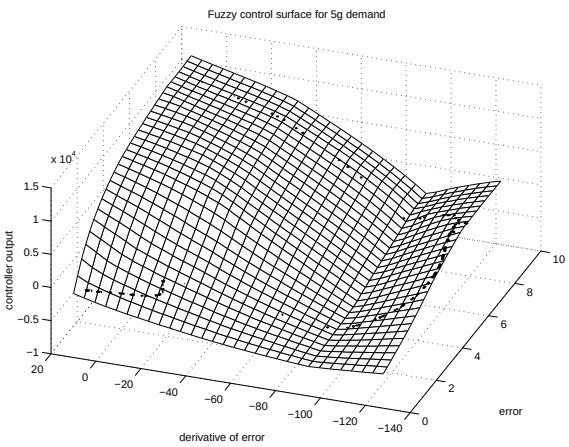
5.5.1 Results of fuzzy multi-objective optimisation

We have chosen to show two alternative solutions in fig. 5.14 with conflicting criteria: one which is not so good on steady state error and one which is too slow on rise time, but is very good on settling time. However, the fuzzy gain surfaces were both smooth and robust in the presence of model uncertainties. In this case, the objectives have been fuzzified as penalties. The objective values of most solutions were within the required range determined by the engineer as the preferable area which is penalised with zero value. According to the way the overshoot criteria has been specified here, it has given the GAs a chance to find solutions with almost no oscillations, which is very important for the autopilot performance, especially when higher demands are required. Using this approach it was quicker to find solutions which satisfy the specified ranges on each objective, although it is difficult to tell how non-dominated they were. This is because, during the minimization process, most of the objectives have been given zero value when they have satisfied the required range, which confuses the non-dominated ranking process. However, in the last generation all solutions have objective values within the specified fuzzified ranges, and hence they were all acceptable and no need of further DM was required. *A priori* DM was sufficient to predetermine desired feasible solutions at the final stage. This way of handling objectives is more convenient for engineers.

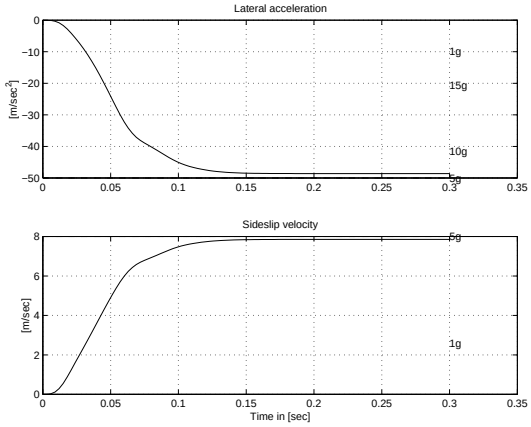
5.6 Concluding remarks

This chapter has examined multi-objective optimisation of the fuzzy logic trajectory controller parameters. This has been achieved by evaluating four closed loop perfor-

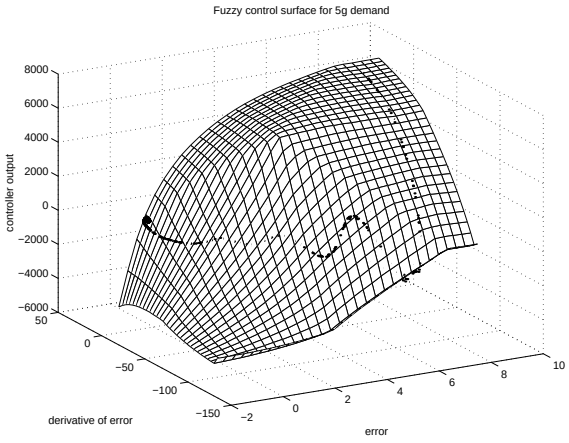
mance criteria which asses the quality of the side-slip velocity or lateral acceleration responses. Evolutionary algorithms, such as genetic algorithms, have produced a set of results that populate the Pareto solution set, allowing the system designer the flexibility of trading one solution against others to achieve a desired performance. By using the Optimistic Reference Point approach we have incorporated preference information into the optimisation procedure, which helps the GAs to converge on areas of preferable solutions for each objective simultaneously. In addition, this idea has been combined with the Pareto based approach which uses non-dominated sorting, hence an efficient Pareto front with optimal (non-dominated) solutions has been produced simultaneously.



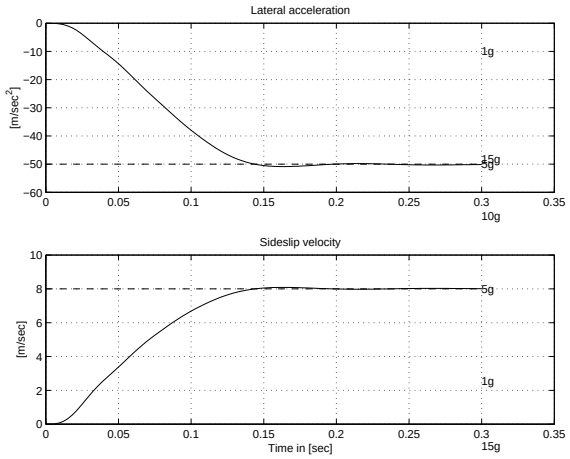
Fuzzy Gain surface for 5g demand



Not good on steady state error



Fuzzy Gain surface for 5g demand



Slow on rise time

Figure 5.14: Two possible alternative solutions for 5g demand

The tradeoff information generated by the evolutionary algorithm can contribute to a better understanding of the control system properties. As we have used non-dominated sorting for forming the Pareto front, the concept of crisp preferability

can be seen to provide useful information to the optimisation process, which would otherwise be faced with sampling a very large tradeoff surface (when all the closed loop performance criteria have been defined in direct form). In consequence, if those objectives are defined as fuzzy constraints (penalties), this can reduce the size of the non-dominated set to search for and it is certain that the final non-dominated solutions at the end of the optimisation are lying within the acceptable regions, so that the engineering design requirements have been satisfied. Conversely, in the first case, the GA is searching in a larger feasible area for acceptable solutions, but only a subset of the non-dominated set can be of practical relevance, which requires further interaction with the DM (i.e. selective process at the end of the optimisation). It is one of the aims of this chapter to analyse the behaviour of the optimisation process, depending on the way objectives have been handled.

The genetic algorithm performance depends on careful adjustment of several parameters and the values assigned to these parameters may affect performance drastically. The determination of GA operators can effect convergence to Pareto optimal front and diversity preservation among Pareto solutions. It is also important to define stopping criteria for GAs based on multi-objective optimisation techniques, because it is not obvious when the population has reached a point from which no further improvement can be reached. The main way used to stop the GA procedure is to use a fixed number of generations or to monitor the population at certain intervals and interpret the results visually to determine when to halt the evolution process. Their parallel nature allows GAs greater ability to explore the search space and lessens the risk of becoming stuck in a local optimum.

In this research all the objectives were treated independently, however, if the missile engineer is interested in solutions that are fast with almost no overshoot, it is possible to alter the rule base structure of the fuzzy trajectory controller to produce desired solutions once the controller has been tuned. By studying in detail the rule base structure it is possible to extract information about which rule affects the rise time, overshoot, settling time or steady state error. This is an area for further research work.

The evolutionary multi-objective algorithms have contributed efficiently to solving a real world problem such as finding the trajectory control parameters of a highly manoeuvrable missile system.

Chapter 6

Conclusions, Discussions and Future work

The main objectives of the thesis were to design an autopilot system for lateral velocity and acceleration control of a highly non-linear missile. The control system was required to be robust in the presence of parametric uncertainties of the model and to be valid for a large range of multiple demands. The other important objective was to obtain multiple solutions of alternative controllers that allow the designer the freedom to choose those which satisfy specified requirements of the closed loop performance so that the autopilot system respond efficiently to guidance commands.

Missiles are mainly employed in a military environment. As technology matures, targets become more agile with a variety of new shapes which are more difficult to detect, hence missiles are required to be highly manoeuvrable and accurate. Generally in a typical guidance scenario they are required to follow a specific trajectory. In order to do that, they should perform agile, fast and flexible manoeuvres which involve rapid changes from low to high altitude and from low to fast speed. Missile systems are well defined in their dynamic behaviour, but are highly non-linear. Most designs have been able to use linear design and analysis techniques. This requires the use of a linear model for an airframe which moves in any direction. The system must be linearized about many combinations of speed along multiple axes in order to account for the complexity of the system. This additionally complicates the design as separate controllers are required for each configuration. The main problem is that there are limited theoretical guarantees of stability in nonlinear operation. Another problem is the computational load, due to the necessity of computing many linear controllers. This research has been focussed at non-linear techniques to design the autopilot system to overcome many of these problems.

Designing an autopilot system is not an easy task. It involves several stages. Starting from the given model we first have to understand the dynamics of that model. How does it behave in open loop? How sensitive is it towards variations in the aerodynamics and then how to design a control algorithm to respond quickly to guidance commands? We have studied non-linear techniques since they can capture the non-

linearities and produce a single design which is valid over the entire envelope. There is very little research reported in the literature for using Feedback Linearization applied to a missile system, which determined our choice to use it. The difference from other people's work is that instead of controlling acceleration through angles and rates we use lateral velocity and augmented acceleration as outputs to control lateral acceleration. Once the system was linearized a simple fixed gain controller was not enough to produce a robust solution, even though for a nominal model, the techniques were successful and the desired trajectory achieved. However in a real scenario the assumption that model parameters will not change is unrealistic, which require robust techniques to replace the fixed gain controller in the outer loop. Using fuzzy logic theory was an appropriate choice since it has been proved in the literature its ability to deal with vague and imprecise information. The only problem with designing a fuzzy logic controller is that it uses trial and error methods to tune its parameters (i.e. the rule base structure and the membership functions). This involves many iterations before an appropriate design is achieved which is time demanding and not very practical. We designed the rule base structure for the nominal model, which wasn't robust to parametric uncertainties. An evolutionary algorithm was then used to tune the fuzzy control parameters which were then robust for the specified uncertainties. Multiple solutions were then obtained by using multi-objective optimisation, which allowed the designer to choose feasible solutions which satisfy specified requirements.

