Multi-objective Fuzzy Design of a Lateral Autopilot for a Quasi-Linear Parameter Varying Missile

Evan J. HUGHES
ejhughes@iee.org
Antonios TSOURDOS
A.Tsourdos@rmcs.cranfield.ac.uk
Brian A. WHITE
B.A.White@rmcs.cranfield.ac.uk

Department of Aerospace, Power & Sensors
Cranfield University
RMCS Shrivenham
Swindon SN6 8LA
England, UK

Tel: +44 (0)1793 785260
Fax: +44 (0)1793 785206

January, 2003

Abstract

Gain scheduled control is one very useful control technique for linear parameter-varying (LPV) and nonlinear systems. A disadvantage of gain-scheduled control is that it is not easy to design a controller that guarantees the global stability of the closed-loop system over the entire operating range from the theoretical point of view. Another disadvantage is that the interpolation increases in complexity as number of scheduling parameters increases. As an improvement, this paper presents a gain-scheduling control technique, in which fuzzy logic is used to construct a model representing a quasi-LPV or a nonlinear missile and to perform a control law. The fuzzy inference system is generated using a multi-objective evolutionary algorithm to optimise the performance characteristics of the plant.

Contents

1 Introduction **1**

2 Quasi-linear parameter varying missile model **2**

3 Quasi-linear parameter-varying representation **4**

4 Performance with pole placement via LMI **5**

5 Gain scheduling **7**
5.1 An overview 7
5.2 Design of the sideslip velocity autopilot 7
5.3 Tracking control design 8
5.4 Stability and performance design 9

6 Fuzzy Inference System **11**

7 Multiobjective Evolutionary Algorithm **11**
7.1 Introduction 11
7.2 Algorithm structure 12
7.3 Chromosome structure and objectives 13
7.3.1 Chromosome 13
7.3.2 Objectives 15
7.4 Non-dominated Ranking 15

8 Experimental Results **16**

9 Conclusions **26**

List of Figures

1 Airframe axes. 2
2 The performance of the pole placement controller for the Horton missile is presented as a D-stability region. All the poles of the system belong to the trapezoid area. 6
3 Block diagram description of the gain scheduling controller. . . . 7
4 A simple case of polytope P for a plant of dimension one and a parameter p of dimension two. 10
5 The gain scheduling controller gains for the flight envelope at 0° roll angle. 11
6 Membership function structure 14
7 Pareto set for 5 Membership Functions (MF) each, 4 MF each and 3 MF each 16
8 Membership function locations for Mach and 3 member functions 17
9 Membership function locations for Incidence and 3 member functions 18
10 Membership function locations for Mach and 4 member functions 19
11 Membership function locations for Incidence and 4 member functions 20
12 Membership function locations for Mach and 5 member functions 21
13 Membership function locations for Incidence and 5 member functions 22
14 Example gain surface for K_1 and 5 MF per input 23
15 Example gain surface for K_2 and 5 MF per input 24
16 Example gain surface for K_3 and 5 MF per input 25

List of Tables

1 Coefficients in nonlinear model (1). 3

1 Introduction

Missiles are required to operate over an expanded flight envelope to meet the challenge of highly manoeuvrable targets. In such a scenario, an autopilot derived from linearisation about a single flight condition will be unable to achieve suitable performance for the whole envelope. Thus, there is an inherent tension in autopilot design between the nonlinearity of the missile and the linearity of the controller [1].

Mathematical representation of nonlinear missile dynamics lends itself to several interpretations. The interpretations aim at deriving models which adequately capture missile behaviour *and* are practical for systematic control design.

One of the most popular methods for applying linear time-invariant (LTI) control theory to time-varying and/or quasi-linear systems is gain scheduling [2]. This strategy involves obtaining Taylor linearised models for the plant at finitely many equilibria ("set points"), designing an LTI control law ("point design") to satisfy local performance objectives for each point, and then adjusting ("scheduling") the controller gains in real time as the operating conditions vary. This approach has been applied successfully for many years, particularly for aircraft and process control problems.

Despite past success of gain scheduling in practice, until recently little has been known about it theoretically as a time-varying and/or quasi-linear control technique. Also, determining the actual scheduling routine is more of an art than a science. While *ad hoc* approaches such as linear interpolation and curve fitting may be sufficient for simple static-gain controllers, doing the same for dynamic multi-variable controllers can be a rather tedious process.

An early theoretical investigation into the performance of parameter-varying systems can be found in [3]. During the 1980's, Rugh and his colleagues developed an analytical framework for gain scheduling using extended linearisation [2]. Also, Shamma and Athans [4] introduced linear parameter-varying (LPV) systems as a tool for quantifying such heuristic design rules as "the resulting parameter must vary slowly" and "the scheduling parameter must capture the nonlinearities of the plant". Shahruz and Behtash [5] suggested using LPV systems for synthesising gain-scheduled controllers.

In this paper a fuzzy pole-placement control design technique is applied to the autopilot design for the missile. The missile motion is modelled to be quasi-linear with unknown parameters. Based on the quasi-linear model, we adopt for design procedure the fuzzy pole-placement method. The performance objectives related with the transient, i.e. settling time, rising time, peak overshoot are achieved with

the fuzzy pole-placement. However, since our problem is one of tracking, an additional performance objective, that of zero steady-state error should be taken into account. This can be achieved with an integral term in forward loop. In this scheme, unknown parameters are estimated and based on these estimates, control parameters are updated. Computer simulations show that this approach is very promising for the motion control design for missiles, which are highly quasi-linear in dynamics. The optimisation of the fuzzy system is performed using a multiobjective evolutionary algorithm [6].

Section 2 details the missile model and coefficients, section **??** describes the design of the controller and the structure of the fuzzy inference system. The multiobjective evolutionary algorithm is detailed in section 7. Section 8 shows typical results from the optimisation process and section 9 concludes.

2 Quasi-linear parameter varying missile model

Missile autopilots are usually designed using linear models of nonlinear equations of motion and aerodynamic forces and moments [7], [8]. The objective of this paper is the design of a lateral acceleration autopilot for a quasi-linear parameter varying missile model. This model describes a reasonably realistic airframe of a tail-controlled tactical missile in the cruciform fin configuration (Figure 1). The aerodynamic parameters in this model are derived from wind-tunnel measurements [9].

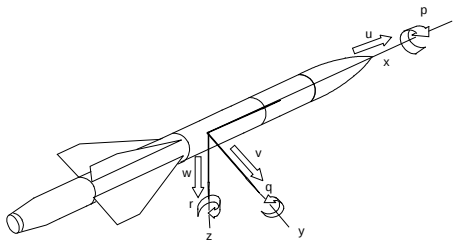

Figure 1: Airframe axes.

The starting point for mathematical description of the missile is the following nonlinear model [10], [9] of the horizontal motion (on the xy plane in Figure 1):

$$\dot{v} \quad = \quad y_v(M, \lambda, \sigma)v - Ur + y_\zeta(M, \lambda, \sigma)\zeta$$

Table 1: Coefficients in nonlinear model (1).

	Interpolated formula
C_{y_v}	$0.5[(-25+M-60\|\sigma\|)(1+\cos 4\lambda)+$ $(-26+1.5M-30\|\sigma\|)(1-\cos 4\lambda)]$
C_{y_ζ}	$10+0.5[(-1.6M+2\|\sigma\|)(1+\cos 4\lambda)+$ $(-1.4M+1.5\|\sigma\|)(1-\cos 4\lambda)]$
C_{n_r}	$-500-30M+200\|\sigma\|$
C_{n_v}	$s_m C_{y_v}$, where: $s_m = d^{-1}[1.3+0.1M+0.2(1+\cos 4\lambda)\|\sigma\|+$ $0.3(1-\cos 4\lambda)\|\sigma\|-(1.3+m/500)]$
C_{n_ζ}	$s_f C_{y_\zeta}$, where: $s_f = d^{-1}[2.6-(1.3+m/500)]$

$$\begin{aligned} & = \frac{1}{2}m^{-1}\rho V_o S(C_{y_v}v + V_o C_{y_\zeta}\zeta) - Ur \\ \dot{r} &= n_v(M,\lambda,\sigma)v + n_r(M,\lambda,\sigma)r + n_\zeta(M,\lambda,\sigma)\zeta \\ & = \frac{1}{2}I_z^{-1}\rho V_o S d\Big(\frac{1}{2}dC_{n_r}r + C_{n_v}v + V_o C_{n_\zeta}\zeta\Big). \end{aligned} \tag{1}$$

where the variables are defined in Figure 1.

