Evolutionary Design of Robust Flight Control for a Hypersonic Aircraft

by

Kevin J. Austin, B.E. Mechanical (Hons)

Department of Mechanical Engineering
The Univeristy of Queensland
Brisbane, Australia

A thesis submitted for the degree of
Doctor of Philosophy
June 2002

Statement of Originality

The work presented in this thesis is, to the best of the author's knowledge and belief, original and the author's own, except as acknowledged by reference or otherwise. None of the material contained in this thesis has previously been submitted, either in whole or in part, for a degree at The University of Queensland or any other institution.

All of the simulation software referred to in this thesis is original and was written by the thesis author, except where acknowledged.

Kevin J. Austin

Abstract

An evolutionary design approach is used to construct an autopilot for a hypersonic air-breathing aircraft. Flight control for this class of vehicle is an extremely challenging problem due to the combination of nonlinear dynamics, parametric uncertainty and complex constraints. Consequently, simultaneous control over the flight path, aerodynamic attitude and propulsion is required.

This thesis develops and applies a design procedure which can explicitly address the challenges of hypersonic flight control. The principal computational results of this thesis focus on the capability of an evolutionary based optimizer to design, without *a priori* knowledge, a robust fuzzy control law for a hypersonic vehicle concept. This work is not meant as an expression of the superiority of a particular control approach or an optimization procedure. Rather, it experiments with the potential of fuzzy control to represent a complex, nonlinear, and robust control function, the incorporation of robustness features in the control performance measure, and the capability of the genetic algorithm as a search procedure. The structure of the fuzzy rule base defines the mapping procedure and the design procedure learns the output profile through a numerical optimization procedure. The evolution of the controller design requires the definition of a scalar objective function which assesses the merit of the particular control solution being tested. For this work the design objective is extracted from a collection of simulated flight responses. Such an approach is computationally demanding, but the benefits are that fewer simplifying assumptions are required in the flight dynamics and aero-propulsive models. There is also the capacity to represent features in the objective function which encourage the development of a robust control law. These include the evaluation of the flight response at many points along the trajectory, the full range of expected attitude and control states, and the inclusion of realistic variations in engine operation, vehicle aerodynamics, and physical properties. Essentially, the controller can be configured based on the best available and most practical model of the system. Stability and performance robustness are therefore a natural derivative of the design exposure to the varied performance of the system.

A conventional autopilot structure has been used for the longitudinal motion study. An outer guidance loop provides vehicle attitude commands for trajectory maintenance, while an inner-loop attitude controller tracks the commanded attitude and provides stability augmentation. The control design focus is on the specification of the control function for

the inner-loop. Aside from the evolutionary based design, the second prominent feature of the control application is the parameterization of the attitude controller through a fuzzy rule base. A fuzzy controller has been used for its inherent robustness, and its simplicity in representing a nonlinear control function. The main performance benefit over a constant gain linear controller is derived from the capacity to locally manipulate the control surface of the fuzzy controller during the design. This allows rapid attitude response while still providing the appropriate control authority about the trimmed condition. In addition, the control surface can be configured to any nonlinearities which are a function of the control inputs.

The development of the flight simulation models and the control design procedure are described in detail in the thesis. For the flight simulator, particular attention was paid to a realistic representation of the flight dynamics behaviour through an aero-propulsive simulation module and a dynamics formulation that used the full six degree-of-freedom equations of motion for flight about a spherical, rotating Earth. Successful application of the evolutionary control design procedure to the hypersonic vehicle is demonstrated through a series of design experiments. These cover some of the many variants available with both the specification of the control function and the application of the genetic algorithm. Within this scope, the benefits and potential pitfalls of the overall procedure are considered. Significantly, the genetic algorithm is able to capture the necessary control features for a design of large dimension, with relatively few function evaluations. To provide guarantees of performance and stability robustness, the fuzzy controller must be assessed against an extensive set of test conditions throughout the design process.

As part of the numerical experiments it was found that to achieve good quality control designs, a modification to a well know non-uniform mutation operator was required. This minor enhancement to the genetic algorithm greatly improved the quality of the control solution. The search performance benefits have also been demonstrated on a collection of standard minimization test problems, as documented in Appendix A.

List of Publications

[1] Austin, K. J. and Jacobs, P. A., "A Newtonian solver for hypersonic flows," Technical Report 5/96, Department of Mechanical Engineering, The University of Queensland, 1996.

[2] Smith, A. L., Johnston, I.A., and Austin, K.J., "Comparison of numerical and experimental drag measurement in hypervelocity flow," *The Aeronautical Journal*, November 1996: pp 473-480.

[3] Austin, K. J. and Jacobs, P. A., "Can trained monkeys design flight controllers for hypersonic vehicles," *The 10th Biennial Computational Techniques and Applications Conference*, Brisbane, Australia, 2001.

[4] Austin, K. J. and Jacobs, P. A., "Application of genetic algorithms to hypersonic flight control," *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, Vancouver, Canada, 2001.

[5] Stewart, B., Morgan, R. G., Jacobs, P. A., Austin, K. J., and Jenkins, D. M., Establishment of test conditions in the RHYFL-X facility, *37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibition*, AIAA Paper 2001-3678, 2001.

[6] Austin, K. J. and Jacobs, P. A., "Numerical flight Simulation of a hypersonic airbreathing vehicle," Technical Report 2002/3, Department of Mechanical Engineering, The University of Queensland, 2002.

[7] Austin, K. J. and Jacobs, P. A., "A real-coded genetic algorithm for function optimization," Technical Report 2002/4, Department of Mechanical Engineering, The University of Queensland, 2002.

[8] Austin, K. J. and Jacobs, P. A., "Benchmark Control Problem - Evolutionary Design of a Robust Controller," Technical Report 2002/5, Department of Mechanical Engineering, The University of Queensland, 2002.

Acknowledgements

> “All things move toward their end, of that you can be sure”, *Song of Joy*, Nick Cave.

The work in this thesis has been made possible through the support of colleagues, friends, and family members. I wish to formally acknowledge their support and pay special thanks to the role they have played in helping to complete this project.

To my friend and supervisor Peter Jacobs: a mentor, a massager of egos, and problem panacea. I envisage eternal gratitude for the guidance and support he and his family have provided.

For their discussions on all things relating to this work and many far removed topics, I sincerely thank the past and present postgraduates from our research group: Andrew McGhee, Adrian Smith, Paul Petrie-Repar, Ian Johnston, Leonard Chan, Mark Sutcliffe, and Chris Craddock.

To the numerous people who have suffered my company outside of work, it is likely I have gained much more from the relationship then they have: Mike Lubke, Samantha Rayment, Mark Wallace, Jae and Marcus from the University Pool (the enigma of efficient/fast swimming continues to be an intensive but pleasant distraction from work), and Gavin and Jeremy from the Uni Bike Shop, for providing an escape from the world of rocket science.

To my family, for their reassuring confidence in the inevitable conclusion of this work. In particular I’d like to thank my mother for her unwaivering faith in me, my father who I deeply admire, and my brother Gary for his friendship, of which I’m truly grateful.

The Australian government afforded me financial support through an Australian postgraduate award, a Department of Mechanical Engineering Scholarship, without which this research would not have been completed. Additional support from Peter Jacobs during the concluding stages of this thesis is also greatly appreciated.

To Peter Jacobs: the importance of Peter’s contribution over the years deserves repeated recognition.

Kevin J. Austin

Contents

List of Figures

List of Tables

Nomenclature

a	Speed of sound, m/s
$\boldsymbol{a}$	Acceleration vector, m/s^2
A_C	Combustor area, m^2
cm_x	Center of mass position, m
c_p	Contant pressure specific heat
$\boldsymbol{C}$	Chromosome vector
C_M	Linear scaling factor
$\boldsymbol{d}$	Disturbance vector
$\Delta u, \Delta w$	Turbulence velocities, m/s
e	Expected value
$\boldsymbol{e}$	Error input
f	Fuel/air mixing ration
f_{st}	Stoichiometric fuel/air mixing ration
$\boldsymbol{F}$	Force vector, N
F	Filter function
F_{att}	Vehicle attitude control function
F_{B}	Body fixed reference frame
F_{G}	Vehicle guidance function
F_{obj}	Objective function
g	Gravity, m/s^2; Generation number
g_0	Gravity at sea level, m/s^2
GA	Index for genetic algorithm application
h	Geopotential altitude, m
h_a	Absolute altitude, m
h_G	Geometric altitude, m
$\boldsymbol{h}$	Angular momentum vector, kg·m^2/s
H	Fuel energy density (heating value), J/kg-fuel; Population entropy
I	Inertia, kg·m^2
J_α	Performance measure for settled angle of attack response
$J_{\int \alpha}$	Integral of absolute error performance measure
J_q	Performance measure for pitch rate response

J_{t_f}	Performance measure for simulation time
J_h	Performance measure for the altitude response
$\boldsymbol{K}$	Feedback gain vector
$\boldsymbol{L}$	Geometric transformation matrix
L_i	Lipschitz constant
m	Vehicle mass, kg
$\dot{m}$	Fuel flow rate, kg/s
M	Mach number
M_1	Upstream Mach number
M_2	Downstream Mach number
M_∞	Freestream Mach number
$\boldsymbol{M}$	Moment vector, Nm
n_x, n_z	Components of surface unit normal (cosines)
N_C	Number of completed test simulations
N_I	Number of control inputs
N_G	Maximum number of generations (search length)
N_r	Number of control rules
N_P	Population size
p	Flow pressure, Pa
P	Surface pressure, Pa
$\boldsymbol{P}$	Population array
q	Dynamic pressure, Pa; heat addition, J/kg; pitch rate, rad/s
r	Random number
$\boldsymbol{r}$	Radius vector, m
R	Gas constant, $J/kg\ K$; Radial position of vehicle, m
R_E	Radius of the Earth
S	Surface area, m^2
T	Temperature, K; Trajectory position index
T_{01}, T_{02}	Upstream and downstream total temperatures
T_i	Flight condition index
U	Uniform random deviate
u_α	Inner loop attitude command
u_{e}	Elevator command
V	Vehicle velocity, m/s
$\boldsymbol{v}$	Velocity vector, m/s
$W_{0,1}$	White noist with 0 mean and unit variance
$\boldsymbol{x}^*$	Solution vector for function minimization problems
x_B, z_B	Body-fixed coordinate axes