The missile model has been provided by Matra British Aerospace Co. as a real research model. In Chapter 2, we have shown the complexity of the model which is of highly non-linear nature with severe cross-coupling. The polynomials of aerodynamic functions were fitted to the set of curves taken from look-up tables derived from wind tunnel measurements. Wind tunnel techniques provide the best estimates of the aerodynamics but they will always be subject to variability and uncertainty. In real flight scenario, for every instance of this missile type, the aerodynamical functions may deviate from their nominal values. These potential variations introduce parametric uncertainties of the non-linear system. A set of convex models is produced that map the vertex points in a high order parameter space (of the order of 16 variables). The multiple model description of the airframe aerodynamics is tested for sensitivity on the aerodynamic coefficients. The analysis has shown that the missile behaviour is most sensitive to the centre of pressure coefficient, which was expected as this is the coefficient most responsible for airframe stability. The model parameters can take any values randomly generated within the vertex points. This allows more than 1000 models to be exercised and the control system tested for robustness.

In Chapter 3, Feedback Linearization technique is detailed as a nonlinear conventional tool to transform the non-linear system dynamics into a linear form by using state feedback, hence a simple linear control technique can be used in the outer loop. In other words this is dynamic inversion, in which a nonlinear control law is designed to globally reduce the dynamics of the selected controlled variables to integrators.

A closed loop system is then designed to make the controlled variables exhibit a specified command response. Approximate Feedback Linearization has been used for lateral motion control. The main difference from other research work is that instead of using angles or body rates as outputs for the linearization process, lateral velocities and body accelerations were used. The design has retained the order and the relative degree of the system in the linearization process, hence has produced a linearized system with no internal dynamics. Both SISO (the reduced 2^{nd} order system, without interaction between lateral motion and roll) and MIMO (full 5^{th} order) systems have been considered. Desired tracking performance has been achieved assuming an exact knowledge of the nominal model parameters such as: aerodynamic coefficients and missile configuration parameters (i.e., reference area, Mach number, mass, moment of inertia).

One of the main problems with applying the Feedback Linearization technique is that the process produces a system with the same relative degree as the original system, but usually with an order that is less. Indeed, the linearized system order is the same as the relative degree unless pre-compensators are used to artificially change the order and the relative degree. This process results in internal dynamics, which are modes that are effectively rendered unobservable by the linearization process. If the system is non-minimum phase, then the internal dynamics are unstable. In order to produce linearized systems that have no internal dynamics, techniques which preserve the dynamic order of the system were needed. In this work we have selected an output that relates to the variable that is to be controlled, which gives a greater relative degree and we have neglected small terms related to the input during the differentiation process that allow the final relative degree to be achieved and which also retains the order of the system in the linearization process. This resulted in approximate FL, which can only be done if the neglecting terms in the non-linear design will not produce unstable zeros, as they will additionally destabilize the closed loop system and degrade the performance. There will always be an inherent zero in the model because the missile system is non-minimum phase but first it is important to check the stability of that zero before claiming the method is valid.

By applying approximate FL to the Horton missile, the design has resulted in a linear equivalent system with no internal dynamics ("no unobservable" states during the linearization), and with a design of a trajectory control which has given small tracking errors for both lateral velocities and accelerations. The simulation results have shown desired tracking performance for a large range lateral acceleration demands up to 100 m/sec^2. Full decoupling for the highly non-linear missile system has been achieved. All designs for the SISO and the MIMO systems have involved increasing the speed of response sufficiently for a linear approximation to be inadequate for design purposes, and the responses for both small and large demands have been shown to be invariant. Although the Horton model has been designed for Cartesian control, it has also been useful to apply Polar Control as it has significantly reduced the computational load of the non-linear control design, which is important (less risky and less expensive - computationally speaking).

The principal disadvantage associated with Feedback Linearization is the lack of established methods for robust synthesis. By applying FL, desired tracking performance has been obtained by assuming an exact knowledge of aerodynamic coefficients and missile configuration parameters in the entire flight envelope. In a real flight scenario, this assumption is not valid and for parameter variations from nominal model, the Feedback Linearization cannot guarantee desired closed loop performance neither is robustness provided.

In Chapter 4, a robust non-linear trajectory controller based on fuzzy logic has been applied in the outer loop to provide robustness for the feedback linearizable system with respect to significant parametric uncertainty introduced into the system through the aerodynamic coefficients. The fuzzy feedback linearized control design has been found to be more effective for improving the transient and steady state performances than the fixed gain feedback linearized control design. The ability to improve the closed loop performance while managing uncertainties has shown the advantage of using the fuzzy logic theory. It has provided the means of systematically synthesising various fuzzy rules to produce decision actions so that the complex missile non-linear system can be controlled. This allows flexible robust manoeuvrability. It has been difficult to determine by hand the fuzzy control parameters which will count for any of the parametric uncertainties generated in the system. The trial and error mechanism has been involved with many iterations before an appropriate design has been achieved which is time demanding and not very practical. Hence it has been replaced by evolutionary algorithm optimisation using genetic algorithm for better adaptation and robustness. The rules and the membership functions of the fuzzy trajectory controller have been generated simultaneously. Each chromosome consisted of a rule set and its associated membership functions. This allowed the GA operators to integrate multiple fuzzy rule sets and their membership functions at the same time. The hybrid control strategy has been validated via extensive computer simulations and has produced a successful robust non-linear autopilot design. Although complex, the control system is reliable over wide variations in plant dynamics and offers an elegant solution to designers.

However, the designer should be careful when looking for quality of the closed loop performance within reasonable computation time. It is usually a trade-off, because fuzzy systems perform better when more membership functions are used, but unfortunately this increases computational time, as the size of the rule base structure in the fuzzy mechanism increases significantly. Hence the trade-off between processing time and performance is important to take into account. Still new fast technology can help for implementation of such systems.

Again in Chapter 4, for the normalised fuzzy logic engine, the three scaling factors (error,derivative of error and output) for each required lateral acceleration demand $(1g, 2g, \ldots, 15g)$ have been determined via simulations based on the results obtained with a fixed gain trajectory controller for the nominal model. Then a polynomial

fitting has been used to interpolate between large range of required demands for side-slip velocities in order to obtain the scaling factors of the FLC inputs and outputs for each demand. As a result smooth transition of the scaling factors has been achieved when different demands have been required within the above mentioned range. This has determined the smooth transition and gradual interpolation between the control surfaces when multiple demands have been required. The FLC structure has been simultaneously tuned for multiple demands. The resulting rule base and membership function's shapes have been achieved by defining the scaling factors for each demand through the polynomial fitting. The purpose of such a tuning process was to improve the system performance with the intention to maintain the linguistic meaning of the fuzzy controller, which has been validated for each required demand. To the best of the author's knowledge this is the first reported fuzzy logic scheduled controller for multiple demands of a missile autopilot design in the literature.

In Chapter 5, multiple solutions were obtained simultaneously by using multi-objective optimisation of the fuzzy logic trajectory control parameters, allowing the system designer the flexibility of trading one solution against others. This has been achieved by evaluating four closed loop performance criteria which asses the quality of the side-slip velocity or lateral acceleration responses. An evolutionary algorithm has produced a set of results that populate the Pareto solution set by using Pareto based approach with non-dominated sorting. The main strengths of this approach are that it can handle any number of objectives independently and can take into account non-domination during the ranking process. An unique way has been proposed to incorporate preference information into the evolutionary multi-objective algorithm by using the Optimistic Reference Point approach to direct the GA-search towards specified areas for preferable optimal solutions on each objective simultaneously. The non-dominated sorting method has provided good performance, both in terms of inferior solutions and in terms of its coverage of the available non-dominated points. The preferability mechanism has helped the designer to implement the design requirements into the optimisation procedure.

In most multi-objective optimisation problems it is not clear what kind of preferences should be specified for each objective, whereas in this particular case the missile engineer is interested in achieving closed loop performance values within specified ranges in order for the missile to respond as fast as possible to guidance commands and be able to fly in supersonic regime. The determination of these ranges has been proposed by the author in two different ways: by using reference points (ideal, maximum and minimum values for each objective), and by handling the objectives as penalties based on the fuzzy logic membership functions principle. Both are different ways to incorporate preference information into the genetic algorithm optimisation process to direct the search towards feasible areas which satisfy specific values of the objectives. As we have used non-dominated sorting for forming the Pareto front, the concept of crisp preferability (using reference approach) has been seen to provide useful information in the optimisation process which would have otherwise been faced with sampling a very large trade-off surface (when all the

closed loop performance criteria have been defined in direct form). Whereas, if those objectives are defined as fuzzy constraints (penalties) this has reduced the size of the non-dominated set to search and it is certain that the final solutions at the end of the optimisation lie within the specified fuzzyfied ranges, so that the engineering design requirements are satisfied and no need of further DM is required. A priori DM was enough to predetermine desired feasible solutions at the final stage. Conversely in the first case, the evolutionary algorithm has searched in a larger feasible area for acceptable solutions, but only a subset of the non-dominated set has been of practical relevance, which required further interaction with the DM (i.e. a selective process at the end of the optimisation).