Here v is the sideslip velocity, r is the body rate, ζ the rudder fin deflections, y_v, y_ζ semi-non-dimensional force derivatives due to lateral and fin angle, n_v, n_ζ, n_r semi-non-dimensional moment derivatives due to sideslip velocity, fin angle and body rate. Finally, U is the longitudinal velocity. Furthermore, $m = 125$ kg is the missile mass, $\rho = \rho_0 - 0.094h$ air density ($\rho_0 = 1.23$ kgm^{-3} is the sea level air density and h the missile altitude in km), V_o the total velocity in ms^{-1}, $S = \pi d^2/4 = 0.0314$ m^2 the reference area ($d = 0.2$ m is the reference diameter) and $I_z = 67.5$ kgm^2 is the lateral inertia. For the coefficients $C_{y_v}, C_{y_\zeta}, C_{n_r}, C_{n_v}, C_{n_\zeta}$ only discrete data points are available, obtained from wind tunnel experiments. Hence, an interpolation formula, involving the Mach number $M \in [0.6, 6.0]$, roll angle $\lambda \in [4.5°, 45°]$ and total incidence $\sigma \in [3°, 30°]$, has been calculated with the results summarised in Table 1.

The total velocity vector $\vec{V_o}$ is the sum of the longitudinal velocity vector $\vec{U}$ and the sideslip velocity vector $\vec{v}$, i.e. $\vec{V_o} = \vec{U} + \vec{v}$, with all three vectors lying on the xy plane (see Figure 1). We assume that $U \gg v$, so that the total incidence σ, or the angle between $\vec{U}$ and $\vec{V_o}$, can be taken as $\sigma = v/V_o$, as $\sin\sigma \approx \sigma$ for small σ. Thus, we have $\sigma = v/V_o = v/\sqrt{v^2+U^2}$, so that the total incidence is a nonlinear function of the sideslip velocity and longitudinal velocity, $\sigma = \sigma(v, U)$.

The Mach number is obviously defined as $M = V_o/a$, where a is the speed of sound. Since $V_o = \sqrt{v^2 + U^2}$, the Mach number is also a nonlinear function of the sideslip velocity and longitudinal velocity, $M = M(v, U)$.

It follows from the above discussion that all coefficients in Table 1 can be interpreted as nonlinear functions of three variables: sideslip velocity v, longitudinal velocity U and roll angle λ.

For an equilibrium (v_0, r_0, ζ_0) it is possible to derive from (1) a linear model in incremental variables, $\bar{v} \doteq v - v_0$, $\bar{r} \doteq r - r_0$ and $\bar{\zeta} \doteq \zeta - \zeta_0$. In particular, for the straight level flight (with gravity influence neglected), we have $(v_0, r_0, \zeta_0) = (0, 0, 0)$, so that the incremental and absolute variables are numerically identical, although conceptually different.

3 Quasi-linear parameter-varying representation

The missile model introduced in Equation (1) is nonlinear with explicit state dependence, v, directly proportional to σ. The technique presented in this paper, gain scheduling, Rugh et al [**?**], start from the quasi-linear parameter-varying form of this model.

Assume the nonlinear model of the form,

$$\begin{aligned} \dot{x} &= f(x, u, q) \\ y &= h(x, u, q), \end{aligned} \tag{2}$$

where the x is the state vector, u is the control input of the system, y is the output and q an exogenous parameter.

The set, $\mathcal{E}$, of operating points (equilibrium points) of Equation (2) depends on parameter $q \in \mathcal{R}$ and is denoted as

$$\mathcal{E} = \{(x_0, u_0, q) \in \mathcal{X} \times \mathcal{U} \times \mathcal{R} | \quad f(x_0, u_0, q) = 0\}. \tag{3}$$

The set of equilibrium for the missile is defined as $\mathcal{E} = \{(v_0, r_0, \zeta_0, U_0, \lambda_0) \in \mathcal{X} \times \mathcal{U} \times \mathcal{R} | \quad f(v_0, r_0, \zeta_0, U_0, \lambda_0) = 0\}$ where $f(v, r, \zeta, U, \lambda)$ represents the nonlinear differential Equation (1). The parameter $p = (v, U, \lambda)$ is introduced and it uniquely determines an operating point (equilibrium point) of the system (a point of $\mathcal{E}$). This parameter p depends on the state variable v and on the external parameters U and λ.

Since the term $y_\zeta(p)\delta\zeta$ will be small, due to the small lateral force generated Note that the control of the lateral acceleration (latax), a_v, can be approximately

performed via the control of the lateral velocity, $a_v = \dot{v} + Ur \approx y_v v$, since the term $y_\zeta \zeta$ will be small, due to the small lateral force generated by the tail fins compared to the tail fin moment.

Assuming incremental state variable, $\delta x = x - x_0(p)$, control input, $\delta u = u - u_0(p)$, and output, $\delta y = y - y_0(p)$, Taylor linearisation of the nonlinear system (2) at an operating point of $\mathcal{E}$, uniquely defined by parameter p, leads to

$$\begin{aligned} \delta\dot{x} &= \frac{\partial f}{\partial x}_{|p} \delta x + \frac{\partial f}{\partial u}_{|p} \delta u \\ \delta y &= \frac{\partial h}{\partial x}_{|p} \delta x. \end{aligned} \tag{4}$$

For an equilibrium (v_0, r_0, ζ_0) uniquely defined by p, the Taylor linearisation of (1) gives a linear model in incremental variables, $\delta v \doteq v - v_0$, $\delta r \doteq r - r_0$ and $\delta\zeta \doteq \zeta - \zeta_0$. In fact, dependence on p states a quasi-linear parameter-varying (QLPV) form (5), with p comprises *both* a state variable, v and external parameters, U and λ.

$$\begin{aligned} \dot{\delta v} &= \left(\frac{\partial y_v}{\partial v} v + y_v + \frac{\partial y_\zeta}{\partial v}\zeta\right)_{|p} \delta v - U_{|p}\delta r + y_{\zeta|p}\delta\zeta \\ \dot{\delta r} &= \left(\frac{\partial n_v}{\partial v} v + n_v + \frac{\partial n_r}{\partial v} r + \frac{\partial n_\zeta}{\partial v}\zeta\right)_{|p} \delta v + n_{r|p}\delta r + n_{\zeta|p}\delta\zeta \\ \delta y &= \delta v. \end{aligned} \tag{5}$$

In particular, for the straight level flight, we have $(v_0, r_0, \zeta_0) = (0, 0, 0)$, so that the incremental and absolute system forms are identical even if conceptually different, see Equation (6).

$$\begin{aligned} \begin{bmatrix} \dot{\delta v} \\ \dot{\delta r} \end{bmatrix} &= \begin{bmatrix} y_v(p) & -p_2 \\ n_v(p) & n_r(p) \end{bmatrix} \begin{bmatrix} \delta v \\ \delta r \end{bmatrix} + \begin{bmatrix} y_\zeta(p) \\ n_\zeta(p) \end{bmatrix} \delta\zeta \\ \delta y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \delta v \\ \delta r \end{bmatrix}. \end{aligned} \tag{6}$$

4 Performance with pole placement via LMI

Before the desired level of performance is described, it has to be noticed that the missile is limited by both its natural behaviour and its actuators performance. The actuators limit the steering of the missile and consequently its performance. These actuators are modelled as second order systems, their characteristic is given by

their angle range, ± 0.3 rad, and their frequency response, 250 rad/s (frequency about 40 Hz). It has been chosen to keep the response of the pole placement controller within these constraints so that the actuators do not operate above their cut off frequency. This is achieved by limiting the frequency response of the pole placement controller less than 125 rad/s (about 20 Hz) and keeping the damping ratio above the critical value, for actuator fatigue and power consumption reasons. Additionally, the desired performance requires that the system performs within 0.1 s of maximum rising time.

The resulting D-stability region, where the poles have to stay in, is the intersection of three primitives: a cone in the left half-plane with half-angle $\pi/4$ and tip at $(0, 0)$ (for the damping), the half-plane left to -40 (for the time rising) and the half-plane right to -87.5 (for the cuff off frequency). The physical angle range of the fins is not taken into account in the design process.

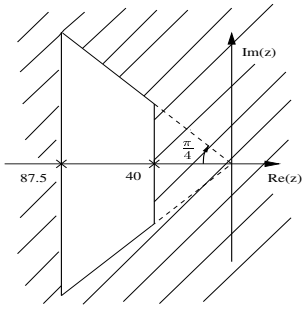

Figure 2: The performance of the pole placement controller for the Horton missile is presented as a D-stability region. All the poles of the system belong to the trapezoid area.