Greek and Miscellaneous Symbols

α	Angle of attack, rad
β	Shock angle with respect to the upstream flow direction, rad
	Mutation strategy parameter
γ	Ratio of specific heats c_p/c_v
	Mutation fine-tuning parameter
Δ	Parametric uncertainty function
	Perturbation function for mutation operator
$\Delta\boldsymbol{V}$	Turbulence velocity vector, m/s
η_c	Combustion efficiency
λ	Latitude, rad
μ	Longitude, rad
	Set membership function, $\mu(x)$
θ	Flow deflection angle, rad; Pitch angle
θ_e	Elevator angle, rad
$\theta_{e,\text{cmd}}$	Elevator actuation command, rad/s
$\theta_{e,\text{trim}}$	Elevator trim angle, rad
ρ	Density, kg/m^3
σ_L, σ_U	Bounds for mutation operation
ν	Prandtl-Meyer function
	Fuel input settings
ϕ	Fuel equivalence ration
(ψ, θ, ϕ)	Euler angles
ω	Angular velocity, rad/s
ω^E	Earth rotation velocity, rad/s

Subscripts

cmd	Command
E	Earth reference frame
err	Error
ref	Reference value
O	Origin of reference frame
U, L	Upper and lower engine modules
unc	Based on a vehicle model with performance uncertainty

Superscripts

E, B, V	Earth, body, and vehicle reference frames

Acronyms

1D	One dimensional
2D	Two dimensional
CFD	Computational fluid dynamics
CGLF	Constant gain linear feedback
EA	Evolutionary algorithm
FACDS	Flight and Control Design Simulator
FC	Fuzzy controller
GA	Genetic algorithm
HABV	Hypersonic air-breathing vehicle (system model)
HXRV	Hyper-X hypersonic research vehicle
HyShot	UQ scramjet flight test program
HySTP	Hypersonic Systems Technology Program
IAE	Integral of absolute error
IMU	Inertial measurement unit
LQR	Linear quadratic regulator
LTI	Linear time invariant
NASA	National Aeronautics and Space Administration of the United States of America
NASP	National Aerospace Plane (US)
RCGA	Real-coded genetic algorithm
SCRAMJET	Supersonic combustion ramjet
SISO	Single input single output
MIMO	Multiple input multiple output
MIT	Massachusetts Institute of Technology
UQ	The University of Queensland
US	The United States of America

CHAPTER 1

Introduction

In Germany during the second World War Eugen Sänger and Irene Brendt conducted research on a long range antipodal bomber. The Sänger "Silver Bird" vehicle was to be launched from a rocket driven sled, climb to an altitude of 300 km, and descend through a series of aerodynamic skips off the atmosphere [186, 6], in a manner similar to the operation of today's space shuttle. Sänger had first proposed the winged hypersonic vehicle in the late 1920's. His work was the genesis of hypersonic research, inspiring the "higher and faster" pursuit of the aerospace industry to achieve manned lunar return and the space shuttle. Ultimately, as a long term project the Sänger space-plane was abandoned for the V1 and V2 missiles favoured by the German military.

Hypersonic flight is defined by a Mach number greater than 5, representing a flight speed five times the local speed of sound in the atmosphere. In 1961 the rocket powered North American X-15 was piloted to Mach 5.3 [6], becoming the first hypersonic aircraft. The vehicle was an essential stepping stone to realizing the potential of rocket powered aircraft which ultimately led to the space shuttle. Rocket propulsion remains today as the only mechanism for achieving hypersonic speeds and for launching objects into earth orbit.

Modern rocket propelled launch systems operate close to theoretical limits. Their continued development is driven by a demanding space market which covets reliability, flexibility, and a reduction in the cost of raising payloads. In pursuit of the lucrative satellite market, a broad range of rocket propelled space transporters have been developed, offering variations in size, staging, operation, and launching [108]. They are however constrained by large infrastructure requirements and a payload penalty incurred from the need to carry oxidant for combustion. Presently, the only foreseeable practical alternative to chemical rockets is the air-breathing supersonic combustion ramjet engine, or scramjet engine. A popular contemporary vision of scramjet application is the civilian transporter, capable of cruising at speeds three to five times greater than current propulsion capabilities [95]. Despite the allure of high speed passenger flight, the primary motivation for the development of hypersonic air-breathing vehicles is their potential as transatmospheric aircraft, capable of accelerating to orbit. Advanced launch systems utilizing scramjets promise to make space more accessible by reducing the cost of inserting payloads into orbit.

Aerospace engines are customarily rated using specific impulse, or thrust per weight of fuel per second. A convenient map of the operational capabilities of high speed propulsion options may be formed by charting the variation of specific impulse with Mach number. Figure 1.1 shows the performance of rockets and existing air-breathing engines, along with predictions for the operation of a scramjet engine. Air-breathing engines in general, offer superior cycle-efficiency compared to chemical rocket engines, because the oxygen needed for operation is captured from the atmosphere, instead of being carried on board. Scramjets, in particular, circumvent the material temperature limitations encountered in ramjets and turbomachinery, and allow more efficient utilization of propellant than chemical rockets [12]. In Figure 1.1 these two features appear as an extension of air-breathing capabilities to hypersonic Mach numbers and, compared to the hydrogen and hydrocarbon fuel rockets, have a higher specific impulse at high Mach numbers. The high specific impulse of a scramjet translates to a capacity to accelerate more mass than a rocket of the same size [118]. The weight saving can be used to provide a better payload fraction while allowing a greater "empty weight" which, in turn, can be directed towards augmenting vehicle ruggedness and flexibility of use. Realizing these capabilities would be a big step towards improving the performance and efficiency of space launch vehicles.

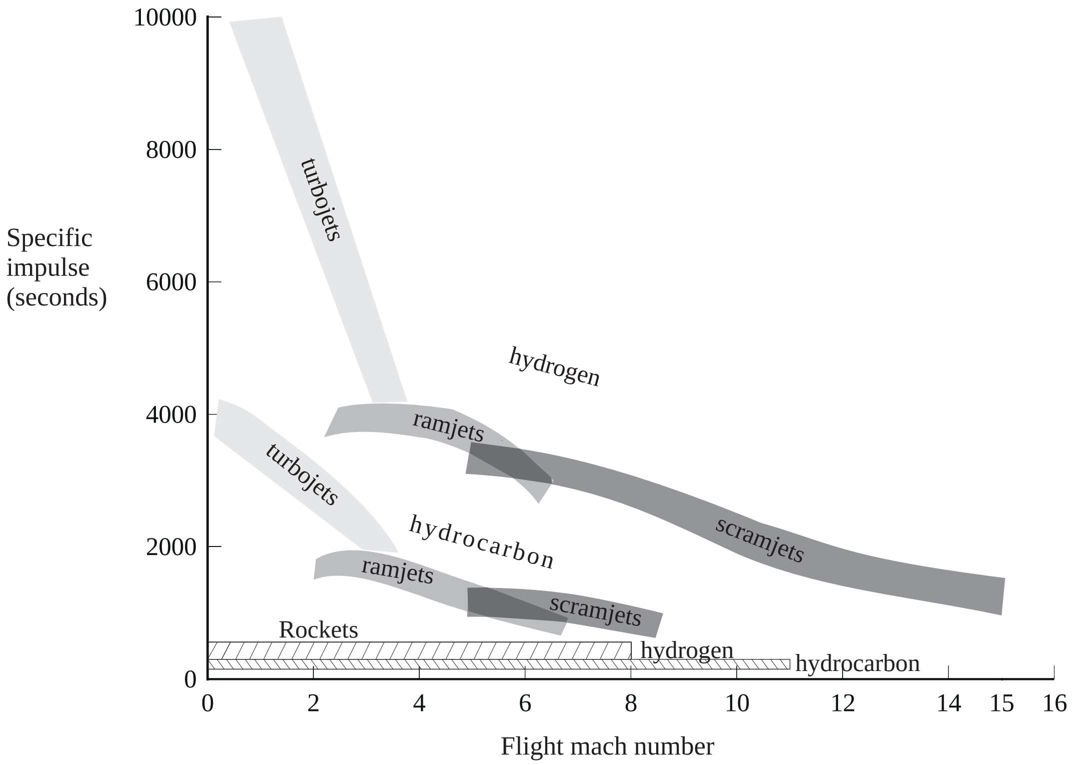


Figure 1.1: Vehicle propulsion alternatives for high speed flight. For each class of propulsion system both hydrocarbon and hydrogen fueled performance is shown. (Source: References [118, 121, 105])

Serious consideration of scramjets for hypersonic flight followed the post-war period of development of hypersonic vehicle concepts [118]. The pioneering period was the 1950s and 1960s, beginning with the demonstration of stable supersonic combustion. Dorsch and others at NACA Lewis Labs, used experiments on external and internal combustion in supersonic airstreams [29] to catalyse the development of the ramjet cycle using supersonic combustion. With the first steps made, many researchers and organizations made significant contributions towards the development of the ramjet cycle using supersonic combustion. For example, Weber and Mackay [226], Ferri [71, 70], and Swithenbank [214] helped establish the basic concepts behind scramjet operation. Several important aspects of scramjet operation were addressed, centering on the practical issues of supersonic combustion and establishing methods of analysis. They included the behaviour of supersonic combustion flames, chemical reaction processes, fuel-air mixing, multidimensional interaction between combustion and fluid dynamics. Simultaneous with the development of the propulsion system was the evolution of a new aerospace vehicle configuration. To address the unique requirements of the inlet, combustor, and nozzle, the airframe and propulsion system formed an integrated design, leading to some configurations being dubbed "flying engines". The extreme conditions associated with hypersonic flight have also promoted the development of lightweight high temperature resistant materials, active cooling of the vehicle structure, dual subsonic and supersonic combustion capability, intelligent trajectory and propulsion controls, simulation codes, and ground testing facilities [95].

The supersonic combustion ramjet remains a conceptually simple concept. Despite this, the development period resembles that which preceded the Wright brother's first powered flight, rather than the 20 - 30 year evolution period post 1900. Forty years of hypersonic air-breathing research has been dominated by engine related issues. Only recently was a small scale scramjet demonstrated to produce net thrust in an experimental facility. In 2001 there were engine-centered flight tests by the American Hyper-X program and the Australian HyShot program. Such tests are an important stepping stone to a practical scramjet powered vehicle suitable for sustained free flight.

Along with engine/airframe research, another key enabling technology is flight control. It is this challenge that provided the stimulus for this thesis on *the application of intelligence based methods to the longitudinal flight control of a scramjet powered launch vehicle*. The vehicle concept follows the proposal for the development of an Australian small scale launch vehicle, with a scramjet powered stage [204, 194]. Closed-loop flight control is required to stabilize the vehicle, provide trajectory maintenance, and stability robustness against performance uncertainty, constraints, and disturbances. With the vehicle additionally defined by highly nonlinear time-varying dynamics, it is generally accepted that most conventional linear control system methodologies are unsuitable [39].