By using the approach of handling objectives as fuzzy constraints, it was much quicker to find solutions which satisfy the specified ranges on each objective. Although many duplicate solutions were produced, most important was that these solutions were all acceptable. We suggest that seeing the multi-objective optimisation problem as a multi-constraint one, may actually help designers to refine acceptable solutions first and then to investigate further on optimising each objective to specific values.

Evolutionary algorithms seems to gain popularity in the multi-objective optimisation world. In this thesis GAs have shown to be promising and reliable optimisators as well as to be useful decision making tool. Finding global optimal solutions was not the only consideration, as providing solutions with robust performance in the presence of uncertainties was equally important. In Chapter 5, genetic algorithms have been also successful in finding multiple solutions when a large range of different velocity demands have been required. Various solutions have been presented to show the powerful interpolative mechanism of the fuzzy scheduled controllers when multiple demands were required. The scaling factors of the fuzzy controller have been determined by using polynomial fitting for each demand. The optimisation procedure has been able to find multiple solutions (alternative fuzzy controllers) in terms of closed loop performance criteria and has been able to tune the control parameters (i.e. the membership functions and the rule base structure) simultaneously for multiple demands.

The conclusions were discussed here as a means of illustrating the role of the autopilot and the importance of designing it to meet specific requirements dictated by the guidance loop. This research work has achieved an elegant and efficient solution of designing a robust autopilot system. The evolutionary multi-objective algorithms have contributed for solving a real engineering problem such as finding the trajectory control parameters of a highly manoeuvrable missile system. To the knowledge of the author these results have not been shown in the literature before.

6.1 Future work

Important areas of work for further investigation can be identified such as:

An important issue is to find a way to simplify the derived nonlinear control law of the Feedback Linearization by using a well-trained neural network within the inner loop of the system. This will be useful to relieve the computational load of the nonlinear control law and to provide better robustness of the closed loop system.

A comparison of a Neuro Networks Feedback Linearization with Fuzzy Feedback Linearization can give us better understanding how intelligent systems behave and how do they differ in terms of performance, complexity and efficiency.

In this research all the objectives were treated independently, however if the missile engineer is interested in solutions that are fast with almost no overshoot, it is possible to alter the rule base structure of the fuzzy trajectory controller to produce desired solutions once the controller has been tuned. By studying in details the rule base structure, is possible to extract information about which rule affects the rise time, overshoot, settling time or steady state error.

Tuning the scaling factors for inputs and outputs of the fuzzy controller for each required demand can be given to the GAs as they should be able to handle this task easily.

Not many people have done interactive decision making combined with genetic algorithms. During the optimisation process, if the DM finds the candidate solutions unacceptable, DM can refine the preferences so to stimulate GAs to move on to a different region of the non-dominated set.

It is an open research area for developing new evolutionary algorithms using genetic programming strategies. Some comparisons with other existing multi-objective optimisation evolutionary Pareto based approaches like MOGA may be an interesting thing to do.

Finally, extend the multi-objective optimisation work to MIMO system.

6.2 Publications

1. White B.A., Blumel A.L. and Hughes E.J., "A Robust Fuzzy Autopilot Design using Multi-Criteria Optimization", International Journal of Fuzzy Systems, Vol2, No2, pp 129-138, 2000

2. Blumel A.L., Hughes E.J. and White B.A., "Fuzzy Autopilot Design Using A Multiobjective Evolutionary Algorithm", Congress on Evolutionary Computations, pp 54-62, july 16-19 2000, San-Diego, USA

3. Tsourdos A., Blumel A.L. and White B.A., "Flight control design for a missile.An approximate feedback linearization approach", pp 56-64, June 28-30 1999, IEEE Control Conference, Haifa-Israel

4. Blumel A.L., Hughes E.J. and White B.A., "Design of robust Fuzzy Controllers for Aerospace Applications", 18^{th} International Conference of the North American Fuzzy Information Processing Society, pp 438-442, June 10-12 1999, New-York, USA

5. Tsourdos A., Blumel A.L. and White B.A., "Autopilot design of a non-linear missile", *In Proceedings of IEE Control Conference*, pp 889-894, 1-4 September 1998, Swansea-UK

6. Tsourdos A., Blumel A., White B.A., "Tracking control design of a non-linear homing missile", *In Proceeding of International Symposium on Nonlinear Systems and Control in Aerospace Applications*, pp 118-124, 24-28 August 1998, Seoul-Korea

7. White B.A., Tsourdos A., Blumel A.L., "Lateral acceleration control Design of a Non-linear homing missile" ,*In Proceeding of International Symposium on Nonlinear System*, pp 708-713, 1-4 July 1998, Twente-Niderland

8. Blumel A. L. and White B.A., "Fuzzy non-linear control of a missile", Poster presentation, Fuzzy Summer School, 16-20 June 1998, Ferrara-Italy

9. Tsourdos A., Blumel A.L., White B.A., Non-linear Horton Missile Model: Internal report, RMCS, Cranfield University, September 1997

Appendix A

Nomenclature

A.1 Abbreviations

fig.	figure
FL	Feedback Linearization
SMC	Sliding Mode Control
VSCS	Variable structure control system
ROV	Remote operated vehicle
RCAM	research civil aircraft model
FLC	Fuzzy Logic Control
SISO	single-input single-output system
MIMO	multi-input multi-output system
DOF	degree of freedom
I/O	Input/Output
CLOS	command to line of sight
NNs	Neural Networks
CMAC	cerebellar model arithmetic computer

GAs	Genetic Algorithms
GGAP	generation gap
SF	scaling factor
VEGA	vector evaluated genetic algorithm
MOGA	multi-objective genetic algorithms
NSGA	non-dominated sorting genetic algorithm
DM	decision maker
MCDM	multi-criteria decision making
BTT	bank-to-turn motion
STT	skid-to-turn motion

A.2 Variables

Chapter 2

x	roll axis
y	pitch axis
z	yaw axis
X, Y, Z	forces
L, M, N	moments in eq 2.6
p	roll rate
q	pitch rate
r	yaw rate
U	velocity along the roll axis
v	velocity along the pitch axis
w	velocity along the yaw axis
V_o	total Velocity
η	elevator angle
ζ	rudder angle
ξ	aileron angle
α	pitch incidence
β	yaw incidence
σ	total incidence
λ	aerodynamic roll
I_y, I_z	lateral inertia
I_x	inertia
m	mass of the airframe
d	missile diameter
S	wing chord
M	Mach number
SoS	speed of sound
ρ	air density
x_{cg}	centre of gravity
X_{cp}	centre of pressure
x_f	fin moment arm
x_{sm}	static margin
x_{sf}	lateral moment arm
x_{sr}	roll moment arm

	$\dot{q}$ equation
m_q	pitch rate moment
m_w	pitch velocity moment
m_ξ	aileron control coupling moment
m_η	elevator control moment
C_{m_q}	damping moment coeff.
C_{m_w}	side-slip control moment coeff.
C_{m_ξ}	fin roll moment coeff.
C_{m_η}	fin side-slip moment coeff.

	$\dot{w}$ equation
z_w	pitch velocity force
z_q	pitch rate force
z_ξ	aileron control coupling force
z_η	elevator control force
C_{z_w}	side-slip normal force coeff.
C_{z_ξ}	fin roll force coeff.
C_{z_η}	fin normal force coeff.

	$\dot{r}$ equation
n_r	yaw rate moment
n_v	yaw velocity moment
n_ξ	aileron control coupling moment
n_ζ	rudder control moment
C_{n_r}	damping moment coeff.
C_{n_v}	side-slip control moments coeff.
C_{n_ξ}	fin roll moment coeff.
C_{n_ζ}	fin side-slip moment coeff.

	$\dot{v}$ equation
y_v	yaw velocity force
y_r	yaw rate force
y_ξ	aileron control coupling force
y_ζ	rudder control force
C_{y_v}	side-slip normal force coeff.
C_{y_ξ}	fin roll force coeff.
C_{y_ζ}	fin normal force coeff.

	$\dot{p}$ equation
l_p	roll rate moment
l_ζ	rudder control coupling moment
l_η	elevator control coupling moment
l_ξ	aileron control moment
C_{l_p}	damping moment coeff.
C_{l_ζ}	fin coupling moment coeff.
C_{l_η}	fin coupling moment coeff.
C_{l_ξ}	fin roll moment coeff.