Modified Lyapunov equations characterise this D-stability region where they turn the classic Lyapunov equation into a family of Linear Matrix Inequalities. In the following, the feasibility of these is solved with the LMI Toolbox, Gahinet et al [**?**] for MATLAB. Assuming a region $\mathcal{D} = \{z \in C| \quad L + Mz + M^T\bar{z} < 0\}$, the matrix A' would have all its eigenvalues in this region $\mathcal{D}$, if there exists a positive-definite symmetric matrix X satisfying the following LMIs (7).

$$\forall i, j \quad \lambda_{i,j} X + \mu_{i,j} A' X + \mu_{j,i} X A'^T < 0, \tag{7}$$

where $\lambda_{i,j}$ and $\mu_{i,j}$ represent respectively the matrix coefficients of L and M. For system (5), it has been applied in the case of state feedback control design where

the matrix A' is represented by the closed-loop matrix $A + BK$. It brings (8) with both $Y = KX$ and X positive-definite symmetric.

$$\forall(i,j) \quad \lambda_{i,j}X + \mu_{i,j}AX + \mu_{j,i}XA^T + \mu_{i,j}BY + \mu_{j,i}Y^TB^T < 0, \tag{8}$$

5 Gain scheduling

5.1 An overview

Figure 3 shows a block diagram description of gain scheduling. The linear parameter-varying system constitutes a family of systems and for each of them a LTI design is carried on. The gain feedback controller is scheduled by the parameter p and becomes a gain scheduling controller. From a discrete family of gain feedback controllers first computed to guarantee performance at some operating points (related to p) and valid in its neighbourhood; the linear interpolation of these controllers carries over the performance of the closed-loop system on the whole flight envelope. The controller provides an incremental control input $\delta\zeta$ and its total control to the plant is recovered as $\zeta(p) = \zeta_0(p) + \delta\zeta$.

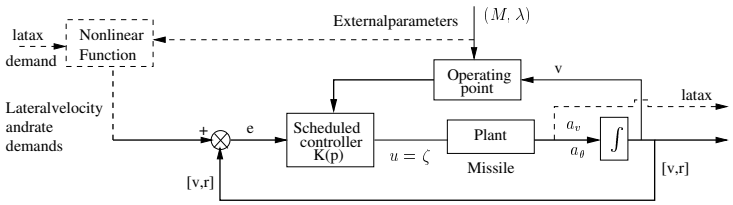

Figure 3: Block diagram description of the gain scheduling controller.

5.2 Design of the sideslip velocity autopilot

Section 3 described the Horton missile in a linearised form and from Equation (5) the incremental closed-loop system is derived for state feedback as follows,

$$\delta\dot{x} \quad = \quad A(p)\delta x + B(p)\delta u = [A(p) + B(p)K(p)]\,\delta x, \tag{9}$$

where $K(p) = [\,K_1(p) \quad K_2(p)\,]$ is the gain scheduled controller.

5.3 Tracking control design

This controller would result only to a desired transient of all local models by placing the poles of all the local systems within a specified area. However since our aim is good tracking for the missile we should include to the design specification except peak overshoot and settling time, zero steady-state error. This can be achieved with an integral term in the forward path.

The new augmented model would contain for this system one more state variable to account for this integral term. This new state variable is defined as:

$$x_i = \int_{t_0}^{t} \mathbf{e} dt = \int_{t_0}^{t} (y - r)\, dt \tag{10}$$

Therefore,

$$\dot{x}_i = [y_d - r] \tag{11}$$

The state space now is described by:

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ - \\ \dot{x}_i \end{bmatrix} &= \begin{bmatrix} A(p) & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} \\ &+ \begin{bmatrix} B \\ 0 \end{bmatrix} \zeta + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} r \end{aligned} \tag{12}$$

The compensated system therefore becomes:

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ - \\ \dot{x}_i \end{bmatrix} &= \begin{bmatrix} A(p) - BK(p) & -BK_i \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ - \\ x_i \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{0} \\ - \\ \mathbf{I} \end{bmatrix} r \end{aligned} \tag{13}$$

The characteristic polynomial of the compensated system is then equated with the desired polynomial at each step to adapt the controller gains.

The compensated system is of one order higher than the nominal one. This is because of the integral term, added for tracking purposes. The third pole has to be placed however in such location that the third order compensated system to behave similar to a second order. This add an extra requirement to the selection of gains for pole placement.

5.4 Stability and performance design

A suitable pole placement for this closed-loop system (9) will guarantee stability and performance accordingly. A polytopic approach is chosen here to capture the system nonlinearities. The polytope has to be sufficiently fine and representative of the closed-loop system on the full flight envelope. However, the convex hull on the full range of flight envelope give only very limited insight of the nonlinearities involved and a family of convex hull representative of the system on the flight envelope is preferred.

Hence, a controller is computed such that a Lyapunov function on each convex polytope of the closed-loop system can be found. This process can lead to some conservative results but there is guarantee of performance criteria introduced in Section 4 on each of these polytopes. In this work, the LMIs involved have been solved with the LMI Toolbox, Gahinet et al [**?**] for Matlab. The process is still difficult since there is no easy way to estimate the refinement needed to meet the requirements and the feasibility of the LMIs. Consequently, using genetic algrithms, the flight envelope is split as many times as necessary to obtain feasibility of the LMIs on each of sub-domain [**?**]. This approach leads to a discrete family of controllers each of them valids in its direct neighbourhood i.e. its corresponding convex polytope. The family of polytopes, noted $\mathcal{P}$ has to cover the whole flight envelope, $\mathcal{F}$,

$$\bigcup_{(i,j,k)\in N^l\times N^m\times N^n} P(i,j,k) = \mathcal{P} \subset \mathcal{F}, \tag{14}$$

where (l, m, n) is the total number of controllers on each dimension. From the design results a family of controllers given by,

$$\mathcal{K} = \{K(i,j,k)| \quad \forall (i,j,k) \in N_l \times N_m \times N_n \quad K(i,j,k) \text{ valid on polytope } P(i,j,k)\} \tag{15}$$

So far, the performance has been guaranteed only on each individual convex polytope. To extend this result to the whole flight envelope further constraints on the design have been imposed. The linear interpolation of the gains of successive controllers has been considered in the following. Each controller $K(i_0, j_0, k_0)$ has to satisfy the LMIs at $p(i_0, j_0, k_0)$, as in Section 4, and in its direct neighbourhood in a common domain with other neighbouring controllers. In this work the direct neighbour of $K(i_0, j_0, k_0)$ considered is as far as the $p(i, j, k) \quad \forall (i, j, k) \in \{i_0 - 1, i_0 + 1\} \times \{j_0 - 1, j_0 + 1\} \times \{k_0 - 1, k_0 + 1\}$. Finally, any linear combination of these controllers in a direct neighbourhood satisfies the LMIs and hence

the performance. Figure 4 is an attempt to visualise such polytopes for low dimensions.

On each polytope a Lyapunov function is found satisfying the LMIs and gives a linear controller. Thus, the linear interpolation between these controllers carries over the performance properties of the closed-loop system to the whole flight envelope.

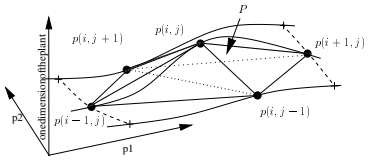

Figure 4: A simple case of polytope P for a plant of dimension one and a parameter p of dimension two.

Next it is now proved, using resuls of Section 4, that this last approach guarantees the performance on whole flight envelope. Let assume the controller $K(i_0, j_0, k_0)$, which is referred in the following of this section by a subscript 0. The common Lyapunov matrices X_0 and Y_0 in Equation (8) lead to K_0 valid on P_0. And according to the discussion before, for instance, K_0 is valid up to point p_{i_0-1,j_0-1,k_0-1} (noted p_{-3}) of sub-polytope P_{-3}, in consequence the following holds,

$$\begin{aligned} \lambda X_0 + \mu A_0 X_0 + \mu^T X_0 A_0^T + \mu B_0 Y_0 + \mu^T Y_0^T B_0^T &< 0 \\ \lambda X_0 + \mu A_{-3} X_0 + \mu^T X_0 A_0^T + \mu B_{-3} Y_0 + \mu^T Y_0^T B_{-3}^T &< 0, \end{aligned} \tag{16}$$

where to simplify $\lambda_{i,j}$ is noted λ, $\mu_{i,j}$ and $\mu_{j,i}$ are noted by μ and μ^T respectively. In a similar way, controller $K(i_0 - 1, j_0 - 1, k_0 - 1)$ is valid at p_{-3} and p_0. The combination of these results shows that the linear interpolated controller between these points still satisfies the LMIs (8), hence the performance criteria.