Hence the application of methods borrowed from intelligent control, which promise the ability to design a complex non-linear controller capable of robustly dealing with variations and uncertainty in vehicle and propulsion performance. Working with a conventional longitudinal autopilot structure, the inner loop is represented by a fuzzy logic rule base while the outer loop guidance is provided by linear feedback. They are designed sequentially using a genetic algorithm (GA) to maximize flight control performance. The design procedure builds robustness through the controller performance randomly sourcing simulated flight responses. These responses can include system model uncertainties, constraints and disturbances. With a noisy, evolving objective function, and a potentially large number of control parameters to configure, it is a demanding optimization problem. GAs are a general global search algorithm inspired by natural evolution. They have gained a reputation for their robustness in the presence of noise, and their ability to search highly non-linear, multimodal, and multivariate problems. As with any brute force approach however, there is a potential for the design to be computationally expensive.

The remainder of this chapter introduces the hypersonic air-breathing vehicle, and explores issues important to high speed flight and control. A formal statement of the research objectives is then provided, together with a guide to the remaining chapters.

1.1 Issues of Hypersonic Flight Propulsion

Convention separates supersonic and hypersonic flight regimes by associating hypersonic aerodynamics with flows greater than Mach 5 or, for example, 1500 km/hr at an altitude of 30 km. The distinction represents an increased importance at Mach numbers much greater than 1, of physical flow effects such as viscous interaction, high temperatures, and low density flow. In a laboratory setting, hypersonic is a term often used when describing high speed flows in wind tunnels where the high Mach number is associated with a low stream temperature, as is typical of large “blow down” tunnels [20]. In real high speed flight such as atmospheric reentry, the high temperatures encountered through airstream interaction with the vehicle are important for vehicle design and performance. The term hypervelocity flow is then used to describe the generation of these high energy flows, where there are both high Mach numbers and high temperatures. Since this thesis is concerned with flight conditions where the term hypersonic is typically applied, there is no further distinction made with these definitions.

Air-breathing engine selection for hypersonic flight essentially deals with the thermal barriers imposed by structural heating and heat release [227, 118]. In Figure 1.1 the performance range of several air-breathing engines indicate a transition from turbojets at low Mach numbers, to ramjets up to low hypersonic and finally to scramjets for hypersonic flight. The Mach 3 limit for turbojets represents a constraint on the turbine inlet temperature, which increases with Mach number, ultimately compromising structural integrity. As the Mach number increases, the continued drop in turbojet cycle efficiency gives the advantage to ramjets. Assuming forecasted technological improvements are met the operation range of hydrogen fueled ramjets extends to a maximum flight speed of around Mach 7. This operational limit is imposed by the heat release accompanying the slowing down of the highly energetic airstream to locally subsonic conditions prior to combustion. Material and structural limits are compromised and the benefits of combustion are reduced through higher initial temperatures and the dissociation of reactants. Both ramjets and scramjets compress the air stream by the forward speed of the aircraft. In ramjets the passive surfaces generate strong normal shock waves with losses that increase with flight speed. Above Mach 6 a scramjet configuration provides less inlet compression, lower shock losses, lower combustion temperatures, and supersonic combustor flow. The lower static temperature and pressure mean less heat transfer to the airframe and lower structural loads, and enable an increased benefit from the burning fuel. Scramjet superiority at hypersonic speeds is thus a result of the thermal and structural advantages of adding heat to a supersonic rather than a subsonic combustor flow.

In addition to bounding high speed propulsion, the extreme thermal environment impacts greatly on the actual engine design and successful operation of the aircraft. The effects of high temperatures in combination with other hypersonic flow features such as shock layers, entropy layer, low density flows, and viscous interaction [6] present many potential problems. For example, interaction between the boundary layer and the inviscid flow reduces the core flow to the engine, and can greatly affect the surface pressure distribution and therefore the lift, drag, and vehicle stability. Also, the high temperature flows can cause dissociation and ionization within the gas and influence the vehicle's aerodynamic parameters. These issues have also inspired the evolution of experimental and computational techniques which are used to predict their influence.

One of the many contributions made through flight testing the X-15 rocket plane [45, 97] was the practical importance of hypersonic effects. A 1967 test flight had a dummy ramjet suspended by a pylon below the X-15 [6]. Localized heating caused by shock wave-boundary-layer interactions prematurely ended the thermal test flight of the axisymmetric ramjet model. The model was completely removed from the pylon when the shock wave from the ramjet nacelle burnt through the connecting pylon surface. Further damage was caused by local surface heating from the pylon bow shock impinging on the bottom of the X-15, allowing the penetration of the hot boundary layer.

The difficulties associated with the hypersonic environment have contributed to the long development time of scramjet technologies. In contrast with the simplicity of the scramjet engine, aerodynamicists continue to face many challenges which obstruct the practical application of scramjet engines.

1.2 The Hypersonic Air-Breather

Air-breathing vehicles generate thrust in direct proportion to the amount of air processed by the engine. A basic scaling argument shows that as the operating speed increases, the cross-sectional area required for the intake becomes a larger fraction of the total frontal area. For hypersonic air-breathing aircraft, there is the additional need of a high dynamic pressure flight trajectory to achieve optimum combustor operation. Therefore, generation of the high specific impulse predicted for scramjets at high Mach numbers, not only brings concerns of heating and structural loads, but also requires the engine intake area to be a large fraction of the vehicle frontal area. The resulting large engine surface area attracts significant drag penalties. Early tests on scramjet concepts showed the high external drag of the engine negated the benefits of the scramjet, if it had to mounted in the traditional manner, isolated from the airframe [118]. More recently, efforts to optimize a Mach 12 axisymmetric scramjet showed that, due to the large viscous forces, the net axial force on the optimized scramjet was actually a drag force [48]. Provided engine designers are able to build a useful vehicle-engine configuration, the drag force will likely remain a large percentage of the total thrust, so that a small increase in engine performance can provide a large increase in acceleration capability.

The basic shape of the hypersonic air-breather is formed by first addressing the inlet processing requirement. Shown in Figure 1.2 are two generic airframe integrated scramjets. Both the accelerator and cruiser configuration have the propulsion system integrated with the airframe structure. In addition to inlet processing, these configurations also address other issues critical to hypersonic operation. They minimize external drag of the airframe and propulsion system combination, minimize the total vehicle weight, and address the cooling requirements of the airframe and engine by reducing the internal surface area thereby reducing the surface area to be protected from high thermal loads. The slender configurations also allow weak bow shocks to be maintained to minimize shock losses within the propulsive flow path.

Major international hypersonic air-breathing projects are being pursued in the United States of America, Europe, and Japan. Their focus is the relatively large, hydrogen-fueled scramjet powered space-plane concept, for which cost effectiveness is achieved through their continual reuse. The cruiser configuration of Figure 1.2 is representative of the space-plane concept. Leading the way for space-plane research is the American Hyper-X project [79]. Early development of this concept was via the NASP (National Aero-Space Plane) and HySTP programs of the 1980s and 1990s [79]. Cancellation of the NASP program led to Hyper-X, a less ambitious sub-scale hypersonic research plane. Presently, it is the only project with the capability of testing the free-flight operation of a scramjet powered vehicle. Hyper-X testing uses a modified Pegasus booster to deliver the vehicle

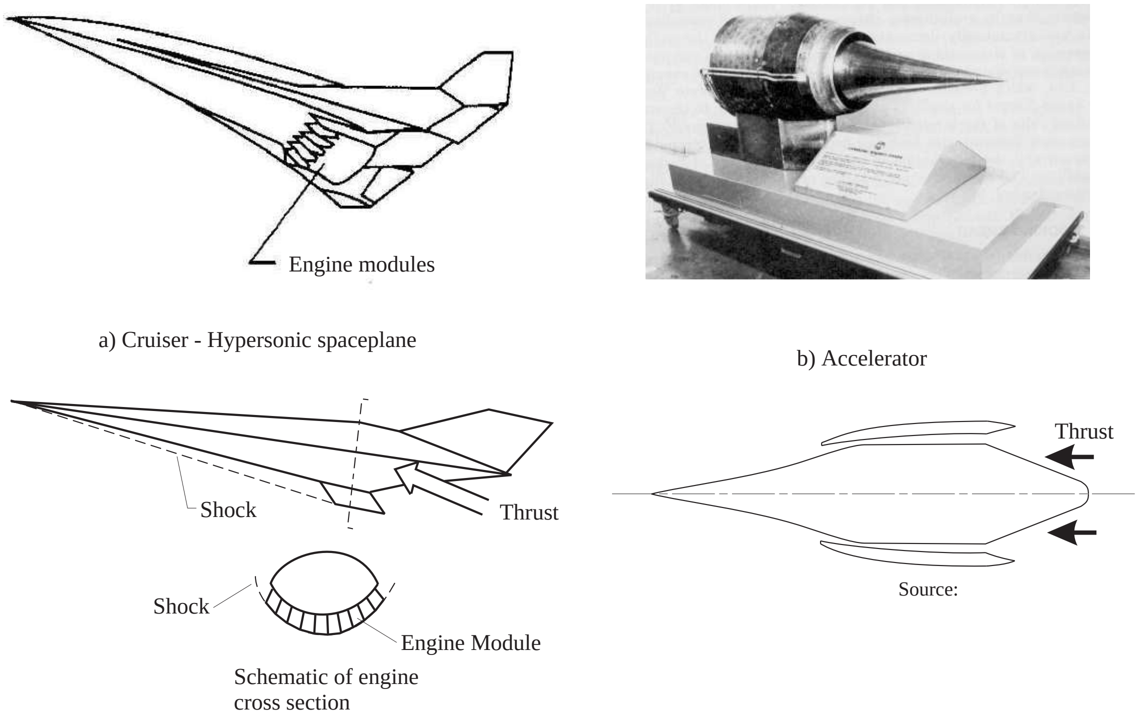


Figure 1.2: Generic hypersonic air-breathing vehicles. (Source: References [227, 97, 221])

to the hypersonic test condition, before separation and free flight.

The large containment volume required when using hydrogen fuel means the spaceplane is a relatively large vehicle. The size of the operational concept vehicle behind the Hyper-X sub-scale project is approximately 61 m, significantly larger than the shuttle orbiter which is 35 m in length. Integration of the airframe and engine is such that the lower portion of the vehicle forms the propulsion system while the upper portion is the airframe. The entire forebody performs the initial compression of the freestream air, with further compression by the inlets of the engine modules. The rear of the airframe is also used, being shaped as a nozzle and producing thrust from the expansion of combustion products. A benefit of using the fore and aft surfaces of the airframe for engine operation is the generation of lift which would otherwise require large wings and attract large drag penalties.