$a_1, \ldots, a_5$	system parameters (see Appendix C)
$b_1, \ldots, b_7$	system parameters (see Appendix C)
$c_1, \ldots, c_4$	system parameters (see Appendix C)
$\triangle a_1, \ldots, \triangle a_5$	change in system parameters (see Section 2.8.1)
$\triangle b_1, \ldots, \triangle b_7$	change in system parameters (see Section 2.8.1)
$\triangle c_1, \ldots, \triangle c_4$	change in system parameters (see Section 2.8.1)
f_v, f_w, f_r, f_q, f_p	non-linear functions of a non-linear system
g_v, g_w, g_r, g_q, g_p	non-linear functions related with the input
z	flight direction used for Polar coordinates
λ	flight angle of orientation used for Polar coordinates

Chapter 3

x	state variable
y	output of a non-linear system
f(x),g(x)	smooth vector fields which have continuous partial derivatives of any required order
h(x)	smooth scalar function of the state x which is the output (i.e. y) of a non-linear system
∇h	gradient of h
$L_f h(x)$	Lie derivative of h with respect to f
$L_g h(x)$	Lie derivative of h with respect to g
ν	the new input to the linearized system
$\alpha(x)$	non-linear state feedback
$\beta(x)$	non-linear state feedback related to the input
E	decoupling matrix of a MIMO system
u	static state feedback for decoupled closed loop behaviour
r	relative degree of a non-linear system
$e, \dot{e}, \ldots, e^{n-1}$	describe the closed loop error dynamics
$k_0, k_1, \ldots, k_{n-1}$	coefficients of a Hurwitz polynomial
$\phi_1(x), \ldots, \phi_i(x)$	series of functions related to h(x)
$\mu_1, \ldots, \mu_r$	a variable which is used to transform a non-linear system into new coordinates
$\psi_1(\mu, u), \ldots, \psi_{n-r}(\mu, u)$	internal dynamics of a non-linear system
$\psi_1(x_1, u_1)$	the neglected term of the approximate feedback linearization (see eq.3.43, eq.3.57, eq.3.64)
$\psi_2(x_2, u_2)$	the neglected term of the approximate feedback linearization (see eq.3.66)
α	lateral acceleration
$\bar{\alpha}$	augmented acceleration
z	flight direction used for Polar coordinates
λ	flight angle of orientation used for Polar coordinates

Chapter 4

μ_A	a membership function of fuzzy set A
μ_B	a membership function of fuzzy set B
A,B	fuzzy sets of the inputs of a fuzzy system
O	fuzzy set of the output of a fuzzy system
U	crisp set called the Universe of Discourse
S_e	scaling factor for error input variable
S_{de}	scaling factor for derivative of error input variable
S_u	scaling factor for control output variable
y^o	defuzzyfied output of a fuzzy system
W^l	degree of fulfillment of the l^{th} rule
A_i	membership function i of input A
B_j	membership function j of input B
O_k	membership function k of output O
O	objective function
O_1	steady state error objective function
O_2	overshoot objective function
O_3	rise time objective function
O_4	settling time objective function
$\triangle f, \triangle g$	uncertainties in the model caused by aerodynamic coefficients
SC_{v-er}	scaling factor of the error input of the normalised fuzzy controller
SC_{v-erd}	scaling factor of the derivative of error input of the normalised fuzzy controller
SC_{out}	scaling factor of the output of the normalised fuzzy controller
$p_0, \ldots, p_3$	coefficients of SC_{v-erd} polynomial in 4.4
q_0, q_1	coefficients of SC_{out} polynomial in 4.5
$b_0, \ldots, b_4$	coefficients of SC_{v-er} polynomial in 4.6
$c_0, \ldots, c_3$	coefficients of SC_{v-erd} polynomial in 4.7
$d_0, \ldots, d_4$	coefficients of SC_{out} polynomial in 4.8

Chapter 5

x	vector of decision variables
$g_i(x)$	inequality constraints
$h_i(x)$	equality constraints
$f(x)$	objective vector function
x^*	vector of decision variables which yields the optimal values of all obj. functions
f^*	ideal vector in the objective space
σ_{share}	sharing value
$\eta_j(x)$	function of losses
y_i^*	ideal objective value
$y_{i_{max}}$	maximum objective value required a priori by DM
$y_{i_{min}}$	minimum objective value required a priori by DM
e_r	steady state error
OS	overshoot
t_r	rise time
t_s	settling time
Er^*	ideal reference point of $\eta_1(x)$ objective function
Os^*	ideal reference point of $\eta_2(x)$ objective function
Tr^*	ideal reference point of $\eta_3(x)$ objective function
Ts^*	ideal reference point of $\eta_4(x)$ objective function
Er_{max}	maximum reference point of $\eta_1(x)$ objective function
Os_{max}	maximum reference point of $\eta_2(x)$ objective function
Tr_{max}	maximum reference point of $\eta_3(x)$ objective function
Ts_{max}	maximum reference point of $\eta_4(x)$ objective function
Er_{min}	minimum reference point of $\eta_1(x)$ objective function
Os_{min}	minimum reference point of $\eta_2(x)$ objective function
Tr_{min}	minimum reference point of $\eta_3(x)$ objective function
Ts_{min}	minimum reference point of $\eta_4(x)$ objective function
m	number of evaluated individuals in a population
$A_1, \ldots, A_m$	alternative solutions
x_{ij}	attributes which measure alternative performance
$Chrom_i$	chromosomes to be evaluated in a generation
Er_i, Tr_i, Ts_i, Os_i	closed loop performance criteria with which alternative solutions are evaluated
μ_{OS}	overshoot penalty
μ_{T_r}	rise time penalty
μ_{T_s}	settling time penalty
$[C_{yv_{min}}\ C_{y\zeta_{min}}\ C_{nr_{max}}\ X_{cp_{max}}]$	vertex points model
$[C_{yv_{max}}\ C_{y\zeta_{min}}\ C_{nr_{max}}\ X_{cp_{min}}]$	vertex points model

Appendix B

Physical Parameters of Horton Missile Model

Symbol	Meaning	Value
ρ_0	Sea Level Air density	$1.23kg/m^3$
ρ	Air Density	$\rho_0 - 0.094h$
d	Reference diameter	$0.2m$
S	Reference area	$d^2/4 = 0.0314m^2$
m	Mass	$125kg$
I_z, I_y	*Lateral Inertia*	$67.5kgm^2$
I_x	*Inertia*	$6.75kgm^2$

Appendix C

Non-linear functions of the state space model

The non-linear functions f_v, f_w, f_r, f_q, f_p and g_v, g_w, g_r, g_q, g_p from the equation of motion (2.42) shown in Chapter 2 are given here:

$$
\begin{aligned}
f_v(v,w,r) &= V^o(\bar{C}_{yv_0} + \bar{C}_{yv_\sigma}\sqrt{v^2+w^2})v - Ur \\
g_v(v,w) &= V^o V_o(\bar{C}_{y\zeta_0} + \bar{C}_{y\zeta_\sigma}\sqrt{v^2+w^2} \\
f_r(w,v,r) &= -R_o\frac{1}{d}[(\bar{x}_{cp_0}\bar{C}_{yv_0} + \bar{x}_{cp_\sigma}\bar{C}_{yv_0})\sqrt{v^2+w^2} \\
&\quad +(\bar{x}_{cp_0}\bar{C}_{yv_\sigma} + \bar{x}_{cp_\sigma}\bar{C}_{yv_\sigma}(v^2+w^2)]v + \frac{d}{2}R_o(\bar{C}_{nr_0} + \bar{C}_{nr_\sigma}\sqrt{v^2+w^2})r \\
g_{r_\zeta}(v,w) &= R_o S_f V_o(\bar{C}_{y\zeta_0} + \bar{C}_{y\zeta_\sigma}\sqrt{v^2+w^2}) \\
g_{r_\xi}(v,w) &= -R_o S_f V_o[(\bar{C}_{y\zeta_0}R_{N_2} + \bar{C}_{y\zeta_\sigma}R_{N_1})\sqrt{v^2+w^2} + \bar{C}_{y\zeta_0}R_{N_1} + \bar{C}_{y\zeta_\sigma}R_{N_2}(v^2+w^2)] \\
f_w(v,w,q) &= W^o(\bar{C}_{zw_0}w + \bar{C}_{zw_\sigma}\sqrt{v^2+w^2}) + Uq \\
g_w(v,w) &= W^o V_o(\bar{C}_{z\eta_0} + \bar{C}_{z\eta_\sigma}\sqrt{v^2+w^2}) \\
f_q(v,w,q) &= Q_o[\frac{d}{2}\bar{C}_{mq_0}q + \frac{1}{2}dC_{mq_\sigma}\sqrt{v^2+w^2}q + \frac{1}{d}(\bar{x}_{cp_0}C_{zw_0}w) \\
&\quad +\frac{1}{d}(\bar{x}_{cp_\sigma}\bar{C}_{zw_0} + \bar{x}_{cp_0}\bar{C}_{zw_\sigma}w\sqrt{v^2+w^2} + \frac{1}{d}(\bar{x}_{cp_\sigma}\bar{C}_{zw_\sigma}w(v^2+w^2))] \\
g_{q_\eta}(v,w) &= -Q_o V_o S_f(\bar{C}_{z\eta_0} + \bar{C}_{z\eta_\sigma}\sqrt{v^2+w^2}) \\
g_{q_\xi}(v,w) &= Q_o S_f V_o[(\bar{C}_{z\eta_0}R_{N_2} + \bar{C}_{z\eta_\sigma}R_{N_1})\sqrt{v^2+w^2} + \bar{C}_{z\eta_0}R_{N_1} + \bar{C}_{z\eta_\sigma}R_{N_2}(v^2+w^2)] \\
f_p(p) &= -\frac{100}{I_x}p \\
g_{p_\eta}(v,w) &= \frac{500R_{L_1}}{I_x} + \frac{R_{L_2}}{I_x}\frac{180}{V_o 3.14}\sqrt{v^2+w^2} \\
g_{p_\zeta}(v,w) &= \frac{500R_{L_1}}{I_x} + \frac{R_{L_2}}{I_x}\frac{180}{V_o 3.14}\sqrt{v^2+w^2} \\
g_{p_\xi} &= -\frac{500}{I_x}
\end{aligned}
\tag{C.1}
$$

where the Mach number M, and the total velocity V_o are slowly varying and where:

$$\begin{aligned} M &= \frac{V_o}{SoS} \\ V^o &= W^o = \frac{1}{2m}\rho V_o S \\ Q_o &= R_o = \frac{1}{2I_{yz}}\rho V_o S d \end{aligned} \tag{C.2}$$