Figure 5 shows the controller gains on the whole range of Mach number and incidence angle for zero roll angle. There are $9 \times 18 = 162$ controllers in order for the system to satisfy the performance criteria. The time responses for the linear interpolated controller is shown on Figure **??** for latax demand, $100\ \mathrm{m/s^2}$ at Mach 2.15. As expected, the velocity satisfies the performance requirements of settling time and damping.

Figure 5: The gain scheduling controller gains for the flight envelope at 0° roll angle.

6 Fuzzy Inference System

A Takagi-Sugeno (T-S) fuzzy controller [11] is used to determine the control gains required for any given Mach and incidence angle in order to generate a system with a given performance characteristic. The system has two inputs, Mach and incidence, and generates three outputs which are the three control gains required for the PI controller.

The Takagi-Sugeno (T-S) fuzzy controller is composed of r rules that can be represented as:

Plant rule i: **If** e_i is M_j and e_i is I_k
Then $\delta K_{n_i} = K_{n_i}$,
$i = l, 2, \ldots r$, and $n = 1, \ldots 3$,

Where M_j and I_k are individual membership functions of the two inputs and K_{n_i} is the required set of gains for the rule.

The T-S fuzzy model infers the gains $K_{n_i}(t)$ as the output of the fuzzy model, given all the rules, as follows, where ν_i is the total degree of membership for rule i.

$$K_{n_i} = \frac{\sum_{i=1}^{r} \nu_i [\delta K_{n_i}]}{\sum_{i=1}^{r} \nu_i} \tag{17}$$

7 Multiobjective Evolutionary Algorithm

7.1 Introduction

Evolutionary Algorithms are optimisation procedures which operate over a number of cycles (generations) and are designed to mimic the natural selection process through evolution and survival of the fittest [6]. A *population* of M independent

individuals is maintained by the algorithm, each individual representing a potential solution to the problem. Each individual has one *chromosome*. This is the genetic description of the solution and may be broken into n sections called *genes*. Each gene represents a single parameter in the problem, therefore a problem that has eight unknowns for example, would require a chromosome with eight genes to describe it.

The three simple operations found in nature, natural selection, mating and mutation are used to generate new chromosomes and therefore new potential solutions. In this paper, an evolutionary strategy was used where new chromosomes were generated by a combination of mating (otherwise known as *crossover*) and applying Gaussian noise to each gene in each chromosome, with a standard deviation that evolved along with each gene. Each chromosome is evaluated at every generation using an *objective function* that is able to distinguish good solutions from bad ones and to score their performance. With each new generation, some of the old individuals die to make room for the new, improved offspring. Despite being very simple to code, requiring no directional or derivative information from the objective function and being capable of handling large numbers of parameters simultaneously, evolutionary algorithms can achieve excellent results.

7.2 Algorithm structure

The evolutionary strategy begins by generating an initial population of 50 chromosomes at random with the standard deviations of the mutations all set initially as one eighth of the total range of each gene. The initial population is evaluated and objective values generated (see section 7.3) and then sorted (section7.4). Crossover and mutation are then applied to the chromosomes to generate another 50 chromosomes. These new chromosomes are then evaluated and the best 50 from all 100 chromosomes are chosen for the next generation. The process is repeated for 100 generations.

The crossover operation takes each chromosome in turn (chromosome a), and for each chooses a second chromosome at random (with replacement) to cross with (chromosome b). A new chromosome (c) is generated 70% of the time using (18), and for the remaining 30% of the time, a copy of chromosome a is made. In (18), a_k, b_k & c_k are gene k of chromosomes a, b & c and U_k is a uniform random number in the range [0,1] chosen anew for each gene and each chromosome a.

$$c_k = a_k + (b_k - a_k)(1.5U - 0.25) \tag{18}$$

The evolutionary strategy updates the standard deviation of the mutation and

the value of each gene for every gene in each new chromosome, using (19). In (19), ${\sigma'}_k(x)$ is the standard deviation of gene k of chromosome x, ${\omega'}_k(x)$ is the value of gene k of chromosome x, $N(0,1)$ is a random number with zero mean and unity variance Gaussian distribution and is chosen once per chromosome, $N_k(0,1)$ is a random number with zero mean and unity variance Gaussian distribution and is chosen afresh for every gene, and n is the number of genes in each chromosome.

$$\begin{aligned} {\sigma'}_k(x) &= \sigma_k(x)\exp(\tau_0 N(0,1) + \tau_1 N_k(0,1)) \\ {\omega'}_k(x) &= \omega_k(x) + {\sigma'}_k(x) N_k(0,1) \\ \tau_0 &= \frac{1}{\sqrt{2\sqrt{n}}} \\ \tau_1 &= \frac{1}{\sqrt{2n}} \end{aligned} \tag{19}$$

7.3 Chromosome structure and objectives

7.3.1 Chromosome

The chromosome structure needs to represent both the membership functions for the two inputs, and the output values for every possible rule. Three, four and five membership functions have been used for each of the two inputs. The member functions are triangular and overlapping to always give a unity sum as shown in figure 6

For the two inputs, the input ranges are $e_0 = 0.6$ to $e_m = 6$ for the Mach number, and $e_0 = 0^\circ$ to $e_m = 30^\circ$ for the incidence. For example, for four member functions on an input, three genes are required to describe the relative positions of the peaks of the member functions as shown in figure 6. This process gives a total of 4 genes to represent the membership functions for three member functions per input, 6 genes for four member functions per input, and 8 for five member functions. Each of the genes must lie in the range (0,1].

With n member functions per input, there will be n^2 possible rules. The output value for each the rules is simply a triplet of constants, one for each of the three outputs. Therefore with say four input member functions on each input, there are 16 possible rules, at 16 possible combinations of Mach and incidence. At each Mach–incidencecombination, the three control gains are calculated by evaluating a local model of the system. The gains calculated by the local model are then associated with the corresponding rule and used to create the fuzzy control surface.

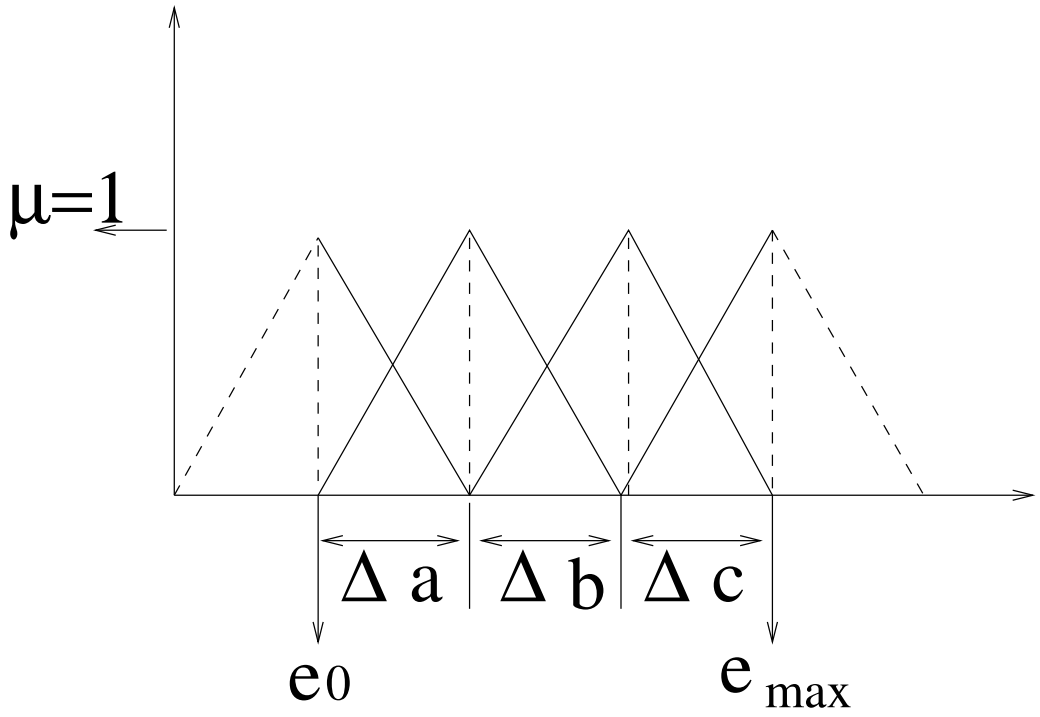

Figure 6: Membership function structure

7.3.2 Objectives

The performance is tested by generating the step response of the system for 100 uniformly spaced points in the Mach / incidence domain (10 per input). The rise time and settling time of the system are recorded at each point. Two objectives are then generated that summarise the performance of each chromosome.