An alternative configuration to the Hyper-X vehicle type is the axisymmetric scramjet accelerator. This vehicle, shown in Figure 1.2 b), maximizes the airflow capture area relative to the airframe area, promoting an adequate thrust margin while minimizing configuration drag at near zero angle of attack. The axisymmetric configuration was used in the first flight tests, of a scramjet engine by the Russian Central Institute of Aviation Motors (CIAM). These tests began in Russia in 1991 with the launch, to Mach 5.5, of an axisymmetric scramjet atop a surface-to-air missile [120, 100]. The test configuration was essentially the same as the American Hypersonic Research Engine (HRE) project. During the 1960s, part of the HRE project involved preliminary tests where an axisymmetric

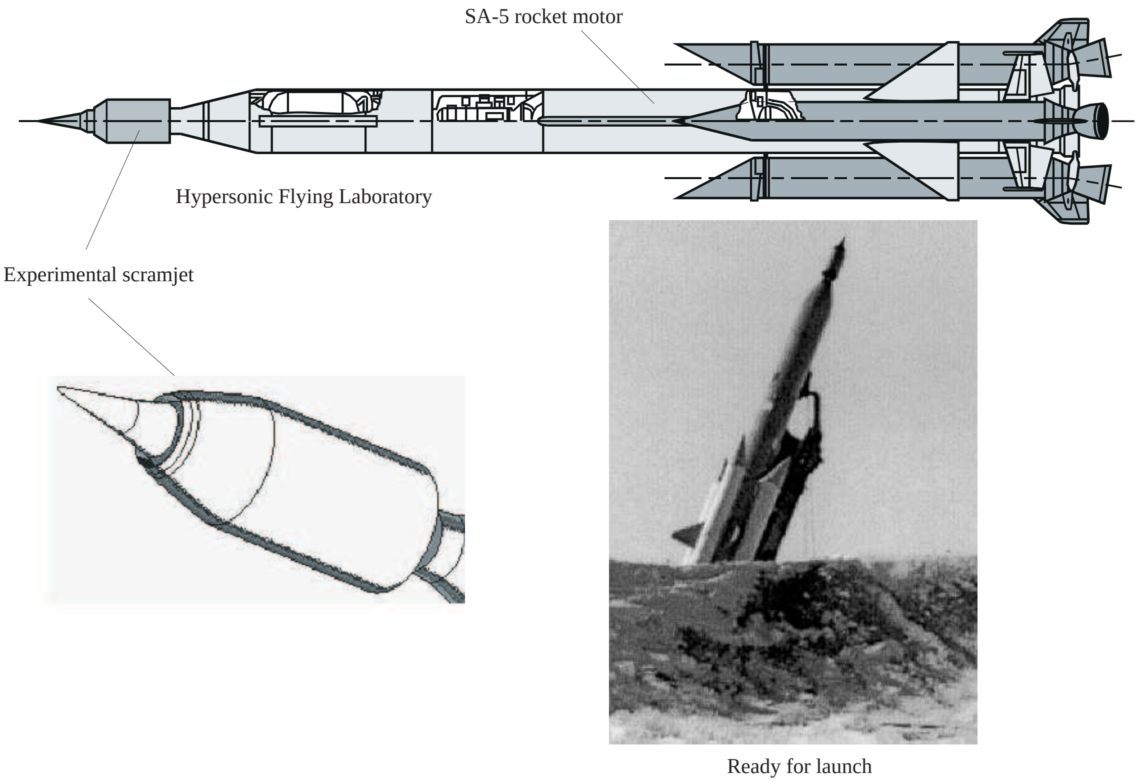


Figure 1.3: CIAM scramjet testing using the Hypersonic Flying Laboratory. (Source: Reference [183])

ramjet was mounted below the X-15 rocket plane [79, 45].

Supporting CIAM's testing was the hypersonic flying laboratory Russia developed using modified surface-to-air missiles [184]. Figure 1.3 shows the arrangement used in the Russian tests. The hypersonic flying laboratory was advertised as being capable of testing a hydrogen-fueled scramjet up to Mach 6. A number of tests were performed with the aim of collecting data to validate predictive numerical codes. The tests also served to provide validation of systems such as the fuel supply. Using a truncated nozzle, thrust was not measured directly, but sustained supersonic combustion at hypersonic flight speeds was reported as being achieved [46, 97, 100].

The Russian flight test capability separately attracted France and America. France's vested interest was their own high speed air-breathing technologies program which began in the 1950s with an emphasis on missile applications. Today though, there is the addition of a long term goal of a single stage to orbit plane. The CIAM testing provided the first steps to validation of the scramjet and its real propulsion capacities [69, 57, 119]. America was also attracted by the opportunity to cost-effectively obtain valuable flight data for their space-plane program. The National Aeronautics and Space Administration of the United States of America (NASA) worked with CIAM to reassess the scramjet design used in the Franco-Russian tests. The higher heat loads associated with full scramjet operation at a higher Mach number, required redesign of the combustor and active cooling system. The culmination of the NASA and CIAM contract was a flight test in 1996 which achieved a new maximum flight Mach number of over 6.4 and provided useful data for the Hyper-X project [183].

At The University of Queensland (UQ), scramjet research has, since the early 1980's, prompted the development of ground based testing facilities and computational modelling capabilities. The basis for scramjet development was the potential of an axisymmetric configuration operating as an acceleration stage in a small launch vehicle application. UQ researchers within the Department of Mechanical Engineering, were the first to demonstrate, in a shock tunnel, a flight-style scramjet generating more thrust than drag [171]. Presently, a UQ team heads an international effort to flight test the supersonic combustion process in a scramjet, under the banner of the HyShot program [170]. For these hypersonic air-breathing experiments, a two-dimensional scramjet is fitted to a Terrier/Orion rocket. The experiments use a flight trajectory which provides the capability of testing at a flight speed of Mach 8. Extensive instrumentation is installed to measure the pressure rise produced by supersonic combustion, allowing correlation with measurements from shock tunnel experiments performed at the university.

An early application concept, shown in Figure 1.4, uses the conventional scout rocket configuration with a scramjet powered second stage [194]. Endorsement for this type of scramjet application is provided by the scaling argument presented by Stalker [204]. The basis for the analysis is a comparison of the cost to payload ratio over a wide range of payloads, for traditional all-rocket launchers and one using a scramjet powered second stage. The scaling argument reinforced the belief that scramjet-powered vehicles are relatively insensitive to scale effects, compared to the quite strong increase in cost with payload for rockets. As such, it was considered that a scramjet powered launch vehicle capable of placing a relatively small 1000 kg payload into low-earth orbit, could be operated competitively against all-rocket launch vehicles.

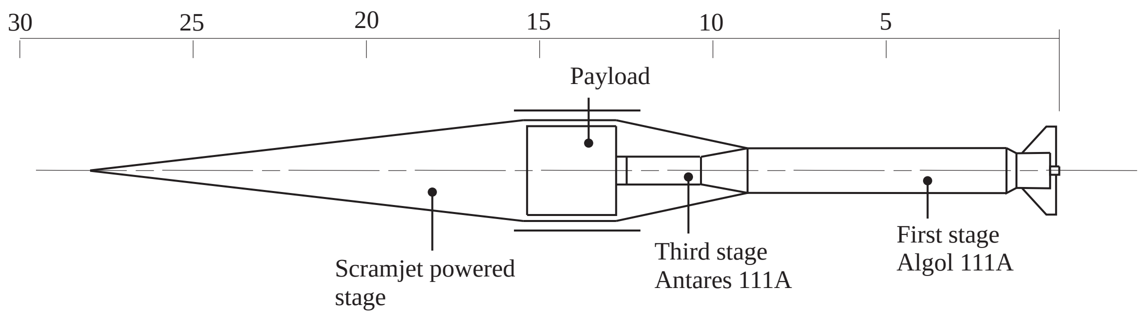


Figure 1.4: A small payload launch vehicle concept formed by integrating a scramjet powered second stage with a conventional scout rocket configuration.

One of the benefits of the axisymmetric design for the small launcher application, is the potential for integration with conventional rocket stages [113]. In addition, changes in angle of attack would not be required to balance the variation in thrust level with increasing velocity and altitude. Figure 1.5 shows a baseline design for the scramjet powered stage, developed and tested in the Department of Mechanical Engineering at UQ [229, 230]. A conical forebody delivers the hypersonic freestream air into six scramjet engines modules which surround the centre body. This allows control of the circumferential distribution of mass flow, and provide the means for differentially throttling the ducts as a method of controlling the vehicle attitude. In contrast, the alternative of a single axisymmetric flow path would, with small angles of attack, have the bulk of the air mass flow on the lee side, making the vehicle very hard to control. The axisymmetric configuration of engine modules is designed to run optimally at zero angle of attack at roughly constant altitude. Conversely, non-zero angle of attack operation produces an uneven distribution of freestream flow amongst the engine modules. Such off-design operation means a significant impairment of engine performance and vehicle operation, and generates large destabilizing moments. Successful operation therefore requires accurate attitude control of the vehicle in level flight.

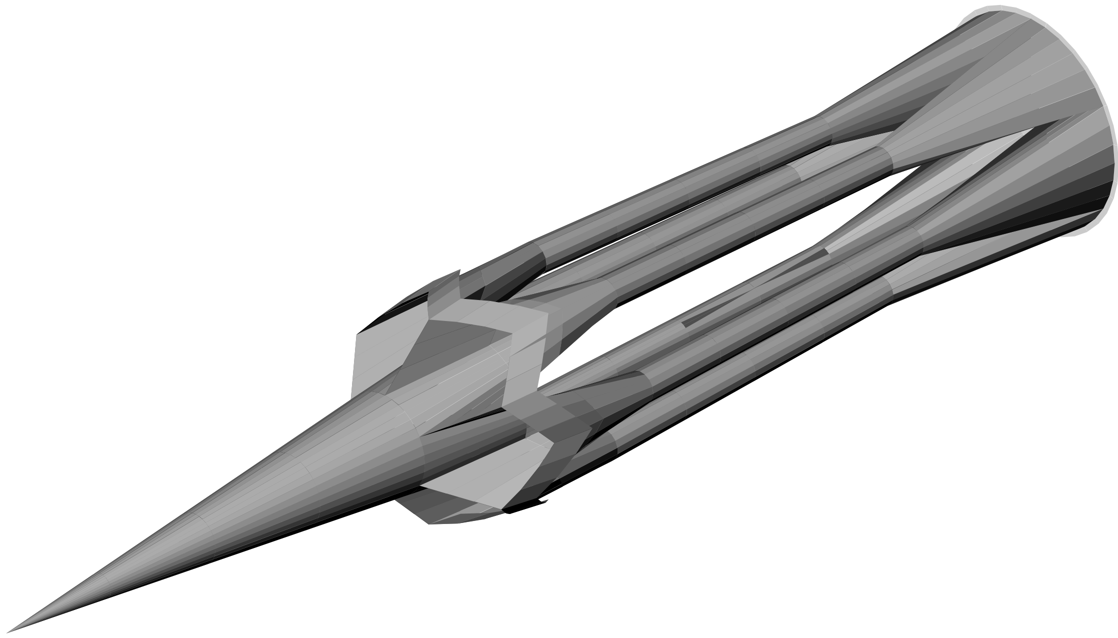


Figure 1.5: Baseline design for the scramjet powered stage. The axisymmetric scramjet propulsion system is shown with the cowl removed. It features a conical forebody, swept inlets providing three-dimensional compression, and scramjet modules with cylindrical combustors.