The parameters $a_1, \ldots, a_5$, $b_1, \ldots, b_{10}$ and $c_1, \ldots, c_4$ are defined here which are used in Chapter 3 to describe the non-linear system in parametric format:

$$\begin{aligned} a_1 &= V^o \bar{C}_{yv_0} \\ a_2 &= V^o \bar{C}_{yv_\sigma} \\ a_3 &= -U \\ a_4 &= V^o V_o \bar{C}_{y\zeta_0} \\ a_5 &= V^o V_o \bar{C}_{y\zeta_\sigma} \\ b_1 &= -R_o \frac{1}{d} \bar{x}_{cp_\sigma} \bar{C}_{yv_\sigma} \\ b_2 &= -R_o \frac{1}{d} (\bar{x}_{cp_0} \bar{C}_{yv_0} + \bar{x}_{cp_\sigma} \bar{C}_{yv_0}) \\ b_3 &= -R_o \frac{1}{d} \bar{x}_{cp_0} \bar{C}_{yv_\sigma} \\ b_4 &= \frac{d}{2} R_o \bar{C}_{nr_\sigma} \\ b_5 &= \frac{d}{2} R_o \bar{C}_{nr_0} \\ b_6 &= R_o S_f V_o \bar{C}_{y\zeta_\sigma} \\ b_7 &= R_o S_f V_o \bar{C}_{y\zeta_0} \\ b_8 &= -R_o S_f V_o (\bar{C}_{y\zeta_0} R_{N_2} + \bar{C}_{y\zeta_\sigma} R_{N_1} \\ b_9 &= -R_o S_f V_o \bar{C}_{y\zeta_0} R_{N_1} \\ b_{10} &= -R_o S_f V_o \bar{C}_{y\zeta_\sigma} R_{N_2} \\ c_1 &= -\frac{100}{I_x} \\ c_2 &= -\frac{500}{I_x} \\ c_3 &= \frac{500 R_{L_1}}{I_x} + \frac{R_{L_2}}{I_x} \frac{180}{V_o \pi} \\ c_4 &= \frac{500 R_{L_1}}{I_x} + \frac{R_{L_2}}{I_x} \frac{180}{V_o \pi} \end{aligned} \tag{C.3}$$

Appendix D

Lie Algebra

D.1 Lie Derivative and Lie Bracket

The terminology used in differential geometry has been introduced here by describing the following mathematical tools: A vector function $\mathbf{f} : \mathbf{R}^n \longrightarrow \mathbf{R}^n$ is called a vector field in $\mathbf{R}^n$ which means to every vector function $\mathbf{f}$ corresponds a field of vectors in an n dimensional space. A smooth vector field is a function $\mathbf{f(x)}$ which has continuous partial derivatives of any required order. Given a smooth scalar function h($\mathbf{x}$) of the state $\mathbf{x}$, the gradient of h is denoted by ∇h

$$\nabla h = \frac{\partial h}{\partial \mathbf{x}} \tag{D.1}$$

The gradient is represented by a *row-vector* of elements $(\nabla h)_j = \frac{\partial h}{\partial x_j}$. Similarly, given a vector field $\mathbf{f(x)}$, the Jacobian of $\mathbf{f}$ is denoted by $\nabla \mathbf{f}$:

$$\nabla \mathbf{f} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \tag{D.2}$$

and it is represented by an $n \times n$ matrix of elements $(\nabla \mathbf{f})_{ij} = \frac{\partial f_i}{\partial x_j}$.

Lie Derivatives
Given a scalar function h($\mathbf{x}$) and a vector field $\mathbf{f(x)}$, a new scalar function $L_{\mathbf{f}}h$ is defined, called the Lie derivative of h with respect to $\mathbf{f}$. By definition [14] the Lie derivative of h with respect to $\mathbf{f}$ is a scalar function defined by $L_{\mathbf{f}}h(x) = \nabla h \mathbf{f}(x)$. Thus the Lie derivative $L_{\mathbf{f}}h$ is the directional derivative of h along the direction of the vector $\mathbf{f}$. First and higher order Lie derivatives can be defined as:

$$L_f h(x) = \frac{\partial h}{\partial x} f(x) \tag{D.3}$$

and respectively:

$$L_f^k h(x) = L_f(L_f^{k-1} h(x)) \tag{D.4}$$

Similarly, if $\mathbf{g}$ is another vector field, then the scalar function $L_{\mathbf{g}}L_{\mathbf{f}}h(\mathbf{x})$ is given by:

$$L_{\mathbf{g}}L_{\mathbf{f}}h(x) = \nabla L_{\mathbf{f}}h(x)\mathbf{g} \tag{D.5}$$

Lie Brackets
By definition the Lie bracket of **f** and **g** is a third vector field defined by:

$$[\mathbf{f}, \mathbf{g}[= \nabla\mathbf{g}\mathbf{f} - \nabla\mathbf{f}\mathbf{g} \tag{D.6}$$

where **f** and **g** are two vector fields on $\mathbf{R}^n$ and where $\nabla\mathbf{g}$ and $\nabla\mathbf{f}$ represent the Jacobians (matrices) of g and f respectively.

Successive Lie brackets $[f, ..., [f, g], ...]$ can be defined as follows:

$$\begin{aligned} ad_f^0(g) &= g \\ ad_f^1(g) &= [f, g] \\ &\vdots \\ ad_f^k(g) &= [f, ad_f^{k-1}(g)] \end{aligned} \tag{D.7}$$

where "ad" represents "adjoint". Both Lie Derivatives and Lie Brackets are the main mathematical tools used by Feedback Linearization for nonlinear dynamic systems.

D.2 Feedback linearization of MIMO systems

Input-Output linearization of MIMO systems [14] is obtained similarly to the SISO case, by differentiating the outputs y_i until the inputs appear. Assume that r_i is the smallest integer such that at least one of the inputs appears in $y_i^{(r_i)}$, then

$$y_i^{(r_i)} = L_f^{r_i} h_i + \sum_{j=1}^{m} L_{g_j} L_f^{r_i - 1} h_i u_j \tag{D.8}$$

with $L_{g_j} L_f^{r_i-1} h_i(x) \neq 0$ for at least one j, in a neighbourhood Ω_i of the point x_0. Performing the above procedure for each output y_i yields

$$\begin{bmatrix} y_1^{r_1} \\ \vdots \\ \vdots \\ y_m^{r_m} \end{bmatrix} = \begin{bmatrix} L_f^{r_1} h_1(x) \\ \vdots \\ \vdots \\ L_f^{r_m} h_m(x) \end{bmatrix} + E(x)u \tag{D.9}$$

where the (m x m) matrix $E(x)$ is defined as

$$E(x) = \begin{bmatrix} L_{g_1} L_f^k h_1(x) \dots L_{g_m} L_f^k h_1(x) \\ \vdots \dots \vdots \\ L_{g_1} L_f^k h_m(x) \dots L_{g_m} L_f^k h_m(x) \end{bmatrix} \tag{D.10}$$

where $0 \leq k \leq 2n - 1$. Define then Ω as the intersection of the Ω_i. If as assumed above , the partial *relative degrees* r_i are all well defined, then Ω is itself a finite

neighbourhood of x_0. Furthermore, if $E(x)$ is invertible over the region Ω, then similarly to the SISO case the input transformation

$$u = E^{-1} \begin{bmatrix} \nu_1 - L_f^{r_1} h_1 \\ \vdots \\ \vdots \\ \nu_m - L_f^{r_m} h_m \end{bmatrix} \tag{D.11}$$

yields m equations of the simple form

$$y_i^{r_i} = \nu_i \tag{D.12}$$

Since the input ν_i only affects the output y_i, (D.11) is called a decoupling control law, and the invertible matrix $E(x)$ is called the decoupling matrix of the system. The system is then said to have relative degree $(r_1, \ldots, r_m)$ at x_0 and the scalar $r = r_1 + \ldots + r_m$ is called the total relative degree of the system at x_0.

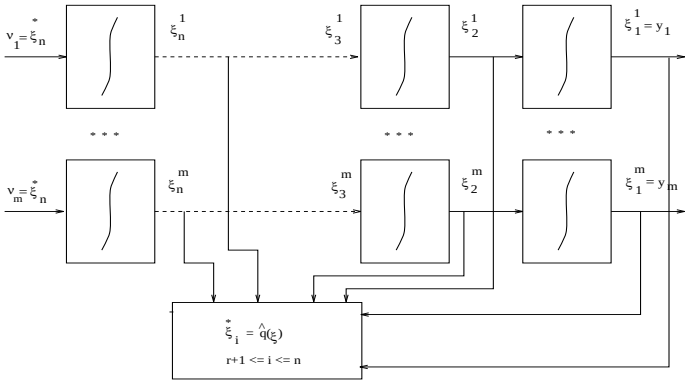


Figure D.1: MIMO control design

An interesting case is when the relative degree of a system is the same as its order, i.e., when the output y has to be differentiated n times (with n being the system order) to obtain a linear input-output relation. In this case, the variables $y, y^1, \ldots, y^{r-1}$ may be used as a new set of state variables for the system, and there is no internal dynamics associated with this input-output linearization. In this case the *Input-Output linearization* leads to *Input-State linearization* by Slotine in [14].

D.3 Complex Nonlinear Derivative

$$\dot{(a_2 x_1 \sqrt{{x_1}^2 + {x_3}^2})} = a_2 \dot{x}_1 \sqrt{{x_1}^2 + {x_3}^2} + a_2 \frac{x_1^2 \dot{x}_1}{\sqrt{{x_1}^2 + {x_3}^2}} + a_2 \frac{x_1 x_3 \dot{x}_3}{\sqrt{{x_1}^2 + {x_3}^2}} \tag{D.13}$$

Such a complex nonlinear derivative has been used for the nonlinear feedback control low derived in MIMO case **Design1**

Appendix E

Fuzzy logic glossary

Aristotle's law of "*A or not A*" always holds in probability, while Buddha's law "*A and not A*" holds in real life situations. The fuzzy entropy theorem has been proved in 1986: "Fuzziness is the ratio of Buddha over Aristotle". The fuzzier the set the more Buddha resembles Aristotle". Fuzzy Logic has two meanings. The first meaning is multi-valued or 'vague' logic. Everything is a matter of degree including truth and set membership. The second meaning is reasoning with fuzzy sets or with sets of fuzzy rules. This dates back to the first work on fuzzy sets in the 1960s and 1970s by Lofti Zadeh at the University of California at Berkeley. Other synonyms: gray logic, cloudy logic, continuous logic.