The first objective is taken as the difference between the slowest and fastest rise times of the 100 trials for each chromosome. The second objective is the difference between the slowest and fastest settling times.

7.4 Non-dominated Ranking

With multiple objectives, a Pareto-optimal set of results [6] may be formed where no single solution is better than any other in all objectives. These solutions are said to be *non-dominated* as no solution can be chosen in preference to the others based on the all objectives alone. There exists a single Pareto-optimal set of solutions to the problem. At any intermediate stage of optimisation, a set of non-dominated results will have been identified. This set may or may not be the Pareto optimal set.

A non-dominated ranking method [6] is used in the evolutionary algorithm to generate and maintain a non-dominated set of results. Conventional evolutionary algorithms often use a ranking method where the calculated objective values are sorted and assigned a rank that is dependent only upon their position in the list, rather than their objective value. The ranking operation helps to prevent premature convergence of the evolutionary algorithm.

The non-dominated ranking system operates by first identifying the non-dominated solutions in the population and assigning them a rank of one. A dummy value (1 in this implementation) is assigned to these solutions and a sharing process is applied. With the sharing, the dummy values of the individuals' are reduced if they have near neighbours (in the objective space). The sharing process ensures that a spread of solutions is obtained across the non-dominated front. The minimum value assigned to the level-one solutions is identified and then reduced slightly (by 1%) and used as a dummy value for the next level of processing. The level-one individuals are removed from the population and the identification–sharing process repeated on the remaining set, using the reduced dummy value for the sharing operation. The ranking process is continued until all of the individuals have been accounted for. The resulting objectives are intended to be used with a *maximisation* strategy and have been adjusted to allow both of the objectives to be

minimised.

8 Experimental Results

Figure 7 shows a set of typical non-dominated surfaces after 100 generations for each of the membership function configurations. Both of the objectives cannot be minimised simultaneously, so the Pareto front forms curves. All of the solutions on the non-dominated front are valid solutions to the problem and it is down to the system designer to choose a single solution for use in the control system.

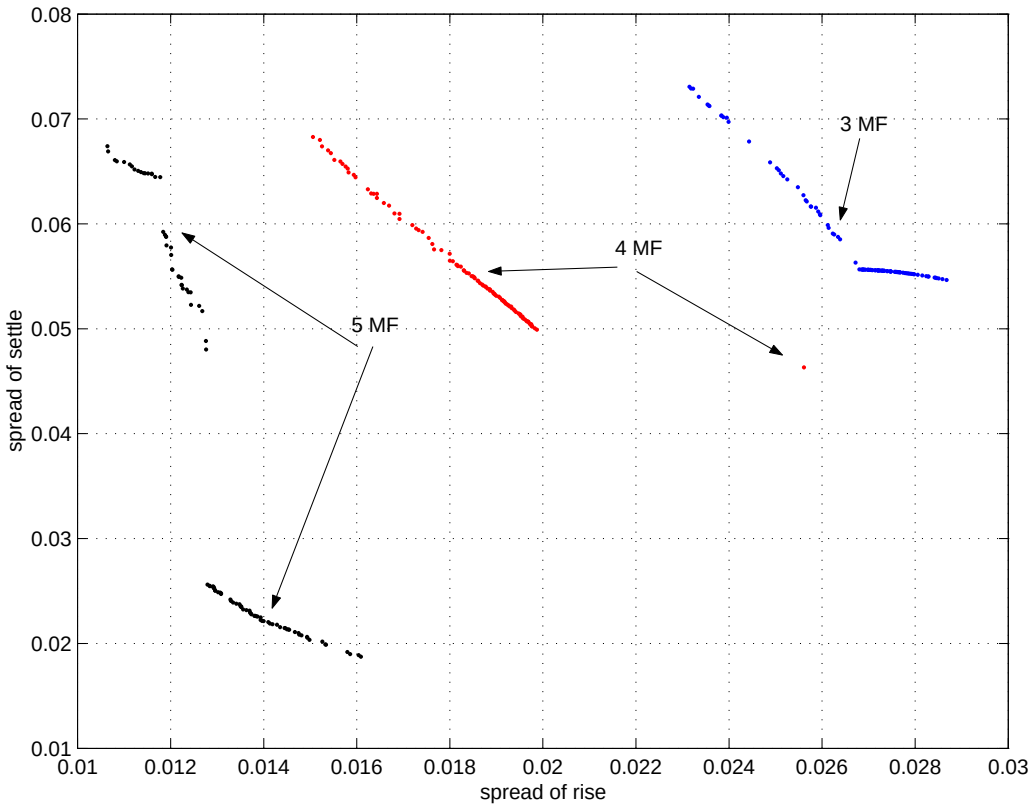

Figure 7: Pareto set for 5 Membership Functions (MF) each, 4 MF each and 3 MF each

Figures 8 and 9 show the locations of the membership functions for the 3 membership function system (MF 1 and 3 are at zero and 1 respectively). The

positions are sorted to correspond to the order of the points on the Pareto set, with point 1 corresponding to the solution in the top left hand corner of the Pareto set. Figures 10, 11, 13 & 8 show the corresponding plots for the trials with 4 and 5 membership functions per input.

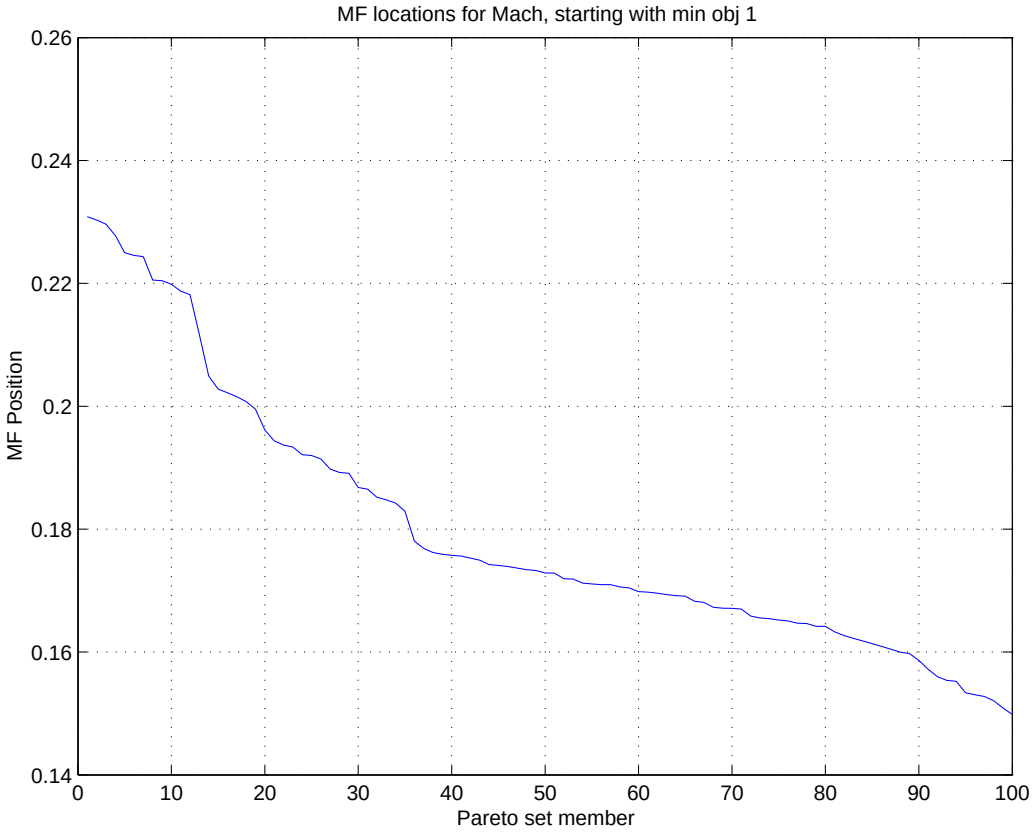

Figure 8: Membership function locations for Mach and 3 member functions

It is clear from figures 8, 10 & 12 that the Mach input is dominant when shaping the control surfaces. The lines on the plots progress smoothly with respect to the objective surface, whereas figures 9, 11 & 13 have little correlation with the progression of the objectives. This effect suggests that fewer membership functions are required for the incidence input.