Operation of the axisymmetric scramjet as depicted in Figure 1.6, is conceptually simple. There are no moving parts within the engine flow path and flow processes are neatly divided into compression, combustion, and expansion. The conical forebody redirects the freestream into the combustor through a series of oblique shocks, raising the temperature and pressure of the airstream. Fuel is mixed with the supersonic airstream, undergoes turbulent mixing and “auto-ignites” due to the high temperature. After traveling the length of the combustor, the combustion products and unburnt fuel are expanded by the vehicle aft-body. Net thrust is then the difference between the thrust generated by the expansion of exhaust gases and the total drag of the engine. Practical operation of such an engine, in contrast to the simple flow structures shown in Figure 1.6, remains an enigma. What is certain is that continued advancement of simulation capabilities is required to provide analysis of the complex flow interactions which makeup hypersonic aerodynamics and supersonic combustion. Also required are innovative methods to limit the impact of skin friction on the overall performance, and the effect of boundary layer development on the performance of the engine.

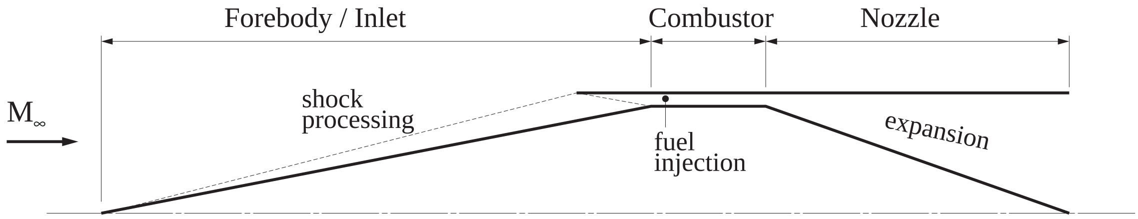


Figure 1.6: Flow schematic of the fully integrated axisymmetric scramjet.

1.3 Early Flight Control

Around the turn of the nineteenth-century, George Cayley established the fundamental defining concepts of an airplane. The majority of modern aircraft still conform to his basic definition: a machine with fixed wings, a fuselage, and a tail, with separate systems providing lift, propulsion, and control [115]. Despite Cayley's insight and decades of aerodynamic research, it wasn't until the later half of the nineteenth century that the real surge toward powered flight began. At this stage, powered human flight was no longer a flight of fantasy, and engineers began approaching it as just another technical challenge. This era of research has been popularized in museums around the world, and through numerous texts [115, 2, 218, 7, 9].

It is worth noting several of the contributions during the Wright era, to contrast with the development of the scramjet vehicle and hypersonic flight control.

There was a seemingly widespread belief among scientists of the late nineteenth-century that enough was known about aerodynamics to achieve powered flight. They were confident that the flying problem could be simply tamed by placing a large enough engine on a strong enough airframe capable of generating lift [115, 7]. Hiram Maxim belonged to this group of thinkers. He assembled a huge four-ton machine powered by two very efficient steam engines, but without many of the elements needed for practical flight. Likewise, Clement Ader attempted to fly a steam-powered aircraft whose only method of control was a highly impractical method of moving the wings fore and aft.

Both Maxim (1890's) and Ader (1890), through their brute force approach, managed to briefly lift their vehicles during tests. Maxim's vehicle crashed after it managed to break free from the guard rail of his test track. To his credit though, a control augmentation device was installed, in apparent recognition of the inherent longitudinal instability of the vehicle [2, 148].

A more productive approach was followed by Samuel P. Langley. Ironically, it also resulted in the most heralded aeronautical failure of the era. Langley began with flight tests on models, conducting a number of successful tests with one-quarter scale, unpiloted steam-and-gasoline- powered aircraft models. Unfortunately for Langely, his success ended when the piloted full-scale version of his successful models collapsed following takeoff. As an internationally respected scientist he attracted considerable ridicule for the failure of his much publicized, late 1903 attempts.

In contrast with both Langley and Maxim, Otto Lilienthal believed it was necessary to obtain a feel of the airplane and thereby understand its flight properties, before attempting to fly with an engine. In this respect he followed Cayley's lead of performing aeronautical studies using gliders. Lilienthal recognized the need for control but his method of shifting mass was limited. The other great limitation of his design was the use of flapping wings for propulsion. Despite many successful flights, he died in 1896, as a result of injuries sustained when a wind gust stalled his glider and it fell to the ground. However, the "aviator" philosophy established by Lilienthal contributed significantly to the aerodynamics and practical design of an airplane, and ultimately inspired the success of the Wright Brothers.

The Wright brothers, like Lilienthal, were concerned with what was needed to fly, rather than the principles behind it. Combining established aerodynamic principles with their own research, they established the design features needed to fly an airplane. The key obstacles as they saw it were the wings for generating lift, a means of propulsion, and a method of balancing and steering. The first two elements had already been well investigated and were considered to be attainable and rapidly developed as required. As Wilbur Wright put it during a presentation before the Western Society of Engineers:

> "... When this one feature (balance and steering) has been worked out, the age of flying machines will have arrived, for all other difficulties are of minor importance." Wilber Wright speaking before the Western Society of Engineers, September 1901, as quoted in [115], page 48.

To address the control problem the Wrights followed the aviator philosophy, learning to fly controlled gliders. Dynamic stability within controllable limits set the boundary for their research. This approach was perhaps a benefit of their experience with bicycle building, and provided the concept of a stable system consisting of the machine and the pilot, rather than simply a stable airframe. Nine days after Langley's failed attempt, the Wrights' research culminated with the epochal flight of the inherently unstable Wright Flyer. It showed that with relative safety, one could fly a powered aircraft.

In the century following the Wright brothers' success, control technology has evolved from a pilot dependent era to an era of computerized automatic control of both piloted and unpiloted vehicles. There was a brief foray into aircraft with inherent stability, as the demanding task of flying as performed by the Wright brothers was not practical for early aircraft with lesser pilots. Also, the more stable an aircraft, the easier the autopilot was to design. Abandoning inherent stability in favour of dynamic stability followed the realization of the performance gains available through the reduction in the weight and drag from stabilizing surfaces. Active flight control has since, been at the forefront of developments in control theory. From the Wright Flyer to the Space Shuttle these advances have seen the application of methods such as gain scheduling, adaptive control, robust control, and optimal control techniques.

The "higher and faster" ideology has catalysed aviation innovation for a hundred years, allowing aircraft to access a greater range of speeds and altitudes. Consequently, it became apparent that vehicle behaviour was dependent on the flight condition. In particular, the aerodynamic and propulsive differences required for subsonic, supersonic, hypersonic, and rarefied flight regimes, place varied demands on the flight control system. The following section considers various issues important to hypersonic flight control.

1.4 Hypersonic Flight Control

While many aviation pioneers pursued the development of the power plant, the success of the Wright brothers lay with their appreciation of flight control. The Wrights argued that a propulsion system is of little use without the capability of controlling the vehicle in flight. This is of course, generally true for applications of hypersonic air-breathing flight. However, what sets the hypersonic era apart is that practical scramjet engine operation is considered one of the greatest challenges in modern aeronautics. There are a host of recognized aerodynamic, material, and propulsion problems which continue to attract the majority of hypersonic research interest [118, 45, 106]. Similarly though, there are numerous issues central to controlled hypersonic flight.

Broadly speaking, the hypersonic air-breathing flight controller must provide stable, robust operation of a vehicle over a broad operating range, and ensure maximal engine performance subject to a highly constrained operating envelop. In other words, simultaneous control over the flight path, aerodynamic attitude, and propulsion is required. This key control requirement is a direct descendant of the integrated vehicle configuration and the conditions needed for efficient engine operation. Engine performance depends on the inlet's capability to capture the airflow, translating to a strong dependency on flight condition and the vehicle angle of attack. Such is the anticipated sensitivity of the propulsion system to variations in flight conditions, it has been suggested [45] that tracking tolerances of the order of 0.1° could be required for some configurations. Since local flow deflections control the engine operation, vehicle performance can also be critically affected by structural deformations [191, 41, 45].

Efficient engine operation also requires tracking an aggressive trajectory of high dynamic pressure, constrained by structural and thermal loading. The extreme operating environment places considerable demand on the instrumentation of sensors and the attachment of actuated stabilizing surfaces to the vehicle. A myriad of sensors are required throughout the vehicle, allowing for example, the detection of structural vibration modes, fuel flow rate, internal engine flow conditions, and the freestream. Air data measurements can be used to determine the pressure altitude and vehicle attitude. Due to the surface exposure to high temperature flow it is likely that all sensors will require cooling. Flow intrusive methods used for subsonic and supersonic aircraft [85] are not suitable for hypersonic vehicles as the sensors protruding from the surface would not survive the high energy environment of hypersonic flow. Also, booms instrumented with pitot tubes are sensitive to vibration and alignment error, and can induce flight instabilities which may degrade aircraft handling [231]. An alternative is the real-time flush air-data sensing (FADS) system [232, 117, 52], which couples a collection of pressure tappings located flush with the surface and an airdata estimation algorithm. Such a system has been flown on the shuttle orbiter [174, 13]. However, since the geometry of the scramjet vehicle is not conducive to such arrangements, further development in sensor design is required. A FADS system for the sharp-nose geometry needed by hypersonic vehicles has been developed for flight at Mach 3 to 8 [54].

Two of the critical parameters needed for the guidance and control systems are the dynamic pressure and angle of attack. Though FADS systems are being developed for this purpose it is likely that air-data sensors will be augmented by estimates available from inertial measurement units (IMU). The gyroscopes and accelerometers of an IMU are used to compute estimates for the vehicle velocity, altitude, and attitude, with respect to a fixed coordinate system.

For most launch vehicle applications a high degree of maneuverability during the acceleration stage, is not a desirable feature. A stable vehicle can be configured through the appropriate distribution of mass and the use of fixed wings, thereby simplifying the autopilot design. In early rocket launcher concepts, massive wings and fins were used to satisfy stability and control requirements. A fine example of this is the 81 m tall space ferry concept designed by von Braun [186]. The multi-staged launcher shown in Figure 1.7, incorporated three recoverable stages, each fitted with large stabilizing fins. The final stage was to function in a manner similar to the more familiar space shuttle orbiter. Like the space shuttle orbiter, the final stage of von Braun's design used aerodynamic surface to generate lift and augment stability for a gliding reentry. The complexities of practical operation do not follow linearly from existing propulsion systems. Scramjet powered vehicles have a fundamentally different operation boundary to rocket-only launchers like the von Braun concept and the space shuttle. Scramjets require prolonged access to high dynamic pressure conditions, implying the need to minimize the exposed area. Inherent stability through large aerodynamic surfaces would severely compromise the vehicle's application as an acceleration stage. While marginal stability characteristics may be desirable for maneuverability, in the scramjet vehicle there are stringent constraints on the vehicle attitude and the dynamic pressure variations.