Fuzzy Rule

A conditional of the form IF X is A,THEN Y is B. A and B are fuzzy sets:"IF the room air is *cool*, THEN set the motor speed to *slow*". In math terms a rule is a relation between fuzzy sets. Each rule defines a fuzzy patch (the product A x B) in the system "state space"- the set of all possible combinations of inputs and outputs. The wider the fuzzy sets A and B, the wider and more uncertain the fuzzy patch. More certain knowledge leads to smaller patches or more precise rules. Fuzzy rules are the knowledge building blocks in a fuzzy system. In math terms each fuzzy rule acts as an associative memory that associates the fuzzy response B with the fuzzy stimulus A. Then stimuli similar to A map to responses similar to B. In this sense each fuzzy rule defines fuzzy associative memory, or FAM. A set of FAM rules in a fuzzy system acts as a FAM at higher level. It too converts similar inputs to similar outputs.

Fuzzy set

A set whose members belong to it to some degree. In contrast a standard or non-fuzzy set contains its members all or none. The set of even numbers has no fuzzy members. Each number belongs to it 0% or 100% . The set of big molecules has graded membership. Some molecules are bigger than others and so belong to it to greater degree. In the same way most properties like redness or tallness or goodness admit degrees and thus define fuzzy sets. In math term a fuzzy set is either a point in a hypercube or a curve. A fuzzy set with n members is equal to a list of n numbers

or fit values. Each fit value lies in the interval from 0 to 1 and stands for the degree that that member belongs to or fits in the fuzzy set. The set of all such lists of n fit values defines a solid unit hypercube of n dimensions (with 2^n corners made up of the 2^n binary lists of 0s and 1s or the 2^n non-fuzzy sets). Each fuzzy set is one point in this fuzzy cube. The same holds as the number n grows to infinity. The same holds as the number n grows to infinity. Three tall men (0.90.50.3) means the first is 90% tall, the other is 50% tall or as much not-tall as he is tall.,the third man is 30% tall or more not tall than tall. A curve defines a fuzzy set for a continuum of cases like all possible temperature values between 50° and 100° or all possible car velocities between 0 mph and 120 mph. The height of the curve between 0 and 1 measures the fit value or degree that the element belongs to the fuzzy set. A non-fuzzy set looks like a step. Part of the curve is the flat line at 100% and the rest is the flat line at 0%. In this world continuity is a useful fiction for math analysis and for engineering design. Up close there are only discrete values and a finite and small set to temperature values or even car velocities. This amounts to "sampling" a fuzzy curve at several places and gives a finite fit list for the fuzzy set. The more samples the more accurate the fit list and the larger the dimension of the hypercube in which it sits as a point.

Fuzzy system

A set of fuzzy rules that converts inputs to outputs. In the simplest case an expert states the rules in words or symbols. In the more complex case a neural system learns the rules from data or from watching the behaviour of human experts. Each input to the fuzzy system fires all the rules to some degree as in a massive associative memory. The closer the input matches the if-part of a fuzzy rule, the more the then part fires. The fuzzy system adds up all these output or then part fuzzy sets and takes their average or centroid value. The centroid is the output of the fuzzy system. Fuzzy chips perform this associative mapping form input to output thousands or millions of times per second. Each map form input to output defines one FLIPS- or fuzzy logical inferences per second. The Fuzzy Approximation Theorem shows that a fuzzy system can model any continuous system. Each rule of the fuzzy system acts as a fuzzy patch that the system places so as to resemble the response of the continuous system to all possible inputs.

Probability

The mathematical theory of chance. A probability is a number assigned to an event. The larger the number the more "likely" the event will occur. In probability theory all uncertainty comes from an undefined "randomness" or "chance".
In math terms all probability numbers must add up to one. All events are bivalent. Either an event happens or not, in which case its opposite happens. The probability that either the event A happens or its opposite not-A happens is 100%. Events in probability theory are just the black-white sets of set theory. In this sense probability theory rests on bivalent logic.

Appendix F

Software implementation

The software is build under Matlab environment using "C" language. The dynamics of the missile is written in "C" as S-function and the non-linear control law is written in "C" as mex function. Both are compiled under Matlab. The simulation of the autopilot system is designed using Simulink library and it is called from the Matlab workspace. The Fuzzy logic trajectory controller is produced by using the Fuzzy Logic toolbox in Matlab language. The genetic algorithm strategy is written with the help of the GA functions provided by Shefield University, UK. The simulations were mainly ran on a 300MHz Unix Workstation or PC. The optimisation algorithm is used to generate the fuzzy control parameters. The obtained fuzzy controller is then tested on a missile model. A performance analysis is done off-line for each autopilot simulation. The maximum objective value is returned to the optimisation algorithm for evaluation of the tested fuzzy controller. The optimisation process repeats for large number of iterations until satisfactory closed loop performance of the autopilot system is obtained.

Bibliography

[1] Shamma J.S. and Athans M. Analysis of nonlinear gain scheduled control systems. *IEEE Transactions on Automatic Control*, 35:898–907, 1990.

[2] Isidori A., Krener A.J., Gori-Georgi C., and Monaco C. Nonlinear decoupling via feedback: A differential-geometric approach. *IEEE Transactions on Automatic Control*, 26:331–345, 1981.

[3] Hunt L.R. and Su R. Linear equivalents of nonlinear time-varying systems. In *Proceedings of International Symposium on Mathematics Theory of New Technology and Systems*, pages 119–123, 1981. Santa Monica, CA.

[4] Su R. On the linear equvalents of nonlinear systems. *Systems and Control Letters*, 2:48–52, 1982.

[5] Marino R. and Spong M.W. Nonlinear control techniques for flexible joint manipulators: A single link case study. In *IEEE Proceedings on International Conference on Robotics and Automatics*, pages 1026–1030, 1986. San Francisko, CA.

[6] Wang B. and Vidyasagar M. Control of a class of manipulators with a single flexible link. Part I: Feedback Linearization. *ASME Journal of Dynamics Systems, Measurement and Control*, 4:655–661, 1991.

[7] Hahn H., Piepenbrink A., and Leimbach K.D. Input-output linearization control of an electro servo-hydraulic actuator. In *IEEE Proceedings on Control Applications*, volume 2, pages 995–1000, 1994.

[8] Bezick S., Rusnak I., and Gray W.S. Guidance of a homing missile via nonlinear geometric control methods. *Journal of Guidance, Control and Dynamics*, 18(3):441–448, May–June 1995.

[9] Tahk M., Briggs M.M., and Menon P.K.A. Application of plant inversion via state feedback to missile autopilot design. In *Proceedings of the* 24^{th} *CDC*, pages 730–735, December 1988. Austin, Texas.

[10] Wee L-B. Nonlinear autopilot design for bank-to-bank sterring of a flight vehicle. *Journal of Guidance, Control and Dynamics*, 19(4):978–989, 1996.

[11] Utkin V.I. *Sliding Modes and Their Application in Variable Structure Systems.* MIR, Moskow, 1978.

[12] White B.A. and Silson P.M. Reachability in variable structure control systems. *IEE Proceedings*, 131(3):85–91, 1984.

[13] Zinober A.S.I. *Deterministic Control of Uncertain Systems.* Peter Peregrinus Ltd., 1990.

[14] Slotine Jean-Jacques.E. and Li W. *Applied Nonlinear Control.* Prentice Hall, 1991.

[15] Edwards C. and Spurgeon S.K. *Sliding Mode Control.* Taylor and Francis, 1998.

[16] Singh S. Asymptotically decoupled discontinuous control of systems and nonlinear aircraft manoeuvre. *IEEE Transaction on Aerospace and Electronic Systems*, 25(3):380–391, May 1989.

[17] McGookin E.W., Murray-Smith D.J., Li Y., and Fossen T.I. Parameter optimisation of a nonlinear tanker control s ystem using genetic algorithms. In *Proceedings of the* 2^{nd} *IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems Innovations and Applications*, 1997.

[18] Trebi-Ollennu A. and White B.A. Multiobjective fuzzy genetic algorithm optimization approach to non-linear control system design. *IEE Proceedings in Control Theory and Applications*, 144(2):137–142, 1997.

[19] Singh S. and Iyer A. Nonlinear decoupling sliding mode control and attitude control of spacecraft. *IEEE Transaction on Aerospace and Electronic Systems*, 25(5):621–633, September 1989.

[20] Thukral A. and Innocenti M. A sliding mode missile pitch autopilot synthesis for high angle of attack manoeuvring. *IEEE Transactions on Control Systems Technology*, 6:359–371, 1998.

[21] Weil R.D. and Wise K.A. Blended aero and reaction jet missile autopilot design using variable structure control techniques. In *IEEE* 30^{th} *Conference on Decision and Control*, pages 2828–2829, 1991.

[22] Hedrick J.K. and Gopalswamy S. Nonlinear flight control design via sliding methods. *Journal of Guidance, Control and Dynamics*, 13:850–858, 1990.

[23] Fossen T. and Sagatun S. Adaptive control of nonlinear systems: A case study of underwater robotic systems. *Journal of Robotic Systems*, 8(2):393–412, 1991.