Figures 14, 15 & 16 show the surfaces generated by the fuzzy inference systems for the three control gains for the solution with 5 membership functions in both inputs that minimises the error in the spread of rise times.

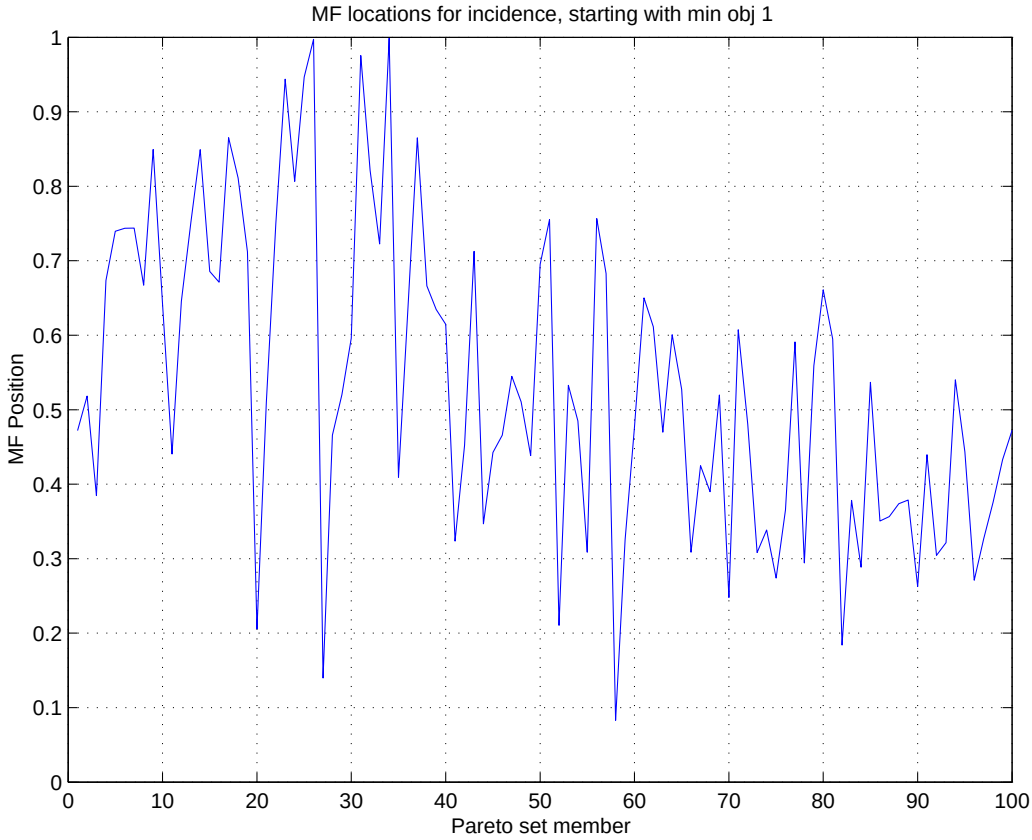

Figure 9: Membership function locations for Incidence and 3 member functions

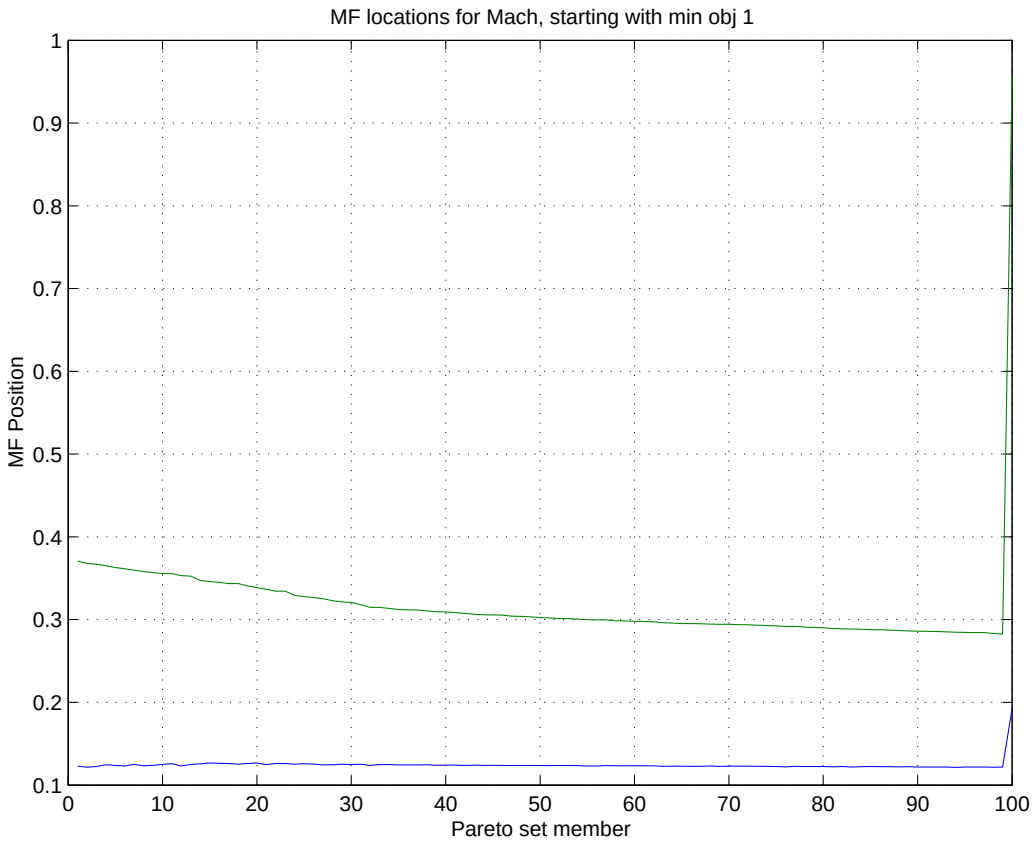

Figure 10: Membership function locations for Mach and 4 member functions

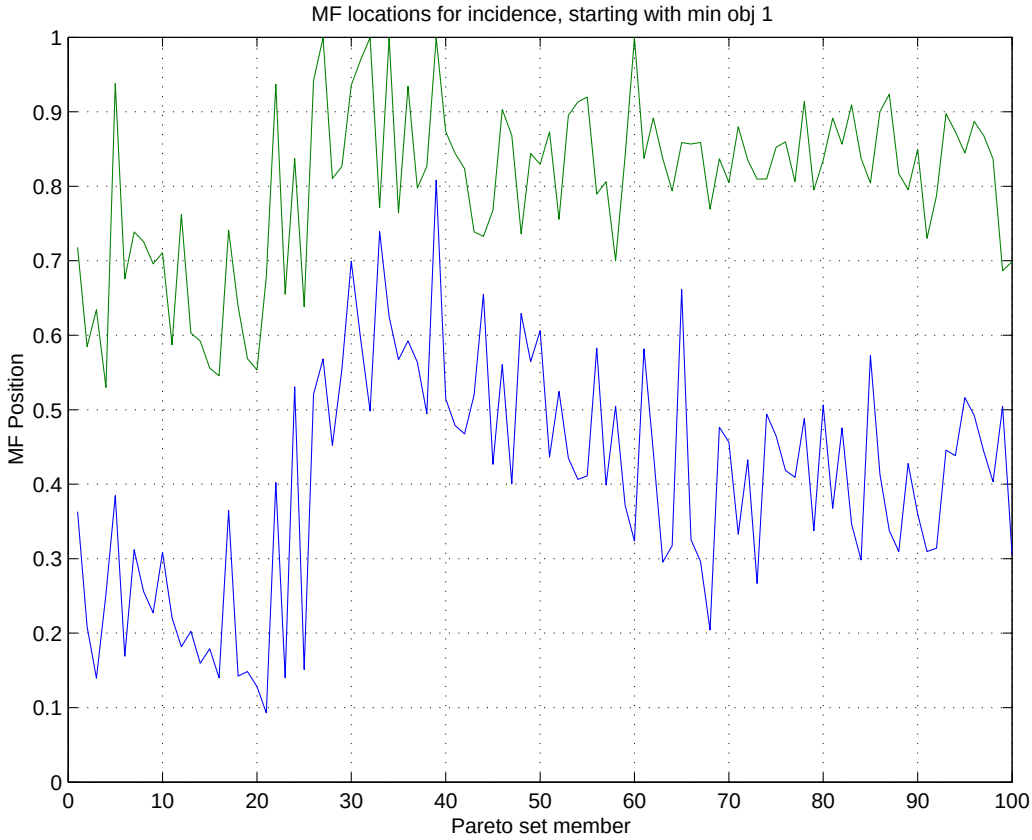

Figure 11: Membership function locations for Incidence and 4 member functions

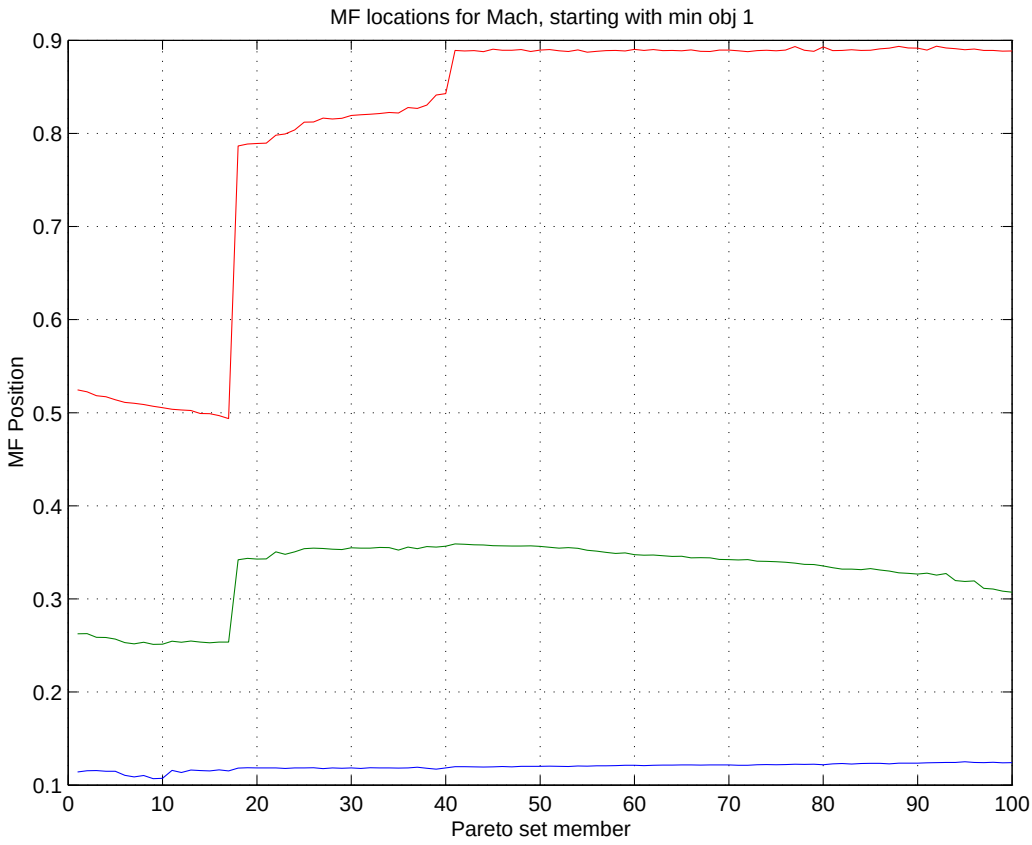

Figure 12: Membership function locations for Mach and 5 member functions

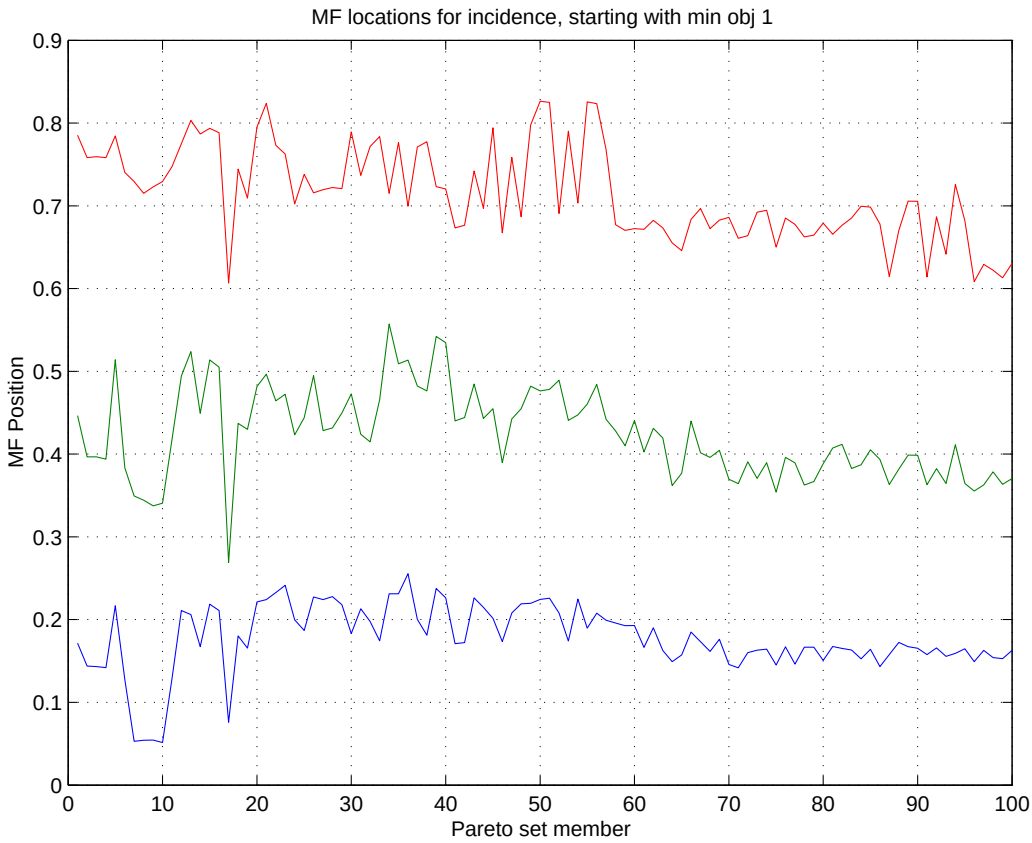

Figure 13: Membership function locations for Incidence and 5 member functions

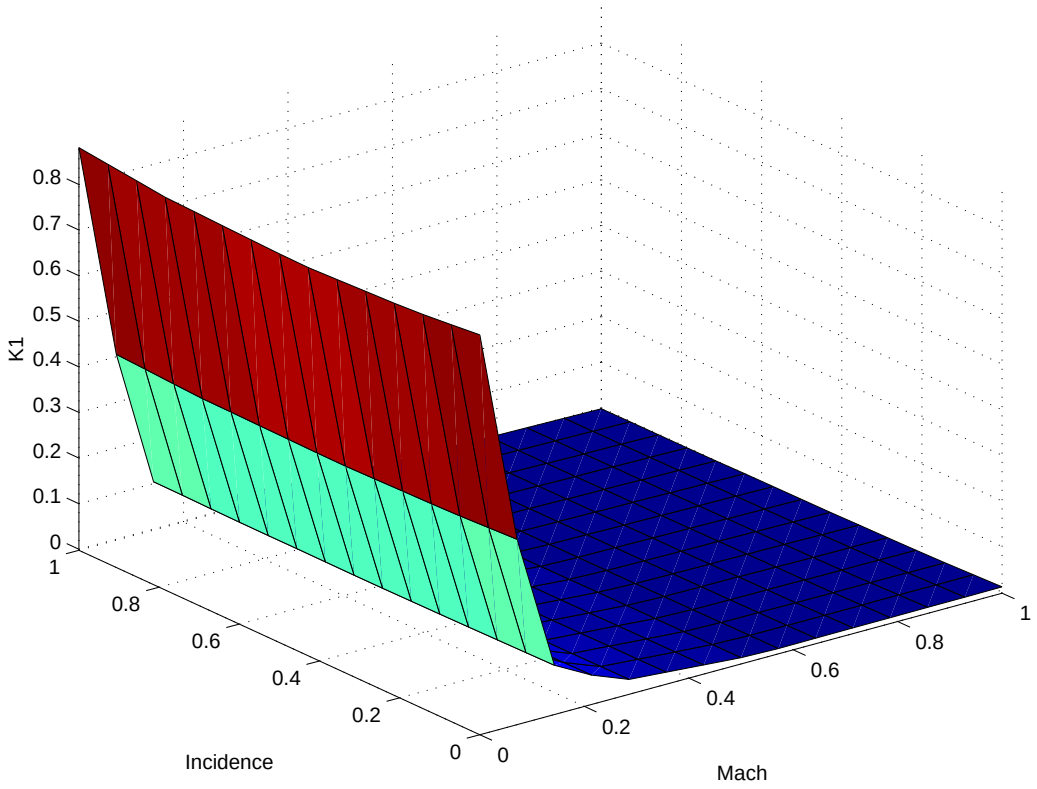

Figure 14: Example gain surface for K_1 and 5 MF per input

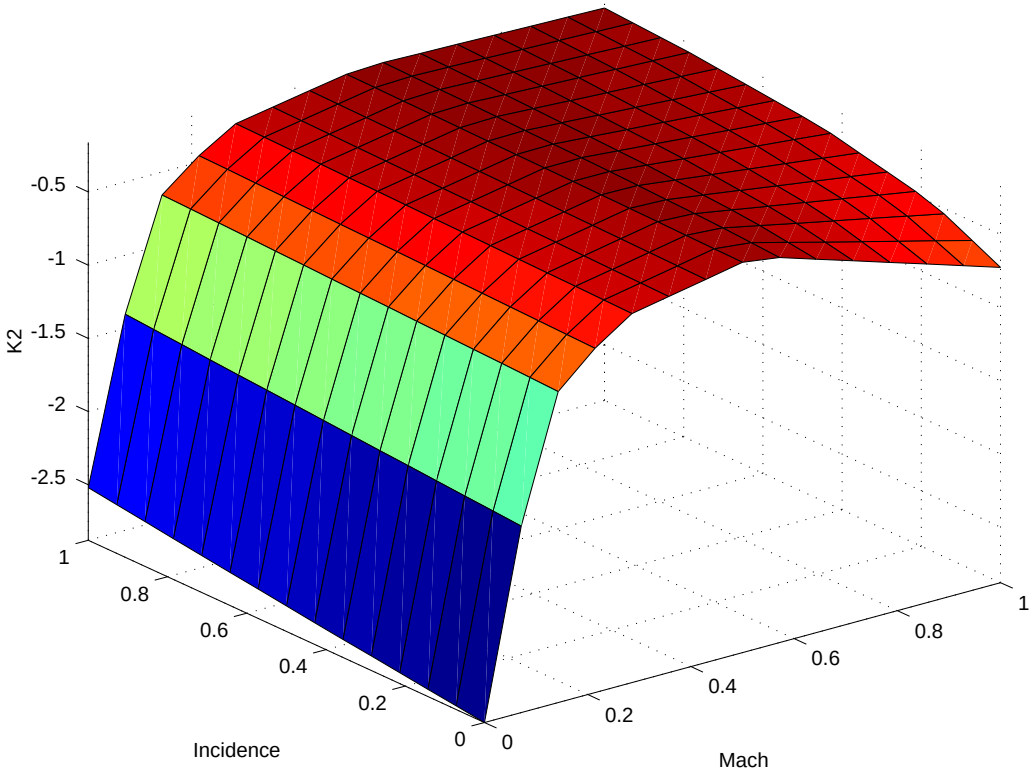

Figure 15: Example gain surface for K_2 and 5 MF per input

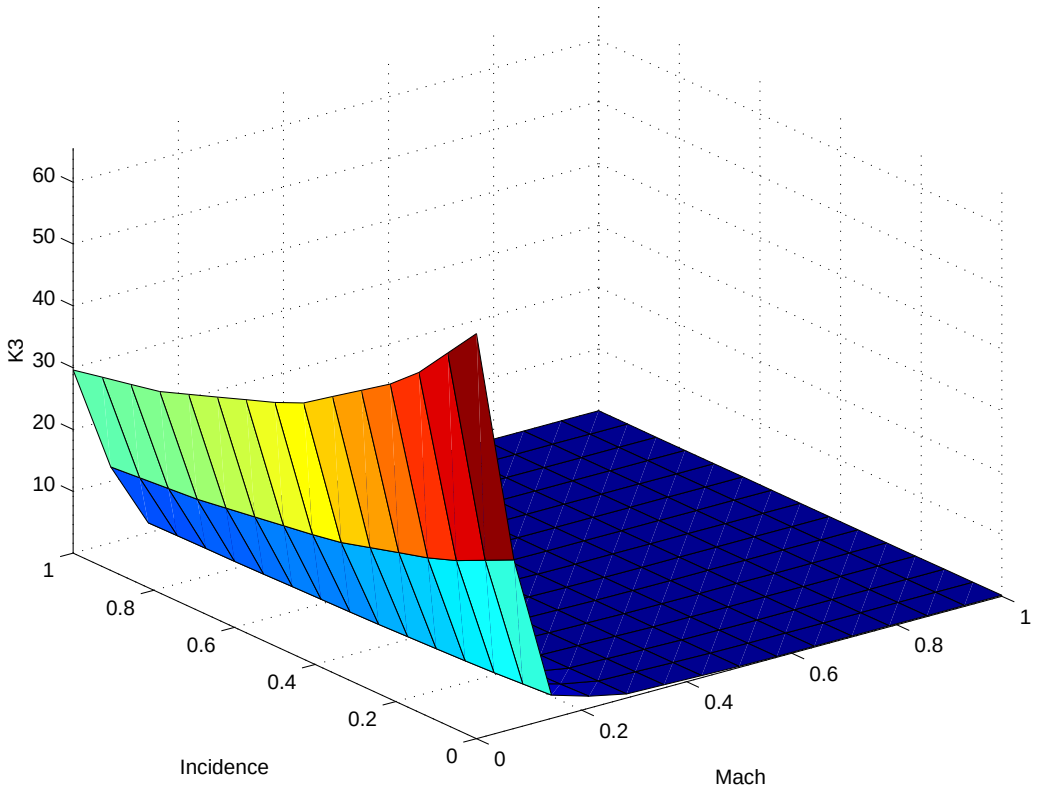

Figure 16: Example gain surface for K_3 and 5 MF per input

9 Conclusions