Figure 1.7: A space ferry concept from the early 1950's, designed by Wernher von Braun. Standing 81 m tall with a launch mass of 6350 tonnes, it had three recoverable stages. (Source: Reference [186])

The selection and performance of control stabilizers is another key issue for hypersonic flight control. Aerodynamic surfaces and thrust vectoring are the primary choices available from modern aircraft and also find application in air-breathing vehicles. Their potential is very much dependent on the vehicle configuration. The Hyper-X configuration, shown in Figure 1.2 a), by virtue of lift generated by the inlet and nozzle surfaces, can make do with relatively small wings. Between the wings and elevators, pitching stability can be achieved at an angle of attack without unreasonably large aerodynamic losses. The axisymmetric scramjet on the other hand requires a relatively large wing area to generate lift and also large elevators to counter disturbances to the optimal zero angle of attack flight condition. The necessary control authority is therefore more difficult to achieve and may require a combination of control mechanisms, like aerodynamic surfaces and differential throttling. Whatever the vehicle configuration, high bandwidth, high strength control actuators are required. Due to the long forebody section and the generation of large thrust forces, the controller must perform a delicate balancing act.

Thrust vectoring has been used with increased frequency in recent years. It has found application in commercial airliners for safe landing after aero-surface failure and military aircraft to enhance manoeuvrability. On modern rockets, thrust vectoring through gimballing of the nozzle has significantly reduced and in many cases eliminated, the need for aerodynamic surfaces. For scramjet vehicles, thrust vectoring can be achieved by manipulation of the thrust surface [122, 213], external burning [29], surface blowing [243, 146] and differentially throttling the engine modules [234]. Considering the potential disturbance to the engine and airframe flow - especially with components being optimally configured - these methods will require accurate simulation to ensure critical vehicle operation is not lost. Also, the necessary fuel for thrust vectoring is in competition with the primary usage of fuel for thrust generation and cooling. It is therefore likely that thrust vectoring on scramjet vehicles will be used to augment vehicle stability and control, rather than act as the primary controller.

Without unconstrained access to hypersonic flight testing, the methods of assessing control feasibility are in the form of numerical and experimental simulations. The most relevant and accessible experimental capability available for hypersonic vehicles are impulse test facilities. Though these facilities are able to reproduce the hypersonic flight condition, they are unable to make dynamic measurements, can have uncertain and contaminated test flows, require scaling of models or partial vehicle simulation, and are still developing their capability, especially in relation to data measurement techniques. The alternative to physical experiments is to perform numerical simulations of the flow interacting with the vehicle. The most advanced numerical tool is computational fluid dynamics (CFD) [223]. CFD avoids many of the constraints associated with experimental facilities, but introduces additional unique constraints. Primarily, CFD applications are constrained by the time needed for accurate simulation of the flow structures, while the accuracy is dependent on the mathematical models used to describe the flow structures and the integration procedure. The continued development of both numerical and physical experimental techniques is crucial to the development of hypersonic flight technologies.

The status of scramjet vehicle development means that flight control design utilizes system models based on the aerodynamics, propulsion and control performance of *concept* vehicles. Accordingly, the system models are generally simplified numerical representations of the interactions between the vehicle and the flow path, or performance coefficients abstracted from experimental data. Due to the limitations of the these approaches and those of more advanced simulation techniques, control design must consider the effects of uncertain and unmodelled system features. Hence, there is a focus on robust control techniques by those researchers investigating the design of hypersonic flight controllers, see for example [188, 145, 157, 41, 38]. Robustness in this case refers to both stability robustness and performance robustness [209]. The former refers to maintenance of vehicle stability subject to parameter variation, while the latter is the assurance of proper response to commands and the reduction of response perturbations caused by disturbances. Contributions to hypersonic control are reviewed later in Chapter 2.

The primary causes of control failure leading to aircraft failure are insufficient control authority, actuator failure, and the effects attributable to uncertain or unmodelled system features. An early space shuttle orbiter reentry provides a cogent example of the effect of system uncertainty and the robustness required by flight controllers. Discrepancies in the predictions of pitching moment with those inferred from flight data, meant body flap deflections twice those predicted were required to maintain trim during the orbiter's first reentry [92]. This issue was finally resolved with a marriage of computational fluid dynamics (CFD) codes and experimental data [228], addressing the real-gas effects and viscous effects, and their influence on vehicle pitching moments and control effectiveness. Compared to air-breathing hypersonic vehicles, the shuttle configuration is relatively simple and CFD codes can be readily applied. The added complexity of scramjet operation means the accurate representation of flow features, to ensure understanding of the vehicle performances, continues to be an area of research.

Additional vehicle operating complexity arises from the highly nonlinear nature of its operation. Nonlinearities appear in the vehicle dynamics and the aerodynamic characteristics through dependencies with angle of attack, flight condition, fuel setting, and elevator position. Following a conventional approach to control design, the vehicle behaviour would be represented as a linearized model and analytical tools used to design the control system. If necessary the system parameters and control parameters would be "scheduled" [180] along the flight trajectory, with flight speed and altitude being probable reference variables. Working against this approach in the hypersonic regime is the broad operation range, rapid variations in aerodynamic response with changing flight condition, and the need for numerous indices such as Mach number, angle of attack, altitude, and dynamic pressure, to schedule the system behaviour. System nonlinearities also place a greater importance on the modelling of the system behaviour for control design. Fewer simplifications are appropriate for the dynamics and propulsive models if a reasonable representation of system performance is expected.

The combination of sensor requirements, actuator design, system uncertainty, and the extreme operating environment, has the potential to make or break the practical application of the scramjet engine to hypersonic flight. It is a complex, multi-dimensional task where isolation of components for analysis is generally not possible, and an integrated approach to trajectory, control, airframe, and propulsion is necessary. The aviator philosophy so deservedly revered by Lilienthal and the Wright brothers has been transformed by the restriction of testing to experimental facilities and computerized numerical simulations. Though full-scale testing is possible with numerical codes, real flight tests remain desirable for the gradual buildup to full-scale vehicle operation. The prohibitive costs of tests with full-scale models mean scaled down versions such as America's Hyper-X will require thorough exploration. The combination of extensive sub-scale simulation experiments with CFD is necessary for avoiding the scaling problems which thwarted Langley's efforts towards powered flight.

Considering the broad range of issues influencing control design and performance, the results of this thesis have been based on the more restricted problem of longitudinal control of a scramjet powered, hypersonic concept vehicle. We are not in a position to examine the complete operational characteristics. Nor is the purpose of this thesis to demonstrate through simulation, the successful six-dimensional operation of a real scramjet launcher concept. Rather, the purpose is to investigate the application of some "intelligence based" techniques to the design of a hypersonic flight controller. The field of artificial intelligence has developed ways of dealing with a very wide range of system uncertainties, non-linear systems, and many degrees of freedom, in the same manner that human intelligence has shown this ability.

1.5 Thesis Outline

The primary motivation for this thesis was the opportunity to address the flight control challenge presented by air-breathing hypersonic flight, through intelligence modelling and novel optimization techniques. Towards this, the aims of this thesis were:

- To develop a numerical system model to describe the aerodynamic, propulsive, and physical properties of a scramjet powered launch vehicle concept, operating in the hypersonic flow regime.
- To develop a simulation program for the dynamic simulation of controlled hypersonic flight about a spherical, rotating Earth. This is the basis of performance analysis used for the control design, and contrasts with the conventional use of linear models or look-up tables.
- Assemble a real-coded genetic algorithm for the optimization of arbitrary functions.
- To investigate the use of genetic algorithms in designing a robust fuzzy logic controller using a benchmark problem as a test case. This benchmark control problem is presented in a separate technical report [17].
- To demonstrate the evolution of a robust flight control function for a scramjet powered vehicle using full-nonlinear flight simulations as a performance measure.

- To evaluate the performance of the longitudinal autopilot through a full hypersonic trajectory flight simulation.

The organization of the thesis is described by the following list:

Chapter 2. Provides a review of developments in flight control, particularly those being applied to hypersonic air-breathing flight. A formal introduction to the autopilot design used for this thesis is also presented.

Chapter 3. Discusses the flight simulation package used for the design and analysis of the flight control laws. Attention is paid to the geometric specification of the scramjet powered vehicle concept, aerodynamic and propulsive modeling, mass properties specification, and the provision of a flight response history through the integration of the flight dynamics model.

Chapter 4. In this chapter the control system design tools are discussed. A fuzzy logic controller has been used in this thesis to describe the control function for inner loop of the autopilot. Fuzzy logic control represents a human based reasoning implementation of a rule based system. An introduction to the application and design possibilities is provided. The second major component of this chapter is the control design tool. Here a genetic algorithm is introduced as a powerful optimization tool for high order, highly nonlinear, and noisy functions. In the control design case, the objective function used to direct the search is sourced from controlled flight simulations.

Chapter 5. Implementation issues for designing a robust flight controller for the hypersonic scramjet are addressed. Control design is essentially an evolution of the control function using full non-linear flight simulations to build a fitness function. A number of techniques are used to encourage the rapid development of a stable robust control function. These include a large sampling of initial conditions to test the controller performance, and a non-uniform objective function.

Chapter 6. Reports on the results of simulated flight control experiments for the hypersonic scramjet. The results are organized with the aim of addressing a series of questions relating to the parameterization of the control function, the design procedure, and the performance of the genetic algorithm. Performance and stability robustness are the key requirements of the inner-loop attitude controller, and these are examined for a constant-gain linear controller and a range of fuzzy control laws. This chapter also presents a longitudinal guidance configuration for the purpose of providing a full trajectory simulation.

Chapter 7. A summary of this thesis is presented in the final chapter. Conclusions are made relating to the major components of this thesis: vehicle design and operation, the inner loop fuzzy controller, overall autopilot design, the genetic algorithm, and the evolutionary design procedure. Some proposals and recommendations are also drawn from these main areas.

CHAPTER 2

Approaches to Flight Control

Developments in flight control are driven by the continual development of flight technologies and the potential for disaster following control failure. This combination has placed flight control at the forefront of many advances in control theory. For example, the first type of adaptive controllers were designed for applications to aircraft control problems [93, 187, 127]. Advances in overall system performance have increased maximum flight speeds and operation range but, simultaneously have presented vehicles with marginal stability and exposure to more extreme and more variable environments. Throughout this period of improvement in propulsion system and aircraft hardware, the contributing factors to control failure have remained similar: unmodelled effects, inadequate control authority, and actuator failure. The inevitable increased demands placed on pilots have led to the evolution of flight control from a pilot dependent era to computerized automatic control of both piloted and unpiloted vehicles.