[24] Trebi-Ollennu A. *Robust Nonlinear Control Designs Using Adaptive Fuzzy Systems.* PhD thesis, Cranfield University, 1996.

[25] Kanellakopoulos I., Kristic M., and Kokotovic P. *Nonlinear and Adaptive Control Design*. John Wiley and Sons Inc., 1995.

[26] Fossen T. and Svein B. Nonlinear vectorial backstepping design for global exponential tracking of marine vessels in the presence of actuator dynamics. In *IEEE Proceedings on Decision and Control Conference'97*, pages 4237–4242, 1997.

[27] Rios-Bolivar M., Zinober A.S., and Ramirez H. Adaptive sliding mode output tracking via backstepping for uncertain nonlinear system. In *Proceeding of the* 3^{rd} *European Control Conference*, pages 699–704, 1995.

[28] Chanho Song and Seung-Hwan Kim. A robust adaptive non-linear control approach to missile autopilot design. In 14^{th} *IFAC Simposium on Automatic Control in Aerospace*, pages 55–58, 1998. Seoul-Korea.

[29] Rios-Bolivar M., Ramirez H., and Zinober A.S.I. Output tracking control via adaptive Input-Output Linearization: A backstepping approach. In *Proceeding of the* 34^{th} *Conference on Decision and Control*, pages 1579–1584, New Orleans,LA, 1995.

[30] Aslaug G. and Fossen T. Nonlinear control of dynamic positioned ships using only position feedback: An observer backstepping approach. *IEEE Transactions on Control System Technology*, 6(1):121–128, 1998.

[31] Linkens D.A. and Nyongesa H.O. Learning systems in intelligent control: An appraisal of fuzzy neural and genetic algorithm control applications. *IEEE Transactions on Control Theory Applications*, 143(4):367–386, April 1996.

[32] Zadeh L. A. Fuzzy sets. *Information and Control*, pages 338–353, 1965.

[33] Lee C.C. Fuzzy logic in control systems: Fuzzy logic controller part one and two. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):404–435, 1990.

[34] Holmblad L.P. and Ostergaard J. Control of a cement kiln by fuzzy logic. *Fuzzy Information and Decision Processes*, pages 389–399, 1982.

[35] Driankov D., Hellendoorn H., and Reinfrank M. *An Introduction to Fuzzy Control*. Springer Verlag, 1993.

[36] Flight Mech. Group 08. Robust flight control design challenge problem formulation and manual: The research civil aircraft model RCAM. Technical report, GARTEUR FM(AG08), January 1996.

[37] Schram G. A fuzzy logic control approach. Technical report, GARTEUR group, Control laboratory, Department of Electrical Engineering, Delft University of Technology, The Netherlands, April 4 1997.

[38] Schram G., Copinga G.J.C., Bruijn P.M., and Verbruggen H.B. Failure-tolerant control of aircraft: Fuzzy logic approach. In *Proceedings of the American Control Conference'98*, 1998.

[39] Schroeder W. and Liu K. An appropriate application of fuzzy logic: A missile autopilot for dual control implementation. In *IEEE Proceedings in Intelligent Control*, pages 93–98, 1994.

[40] Chiu S., Chand S., Moore D., and Chaudhary A. Fuzzy logic for control of roll and moment for a flexible wing aircraft. In *IEEE Proceedings in Control Systems*, pages 41–48, 1991.

[41] Hunt K., Sbarbaro D., Żbikowski R., and Gawthrop P. Neural networks for control systems - A survey. *Automatica*, 28(6):1083–1112, 1992.

[42] Greenfield S. *The Human Mind Explained.* Marshall Editions Developments Ltd., 1996.

[43] Sanner R. and Slotine J. Gaussian network for direct adaptive control. *IEEE Transactions on Neural Networks*, 3(6):837–863, 1992.

[44] Collins D. and Dror S. Application of neural network to reconfiguration of damaged aircraft. *AGARD Advances in Soft Computing Technologies and Application in Mission Systems*, 7:1–15, 1997.

[45] Widrow B., Gupta N.K., and Maitra S. Learning with a critic in adaptive threshold systems. *IEEE Transaction on Systems Man and Cybernetics*, 3:455–465, 1973.

[46] McKelvey T. Neural networks applied to optimal flight control. In *Proceedings on IFAC Symposium on Artificial Intelligence in Real-Time Control*, pages 43–47, Delft, The Netherlands, 1992.

[47] Balakrishnan S. and Biega V. Adaptive critic based neural networks for aircraft optimal control. *Journal of Guidance, Control and Dynamics*, 19:893–898, 1996.

[48] Napolitano M. and Kincheloe M. On-line learning neural network controllers for autopilot systems. *Journal of Guidance, Control and Dynamics*, 33(6):1008–1015, 1995.

[49] McDowell D.M., Irwin G.W., Lightbody G., and McConnell G. Hybrid neural adaptive control for bank-to-turn missiles. *IEEE Transactions on Control Systems Technology*, 5(3):297–308, 1997.

[50] Steinberg M. Potential role of neural networks and fuzzy logic in flight control design and development. In *AIAA-92-0999, Aerospace Design Conference, Irvine,CA*, 1992.

[51] Steinberg M. Towards intelligent flight control. *IEEE Transactions on Systems, Man and Cybernetics*, 23(6):1699–1717, 1993.

[52] Linse D. and Stengel R. Identification of aerodynamic coefficients using computational neural networks. In *Proceedings of the* 30^{th} *AIAA Aerospace Sciences Meeting*, 1992.

[53] Sugeno M., Griffin M.F., and Bastian A. A fuzzy hierarchical control of an unmanned helicopter. In 5^{th} *IFSA World Congress*, pages 179–182, 1993.

[54] Berenji H., Lea R., Jani Y., and Khedar P. Space shuttle attitude control by reinforcement learning and fuzzy logic. In 2^{nd} *IEEE Conference on Fuzzy Systems*, pages 1396–1401, 1993.

[55] Napolitano M., Chen C., and Naylor S. Aircraft failure detection and identification using neural networks. *Journal of Guidance, Control and Dynamics*, 16(6):999–1009, 1993.

[56] Hariri A. and Malik O.P. A fuzzy logic based power system stabilizer with learning ability. *IEEE Transaction on Energy Conversion*, 11(4):721–727, 1996.

[57] Shin C. and Vishnupad P. Neuro-fuzzy control of complex manufacturing processes. *International Journal of Production Research*, 34(12):3291–3309, 1996.

[58] Geng and McCullough C. Missile control using fuzzy cerebellar model arithmetic computer neural networks. *Journal of Guidance, Control and Dynamics*, 20(3):557–565, 1997.

[59] Buckley J. and Hayashi Y. Hybrid fuzzy neural nets are universal approximators. In *IEEE Proceedings on Fuzzy Systems*, pages 238–243, 1994.

[60] KrishnaKumar K., Gonsalves P., Satyadas A., and Zacharias G. Hybrid fuzzy logic flight controller synthesis via pilot modelling. *Journal of Guidance, Control and Dynamics*, 18:1098–1105, 1995.

[61] Cao Y.J., Cheng S.J., and Wu Q.H. Sliding mode control of nonlinear systems using neural network. In *IEE Proceedings Control Conference 94'*, pages 855–859, 1994.

[62] Qin H., Mei S., Zhu Q., and Hong Y. Robust attraction for mimo affine nonlinear uncertain systems. In *Proceedings of* 13^{th} *IFAC World Congress*, pages 139–144, 1996.

[63] Chiou J.C., Hwang Ming-Chan, and Wu Shuess-Der. Robust control of nonlinear dynamic systems using sliding mode control and productive networks. In *Proceedings of the American Control Conference' 95*, pages 4231–4235, 1995.

[64] Fu Li-Chen, Chang Wei-Der, Yang Jung-Hua, and Kuo Te-Son. Adaptive robust bank to turn missile autopilot design using neural networks. *Journal of Guidance Control and Dynamics*, 20(2):346–354, 1997.

[65] Palm R. Sliding mode fuzzy control. In *IEEE Proceedings on Fuzzy Systems*, 1992. San Diego.

[66] Trebi-Ollennu A., Stacey B.A., and White B.A. A multivariable decoupling design of an ROV depth control system. In *IEEE Proceedings on American Control Conference*, pages 3244–3248, 1995.

[67] Sun F.C., Sun Z.Q., and Feng G. Design of adaptive fuzzy sliding mode controller for robot manipulators. In 5^{th} *IEEE Conference on Fuzzy systems*, pages 62–67, 1996.

[68] Lu Y-S. and Chen J-S. A self organizing fuzzy sliding-mode controller design for a class of nonlinear servo system. *IEEE Transactions on Industrial Electronics*, 41:492–502, 1994.

[69] Horton M.P. A study of autopilots for the adaptive control of tactical guided missiles. Master's thesis, University of Bath, 1992.

[70] Blakelock J. *Automatic Control of Aircraft and Missiles.* John Wiley, 1991.

[71] White B.A. Non-linear control of a missile. Technical report, The Royal Military College of Science, Cranfield University, DAPS, RMCS, Shrivenham, Wilts, SN6 8LA, September 1997.

[72] Suykens J. and Vandewalle J. Feedback Linearization of nonlinear distortion in electrodynamic loudspeakers. *Journal of the Audio Engineering Society*, 43(9):690–694, 1995.

[73] Henson M.A. and Seborg D.E. Input-output linearization of general nonlinear processes. *AIChE Journal*, 36(11):1754–1757, November 1990.

[74] Hauser J., Satry S., and Kokotovic P. Nonlinear control via appriximate input-output linearization: The ball and beam example. *IEEE Transactions on Automatic Control*, 37(3):392–398, March 1992.