This paper has shown that a fuzzy pole-placement controller can be designed for complex non-linear systems to produce given performance over a range of plant conditions. The use of evolutionary algorithms to optimise the fuzzy inference system removes the requirement of expert knowledge to design the fuzzy landscape as the multiobjective algorithm is capable of discovering a range of solutions with little designer intervention.

The multiobjective formulation allows many potential solutions to be generated simultaneously. The designer can then choose a candidate solution whilst being informed of what other solutions to the problem may exist.

References

[1] Tsourdos, A., Żbikowski, R., and White, B. A., "Robust design of sideslip velocity autopilot for a quasi-linear parameter-varying missile model," *Journal of Guidance, Control and Dynamics*, , 2000, Accepted.

[2] Rugh, W. J., "Analytical Framework for Gain Scheduling," *IEEE Control Systems Magazine*, Vol. 11, 1993, pp. 799–803.

[3] Kamen, E. W. and Khargonekar, P. P., "On the Control of Linear Systems Whose Coefficients are Functions of Parameters," *IEEE Transactions on Automatic Control*, Vol. 29, No. 1, 1984, pp. 25–33.

[4] Shamma, J. S. and Athans, M., "Guaranteed Properties of Gain Scheduled Control for Linear Parameter-Varying Plants," *Automatica*, Vol. 27, No. 3, 1991, pp. 559–564.