In June of 1903, a few months prior to the maiden flight of the Wright flyer, aeronautical theoretician G. H. Bryan made the following prediction:

> "The problem of artificial flight is hardly likely to be solved until the conditions of longitudinal stability of an aeroplane system have been reduced to a matter of pure mathematical calculation." Quoted in [218], page 2.

The Wrights had clearly developed an understanding of the principles of flight and the dynamic characteristics of their vehicle. By 1911, Bryan had established the mathematical theory for the motion of aircraft in flight, which are essentially the rigid body, six degrees of freedom equations used today [36]. The linearization of the equations of motion to form perturbation equations, thus simplifying the simulation of vehicle responses, was also a product of Bryan.

The need for a strong analytical approach to aircraft stability and control came much later, with the extension of the flight capabilities to point where piloting became difficult. Classical control theory developed from this need, employing frequency domain methods such as pole placement, root locus, and frequency response. Applying conventional methods relies on interpreting the system's dynamic response through descriptions such as settling time, oscillation frequency, rise time, overshoot and so on. The approach is generally most useful when dealing with single-input single-output (SISO) systems and linear time-invariant (LTI) systems.

The development of what is commonly referred to as modern control theory, was based on design with a state variable model and the use of mathematically precise performance functions to provide a solution for the control gains [211]. Modern control theory is applicable to multiple-input multiple-output (MIMO) systems, which may be linear or nonlinear, time-invariant or time varying. Compared to classical control theory which requires successive loop closure to select control gains in multivariable systems, modern control theory allows the simultaneous determination of all control gains. In general, the control element is introduced in a linear manner, with a quadratic performance index used to provide an algebraic equation for optimal gain design.

Though many aircraft may behave in a locally linear manner, the application of linear control theory over a broad operational envelope requires some means of adapting the control gains to maintain performance over the range of flight conditions to be encountered. Traditional flight control designs thus involve linearizing the vehicle dynamics about several operating conditions throughout the flight envelope, designing linear controllers for each, and using an interpolation scheme to blend the design points. “Gain scheduling” typically follows some predetermined schedule for the variation in the control gains, and is often expressed in terms of flight speed, angle of attack, altitude, or dynamic pressure. Gain scheduling is generally sufficient to ensure acceptable dynamic performance, though there are some situations where it cannot easily provide acceptable performance. These include rapid climbing and diving with large variations in dynamic pressure, rapid manoeuvres involving large angles of attack, and booster separation which involves considerable mass change.

In cases where gain scheduling is not feasible a self-adaptive control system can be implemented. The two types of adaptive systems are model reference (direct adaptive) and parameter-adaptive. The basic idea of an adaptive control system is the maintenance a constant or invariant closed loop dynamic response throughout the vehicle flight range. There was a large effort toward adaptive control research during the 1950’s and 60’s, including those by General Electric, Honeywell and MIT [14]. The early attraction of adaptive control of aircraft was the promise of a universal autopilot. The Honeywell adaptive control system was flight tested on the X-15 hypersonic research vehicle [202], though it was considered partially responsible for the failure of a vehicle flight test which tragically resulted in the loss of the pilot [202, 187].

As vehicle design advances, the level of certainty associated with representation of system model for the purpose of control design, is of increasing importance. Parametric uncertainties describe unknown parameters in an otherwise known model structure, such as that arising from linearized equations of motion. Example uncertainties include performance dependency on variations in the flight environment (dynamic pressure, Mach number, angle of attack), and large configuration variations relating to location of center of gravity, fuel and payload, and geometric variations. Another form of uncertainty consists of unknown and unmodelled dynamic processes at high frequencies. These include structural resonances and unsteady distributed aerodynamics. In the application of classical control theory, gain and phase margins are used to satisfy robustness of SISO systems. Multivariable techniques such as the linear quadratic regulator (LQR), provide optimal control strategies with guaranteed multivariable robustness properties. Recent developments in modern control theory through the use of H_∞ and μ synthesis, have developed methods of including uncertainty in the mathematical representation of the system, thus forming “robust control” theory. The aim of robust control is the development of control algorithms which guarantee a certain level of performance in the presence of uncertainties and disturbances.

The challenges posed by hypersonic air-breathing flight represent a development and application opportunity for advanced control techniques. Not surprisingly then, there have been many approaches presented as solutions to the flight control problem: sliding mode control, predictive control, quantitative feedback theory, nonlinear control, robust control theory, stochastic optimal control, and intelligence based control for example. Contributions in some of these areas are covered in the review in the following section.

The work of this thesis centres on the application of an evolutionary design approach, whereby the control parameters are evolved according to the performance of simulated, nonlinear, controlled flight responses. With a fuzzy controller performing the active stabilization of the vehicle, the control design procedure is effectively a search for knowledge process. Performance and stability robustness are developed by exposing the design to the full range of hypersonic flight conditions, and by representing parametric uncertainty and disturbance through randomized variations.

The remainder of this chapter provides a review of international efforts towards hypersonic vehicle control, followed by an introduction to a specific autopilot configuration and its evolutionary design.

2.1 Developments in Hypersonic Flight Control

Despite the relatively unproven concept of hypersonic air-breathing flight, there have been many contributions to the hypersonic flight control problem. The interest is driven by the challenges posed by hypersonic air-breathing flight, covered in Chapter 1, and the desire for control theory not to be trailing the development of flight hardware. The high performance of the first jet aircraft stepped ahead of stability and control technology [2], as did the first supersonic flight, where the difference between success and failure was getting the elevators to work.

In terms of hypersonic flight control design, the challenges generated are twofold. The first relates to the flight constraints of a highly nonlinear time-varying vehicle performance and the second is due to the degree of uncertainty in the performance of airframe, propulsive and control components. The common theme amongst developments in control theory is therefore the optimal design of a robust controller. Another recognized feature is the integration of guidance and control [192, 191, 96], due to the coupling of airframe and propulsion systems and the the sensitivity of both to the flight conditions and vehicle attitude. The following reviews some of the contributions to hypersonic air-breathing flight from the past decade, under the headings of intelligent control, adaptive control, stochastic robustness and parameter tuning, multivariable robust control, optimal control, fuzzy control and hypersonic maneuvering.

2.1.1 Robust Intelligent Control

Chamitoff's thesis [39] applied "intelligent" optimization methods to the development of a robust predictive flight control strategy. The flight control was formulated as receding horizon optimal control problem, which provides stable tracking of a desired trajectory. Lyapunov stability was combined with an enhanced A* optimization algorithm, to search through possible short term trajectories. With the inclusion of parametric uncertainty, a robust control solution can be obtained by minimizing the worst case tracking error. The performance of candidate solutions was assessed by simulating the full nonlinear dynamics, which incorporate vehicle constraints and parametric uncertainty. Due to the emphasis on the development and analysis of the trajectory control system, an appropriate inner-loop feedback is assumed for the rejection of high frequency disturbances while tracking the outer-loop commands. The results clearly show the benefits of a multi-step trajectory prediction compared to a single step optimal controller. A feature of the work was the development of a simulation environment for representing the dynamic behaviour of the vehicle.

2.1.2 Adaptive Control

The two forms of adaptive control referred to here are the scheduling of control gains with respect to the flight condition and the vehicle attitude [52], and the adaption of controller parameters using model reference adaptive control [158]. With the proven success of gain scheduling in flight control, it is not that surprising that, for the first flight testing of a flight style hypersonic air-breathing vehicle [79, 177, 52], a gain scheduled controller has been adopted. The flight control laws for NASA's Hyper-X Research Vehicle are required for stage separation, maintenance of the design condition during the engine tests, and a controlled descent. Conventional longitudinal and lateral control loops are used with guidance commands and sensor feedbacks combining to generate aerodynamic surface commands. For the longitudinal control law, angle of attack and pitch rate are used to derive a symmetric command for the all-moving wing. The control laws were designed using classical linear control design techniques with feedback gains scheduled with angle of attack, Mach number, and dynamic pressure. Robustness analysis included full nonlinear simulations of numerous parametric arrangements using a Monte Carlo analysis. Stability analysis concluded that gain and phase margins were within guidelines. With the spectacular failure of the booster elevators in a recent flight test [199], it remains to be seen whether the flight control developed for the hypersonic portion of the flight test is successful.

Another form of adaptive control is model reference adaptive control (MRAC), where the feedback element is based on matching the vehicle performance with that of a reference model. A paper published by Mooij [158], provides a numerical investigation of MRAC for a hypersonic aircraft. The vehicle model is the winged-cone configuration, representing a generic accelerator vehicle. [196]. The database of aerodynamic properties for the accelerator vehicle has been widely used in control studies [225, 145]. For the reference model, a linearized model of the rotational dynamics was used, following the assumption that the translational motion has no influence on the rotational model. It is interesting that the vehicle is described by nonlinear differential equations, yet the adaptive algorithm assumes a linear time-invariant system. There is also a stability requirement that the controlled nonlinear system is almost strictly passive. Though the fundamental design of the MRAC control system is relatively easy, there are many control design parameters, of which quite a number are configured in a trial and error process.

2.1.3 Stochastic Robustness and Parameter Tuning

Stochastic robustness [178] characterizes a compensator in terms of the probability that the closed loop system will have unacceptable behaviour when subjected to parametric uncertainties. The stability and performance metrics used to indicate the system behaviour can include classical and frequency domain metrics, such as closed loop eigenvalues for stability of the locally linearized system, settling time, and overshoot. Since the scalar robustness cost function is simply a weighted sum of individual behaviour probabilities, it is also possible to include bounds on the operating envelop and actuator constraints as performance metrics.

Marrison and Stengel [145] combined Monte Carlo evaluation and genetic algorithms to design robust, linear-quadratic control parameters, dependent on a stochastic cost function. Monte Carlo evaluation allows a finite set of samples over the expected system parameter space, thus providing a practical method for the estimation of the behaviour probabilities and cost functions. The subsequent computational penalty of the large number of Monte Carlo evaluations to design the controller can be reduced by an efficient search procedure such as that provided by genetic algorithms. Longitudinal flight dynamics were modelled using aerodynamic coefficients from the winged-cone configuration model of [196], with 28 uncertain parameters, each based on a normal probability density function. Aerodynamic coefficients and air data are interpolated from lookup up tables or spline fits, to data around the nominal cruising condition. Their stochastic robustness analysis was also used to design robust compensators for a benchmark control problem [144], and through stochastic robustness analysis, used to compare the robustness of compensators designed by different groups [210]. The approach appears to be an effective basis for flight control design and analysis yet, despite the large number of uncertainty parameters, the system features are limited by the complexity available with the winged-cone data. Another limitation of the analysis is that the uncertainty parameters appear to be applied as constants throughout each simulation trial, rather than carrying frequency components according to the physical nature of the uncertainty. As with any multi-objective optimization problems, care must be afforded to the specification of the weights used for each performance metric.