[75] Lu X.Y., Slotine E., Spurgeon S.K., and Postlethwaite I. Robust variable structure control of a PVTOL aircraft. *International Journal of Systems Science*, 28(6):547–558, June 1997.

[76] Wang L. Design of adaptive fuzzy controllers for nonlinear systems by Input-Output Linearization. In *Proceedings of the* 1^{st} *International Joint Conference of NAFIPS IFIS NASA*, pages 89–93, 1994.

[77] Kravaris C. and Soroush M. Syntesis of multivariable nonlinear controllers by input-output linearization. *AIChE Journal*, 36(2):249–264, February 1990.

[78] Tsourdos A., Blumel A.L., and White B.A. Trajectory control of a nonlinear homing missile. In *Proceedings of the* 14^{th} *IFAC Symposium on Automatic Control in Aerospace*, pages 118–124, 1998. Seoul, Korea.

[79] Isidori A. *Nonlinear Control Systems*. Springer-Verlag, 1985.

[80] White B.A., Tsourdos A., and Blumel A.L. Lateral acceleration control design of a nonlinear homing missile. In *Proceedings of the* 4^{th} *IFAC Nonlinear Control Systems Design Symposium*, pages 708–713, 1998. Twente, Niderland.

[81] Tsourdos A. Blumel A.L. and White B.A. Autopilot design of a nonlinear missile. In *IEE Proceedings on Control'98*, pages 889–894, 1998. Swansea, UK.

[82] Tsourdos A., Blumel A.L., and White B.A. Flight control design for a missile. An approximate feedback linearization approach. In *IEEE Medeterenian Control Conference*, pages 56–64, 1999. Haifa, Israel.

[83] Kosko B. *Fuzzy Thinking*. Flamingo, 1994.

[84] Bonivento C., Fantuzzi C., and Rovatti R., editors. *Fuzzy Logic Control Advances in Methodology*. World Scientific Publishing Co, 1998.

[85] Palm R. and Driankov D. *Model Based Fuzzy Control*. Springer Verlag, 1997.

[86] Mcneill D. and Freiberger P. *Fuzzy Logic*. Touchstone, 1993.

[87] Jang J.R. and Gulley N. *Fuzzy Logic Toolbox*. The Math Works Inc., 1995.

[88] Hamm C. and Splettstoser W. Tuning a fuzzy controller by systematic selection of parameters. In *Proceedings of EUFIT*, pages 1714–1721, 1994. Aachen,Germany.

[89] Lotfi A. and Tsoi A. Importance of membership functions: A comparative study on different learning methods for fuzzy inference systems. In *IEEE Proceedings on Fuzzy Systems*, pages 1791–1796, 1994.

[90] Trebi-Ollennu A. and White B.A. A robust nonlinear control design for remotely operated vehicle depth control systems. *Transactions of Institute MC*, 19(3):119–125, March 1997.

[91] Lih-Chang Lin and Gau T. Feedback linearization and fuzzy control for conical magnetic bearings. *IEEE on Control Systems Technology*, 5(4):417–426, April 1997.

[92] Kwan C., Xu R., and Haynes L. Nonlinear control of missile dynamics. *IEEE Transactions on Decision and Control*, 4(4):4685–4690, April 1998.

[93] Leland R.P. Feedback linearization control design for systems with fuzzy uncertainty. *IEEE Transactions on Fuzzy Systems*, 6(4):492–503, April 1998.

[94] Goldberg D.E., editor. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Publishing Company, 1989.

[95] Rao S.S., editor. *Optimization: Theory and Applications.* The Institution of Electrical Engineers, 1990.

[96] Bica B., Akat G., Chipperfield A.J., and Fleming P.J. Multiobjective design of a fuzzy controller for a gas turbine aero-engine. In *IEE Proceedings of Control'98*, pages 901–906, 1998.

[97] Hong T.P., Wang C.H., and Tseng S.S. Integrating fuzzy knowledge by genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 2(4):138–149, April 1998.

[98] Liska J. and Melsheimer S. Complete design of fuzzy logic systems using GA. In *IEEE Proceedings in Intelligent Control*, pages 1377–1382, 1994.

[99] Kim Ng and Yun Li. Design of sophisticated fuzzy logic controllers using GA. Technical report, Control 931228, University of Glasgow, Glasgow, September 1994.

[100] Hughes E.J. Multi-objective, probabalistic selection evolutionary algorithms. Technical report, DAPS/EJH/56/2000, Department of Aerospace, Power and Sensors, RMCS, Cranfield University, UK, 2000.

[101] Pena-Reyes C. and Sipper M. Fuzzy CoCo: A cooperative coevolutionary approach to fuzzy modelling. *IEEE Transactions on Fuzzy Systems*, XX(Y):100–110, April 2000.

[102] Blumel A.L., Hughes E.J., and White B.A. Design of robust fuzzy controllers for aerospace applications. In *Proceedings of the* 18^{th} *NAFIPS Conference*, volume 1, pages 438–442, 1998. New York, USA.

[103] Chen Y.Y. and Perng C. Input scaling factors in fuzzy control systems. In *IEEE Proceedings on Fuzzy Systems*, pages 1666–1670, 1994.

[104] Lin C.F., editor. *Advanced Control Systems Design.* Prentice Hall Series in Advanced Navigation Guidance and Control, 1994.

[105] Ocyczka A. An approach to multicriterion optimization problems for engineering design. *Computer Methods in Applied Mechanics and Engineering*, 15:309–333, 1978.

[106] Hwang C.L., Paidy S.R., Yoon K., and Masud A.S.M. Mathematical programming with multiple objectives: A tutorial. *Journal of Computing and Operasions Research*, 7:5–31, May 1980.

[107] Fonseca C.M. and Fleming P.J. An overview of evolutionary algorithms in multiobjective optimisation. *Evolutionary Computation*, 3(1):1–16, 1995.

[108] Carlos A. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimisation techniques. *International Journal on Knowledge and Information Systems*, 1(3):269–308, august 1999.

[109] Begg D. Lin X. and Fishwick R. Genetic approach to optimal topology, controller design of adaptive structures. *International Journal for Numerical Methods in Engineering*, 41(2):815–830, 1998.

[110] Gen M., Ida K., and Li Y. Solving bicriteria solid transportation problem with fuzzy numbers by genetic algorithms. *International Journal of Computers and Industrial Engineering*, 29:537–543, 1995.

[111] Ritzel B.J., Eheart J.W., and Ranjithan S. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. *Water Resources Research*, 30(5):1589–1603, May 1994.

[112] Zeleny M. *Multiple Criteria Decision Making.* McGraw-Hill, 1982.

[113] Weistroffer H.R and Narula S.C. The current state of nonlinear multiple criteria decision making. Technical report, School of Business, Virginia Commonwealth University, Richmond, Virginia 23284-4000, USA, 1989.

[114] Weistroffer H.R. A combined over and under-achievement programming approach to multiple objectives decision making. *Journal of Large Scale Systems*, 7:47–58, May 1984.

[115] Sandgren E., editor. *Multicriteria Design Optimization by Goal Programming.* Chapman and Hall London, 1994.

[116] Wienke P.B., Lucasius C., and Kateman G. Multicriteria target optimization of analytical procedures using genetic algorithms. *Analytical Chimica Acta*, 265(2):211–225, February 1992.

[117] Wilson P. and Macleod M. Low implementation IIR digital filter design using GA. In *IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, pages 1–8, 1993.

[118] Schaffer J.D. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of* 1^{st} *International Conference on GA*, pages 93–100, 1985.

[119] Richardson J.T., Palmer M.R, Liepins G., and Hilliard M. Some guidance for genetic algorithms with penalty functions. In 3^{rd} *International Conference on GA*, pages 191–197, 1989.

[120] Hajela P. and Lin C.Y. Genetic approach strategies in multicriterion optimal design. *Structural Optimization*, 4:99–107, 1992.

[121] Goldberg D. and Richardson J. Genetic algorithm with sharing for multimodal function optimization. In *Proceedings of the* 2^{nd} *International Conference on genetic algorithms*, pages 41–49, 1987.

[122] Zalzala A.M.S. and Fleming P.J., editors. *Genetic Algorithms in Engineering Systems*. The Institution of Electrical Engineers, 1997.

[123] Deb K. Non-linear goal programming using multi-objective genetic algorithms. Technical report, CI-60/98, Department of Computer Science/LS11,University of Dortmund,Germany 1999, 1999.

[124] Srinivas N. and Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, March 1995.

[125] Fonseca C.M. and Fleming P.J. Multiobjective optimisation and multiple constraint handling with evolutionary algorithms - Part 1: A unified formulation. *IEEE Transactions on Systems, Man and Cybernetics*, 28(1):26–37, January 1998.

[126] Zitzler E. and Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3:257–271, 1999.

[127] Fonseca C.M. and Fleming P.J. Multiobjective optimisation and multiple constraint handling with evolutionary algorithms - Part 2: Application example. *IEEE Transactions on Systems, Man and Cybernetics*, 28(1):38–47, January 1998.

[128] Stoyanov S., Tellalyan J.K., Panchev E., and Preece P.E. Analysis of different types of scalarising functions in multicriteria optimal control. *SAMS*, 24:221–231, 1996.

[129] Blumel A.L., Hughes E.J., and White B.A. Fuzzy autopilot design using a multiobjective evolutionary algorithm. In *Congress on Evolutionary Computations*, volume 1, pages 54–62, 2000. San Diego, USA.

[130] White B.A., Blumel A.L., and Hughes E.J. A robust fuzzy autopilot design using multi-criteria optimisation. *International Journal of Fuzzy Systems*, 2(2):129–138, 2000.