[5] Shahruz, S. M. and Behtash, S., "Design of Controllers for Linear Parameter Varying Systems by the Gain Scheduling Technique," *Journal of Mathematical Analysis and Applications*, Vol. 168, No. 1, 1992, pp. 195–217.

[6] Deb, K., *Multi-objective optimization using evolutionary algorithms*, John Wiley & Sons, , 2001.

[7] Horton, M. P., "Autopilots for Tactical Missiles: An Overview," *IMechE: Journal of Systems and Control Engineering*, Vol. 209, 1995, pp. 127–139.

[8] Wise, K. A., "Comparison of 6 Robustness Tests Evaluating Missile Autopilot Robustness to Uncertain Aerodynamics," *Journal of Guidance, Control and Dynamics*, Vol. 15, No. 4, 1992, pp. 861–870.

[9] Horton, M. P., "A Study of Autopilots for the Adaptive Control of Tactical Guided Missiles," Master's thesis, University of Bath, , 1992.

[10] Tsourdos, A., Blumel, A., and White, B. A., "Trajectory Control of a Nonlinear Homing Missile," in *14th IFAC Symposium on Automatic Control in Aerospace*, 1998 pp. 118–123.

[11] Tanaka, T. and Sugeno, M., "Fuzzy indentification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, No. 1, 1985, pp. 116–132.