Continuing the work on flight control by Marrison and Stengel, is the contribution from Wang and Stengel [225]. Here control laws based on nonlinear dynamic inversion of the aircraft model are developed with stochastic robustness. Apart from the dynamic inversion aspect, the work follows directly that covered by Marrison and Stengel in Reference [145]. To characterize the system robustness, the probability of instability and the probabilities of violations of 39 performance metrics are used.

Research motivated by the development program for the Hope-X Japanese reentry space vehicle has applied the stochastic robust analysis methods established by Ray and Stengel [178], in the form of a stochastic parameter tuning method [157]. More than 100 uncertain properties were modelled, with the performance analysis based on landing performance requirements. The parameter optimization problem is characterized by a noisy performance index which can be computationally expensive to evaluate, and by the potential for many adjustable design parameters. A mean-tracking technique was adopted as the optimization algorithm, citing its reliability and efficiency. However with the mean-tracking technique being equivalent to gradient-based methods, the solution returned may not be the global optimum. The authors advocate the use of distributed computation for the Monte Carlo flight simulation and have developed parallel computing software for the implementation of the design over a TCP/IP network protocol. They also envisage the application of the stochastic parameter tuning method in combination with advanced robust control theory and design methods.

2.1.4 Multivariable Robust Control

Though all the hypersonic flight control developments have focused on robustness, this section refers to the application of advances in modern control theory in the direction of multivariable robust control techniques using H_∞ and μ-synthesis techniques. Because the literature on these techniques has become very large over the past few years, readers interested in the development and application of H_∞ and μ-synthesis are directed toward the review references [62, 61, 163, 220, 37]. The objective of H_∞ control is to find the compensator transfer function matrix such that H_∞ norm of the closed-loop transfer function is minimized. H_∞ theory does not take into account the possible structure in the uncertainty, and may therefore lead to conservative controllers unable to satisfy performance measures. μ-synthesis allows the introduction of uncertainty structure in the controller design process.

The vehicle configurations tested include both the hypersonic cruise aircraft [10] and the winged-cone accelerator configuration [91, 38]. The basis for the synthesis of the controllers is a linear time-invariant system model which consequently means the analysis point is generally an equilibrium condition (as opposed to one where the vehicle is accelerating) or has some steady state flight condition. Gregory *et al.* [91], using the winged-cone model with an updated propulsion model, represented the time varying parameters by a multiplicative uncertainty for the system model, allowing the linear system to be considered time invariant over some interval. For their initial application of robust control theory to the problem, the structured uncertainty representing parameter variation with time, was described for elevator effectiveness and fuel flow rate. Atmospheric turbulence and signal noise were also included in the design of the controller.

Buschek and Calise [38] have developed a fixed-order design dealing with mixed real/complex uncertainties. To reduce the conservatism in the design procedure, a μ synthesis method is used, providing an iterative procedure where the H_∞ design is a subproblem in the mixed μ-synthesis procedure. The hypersonic vehicle model was the winged-cone configuration of [196], with the linear model configured for a trimmed, accelerated flight condition. A limitation of the source data is that the propulsion system model does not include sensitivity to angle of attack variations. Following the same reasoning as Gregory *et al.*, the inclusion of signal noise is not simple for realistic sensor data, but the application of H_∞ theory to output feedback problems requires the corruption of all measurement signals by noise. Performance sensitivity to angle of attack is represented by parametric uncertainty in the pitching moment sensitivity to angle of attack variations. It was included in the system model through a scalar perturbation to the nominal model. There was also an attempt to represent the elastic modes of the vehicle, by introducing uncertainty in the rigid-body behaviour. In the results presented, the controllers demonstrated robustness to atmospheric turbulence and a worst-case disturbance model. However, due to the linear nature of the control synthesis, gain scheduling would be required for the different flight conditions. Another significant result was the illustration of the general superiority of the fixed-order design technique over the order reduction approach which had previously being used to reduce the order of H_∞ controllers. Since the fixed order controller synthesis requires a numerical iterative approach to defining the optimal design, the potential for numerical ill-conditioning for large systems led to the suggestion that alternative numerical approaches such as genetic algorithms and simulated annealing may be useful.

Chavez and Schmidt [41] have also reported on the application of multivariable control robustness, but the focus was on the modelling of uncertainty with underlying physics of the real system in mind. Three forms of uncertainty are considered: real parameter, unstructured, and structured uncertainty. Multivariable robustness analysis requires the representation of uncertainty as some combination of structured uncertainty (allowing specific sources to be identified and represented) and unstructured uncertainty (where elements are arbitrary, mutually independent, and complex). Representing the uncertainty in a feedback system allows augmentation of the system matrix. The application of the generalized Nyquist theory [137] to establish stability robustness produces a conservative inequality criteria, which can be mitigated if the actual uncertainty has some structure. Chavez and Schmidt used a generic hypersonic vehicle similar to the X-30, which was the predecessor to the Hyper-X project. The model included an elastic degree of freedom, the neglect of which was shown to be not justified. The aerodynamic, propulsive, and structural models where described analytically in the form of stability derivatives which are nonlinear functions of vehicle geometry and mass properties, atmospheric pressure, structural-vibration mode shapes, and flight number [40].

2.1.5 Optimal Control

In addition to an emphasis on robust control, all the control applications also fall under the umbrella of optimal control. The original format of optimal control in modern control theory was the derivation of the optimal feedback law using the linear quadratic regulator (LQR). McLean and Zaludin [149] applied LQR theory to design an optimal feedback control law robust to modelling uncertainties in the aircraft dynamics. A linear quadratic Gaussian regulator solution was considered unnecessary due to the assumption of negligible atmospheric turbulence. The complexity of an H_∞ controller was also considered unnecessary, due to the resulting high order controller which makes it difficult to assure the degree of controller reliability necessary to ensure stability. Analysis was based on a linear state equation for the longitudinal motion which, from inspection of the eigenvalues, was both statically and dynamically unstable. In addition to ensuring stability, the controller design would need to limit the change in angle of attack due to the coupling between the aerodynamics and propulsion systems. Some additional flying qualities were quantified in relation to the pitch, height, and speed responses. The main focus of the work presented was to address the difficulty associated with assigning the state and control weighting matrices. Their approach was to determine the state weighting matrix based on matching the closed-loop eigenvalues with a predefined set. The linear state equation describing the longitudinal motion of the hypothetical hypersonic transport aircraft was based on the mathematical model defined in [40], with the state vector representing small perturbations about an equilibrium condition and included variables for flexible mode. A single altitude response simulation was plotted, showing a very slow height response generated by very small angle of attack and pitch angles.

2.1.6 Fuzzy Control

Fuzzy control is seemingly well suited to the hypersonic control problem due its robustness to variations in the vehicle performance, and the capability of describing a nonlinear control law [131]. There have been many proposals for the application of fuzzy logic based guidance and control, including conventional proportional derivative control, adaptive control, sliding mode control, hierarchical systems, optimal control, and fuzzy gain scheduling [135].

There have been limited studies on the application of fuzzy control to hypersonic flight control. Christian [43] reported the application of a fuzzy logic controller for the regulation of the acceleration of a hypersonic interceptor. A linearized longitudinal dynamics model was used with the aerodynamic coefficients defined by nonlinear functions of angle of attack, providing an unstable airframe. The primary objective of the study was the design of a broad range fuzzy controller to express the thrust level as a function of acceleration error and pitch rate. It appears that the rules were heuristically determined. That the controller was so effective is probably a reflection of the simple system model used in the analysis. With the addition of an adaptive scheme based on changing the membership functions, the acceleration response showed considerable robustness to large changes in the aerodynamic parameters.

A Sänger-type hypersonic transporter was the study vehicle for an application of fuzzy logic based flight control by Zhou *et al.* [244]. The purpose of the controller was to provide longitudinal stability and attitude command tracking. The flight characteristics were defined through the longitudinal linearized equations of motion about a horizontal reference flight condition, with elevator deflection angle as the control variable. Four reference flight conditions were used, the two hypersonic conditions possessing short-period modes which were dynamically unstable. Angle of attack and pitch rate were used as inputs, and the rule base was developed according to the behaviour of a human pilot. Simulated angle of attack responses depicted a favourable comparison between the fuzzy controller and standard linear proportional-derivative feedback control system, and showed the robustness of the fuzzy controller to variations in the flight condition. The superiority of the fuzzy control law in this case is attributable to the non-linear control law which was generated by localized manipulation of the control surface.

2.1.7 Hypersonic Maneuvering

Maneuvering in hypersonic flight is significantly affected on by the operating constraints of the vehicle and high speed flight effects such as centrifugal relief, requiring a non-zero normal load factor required to maintain a constant altitude. This was one of the conclusions drawn by Raney and Lallman [176] while addressing a control concept for maneuvering in hypersonic flight. The overall control system consisted of outer loop controls to track guidance commands and reject disturbances, inner loop controls to provide stability augmentation and a resolver to communicate between the two loops by translating acceleration commands to a normal load factor and a bank angle. Lateral flight simulation experiments of the resolver concept were performed with the winged-cone configuration [196], trimmed at several Mach numbers for a constant dynamic pressure. Angle-of-attack commands were limited to $\pm 0.4°$ about the trim value, representing the performance degradation likely with small angle of attack perturbations. The assumption of perfect angle-of-attack control and negligible propulsive and aerodynamic variations with angle-of-attack could be expected to make real flight hypersonic maneuvering considerably more difficult.

2.2 A Longitudinal Autopilot for Hypersonic Flight

In Chapter 1, the demands of flight control for the hypersonic accelerator where discussed. The inherently unstable vehicle operates over a large flight envelope for which there is significant uncertainty over all facets of its performance and the environment in which it is traveling. It exhibits non-linear behaviour on account of performance variation with flight condition and altitude, and also due to non-linear flight dynamics. The principal aim of this thesis is the production of an autopilot for the air-breathing hypersonic stage of a small launch vehicle concept with an emphasis on inner-loop controller. The objectives of the autopilot are to provide robustness of stability and tracking performance over the entire hypersonic regime, subject to the presence of modelling uncertainty and disturbances;

and to satisfy operational constraints represented by limits on the vehicle attitude, flight envelope, and actuator limits.

The block diagram shown in Figure 2.1 represents a conventional longitudinal autopilot structure. The overall control system consists of two main subsections: (i) an outer loop guidance system which provides trajectory maintenance and (ii) an inner loop stability augmentation and attitude controller for the tracking of guidance commands and the rejection of disturbances. Each of the major blocks is introduced in the following sections.

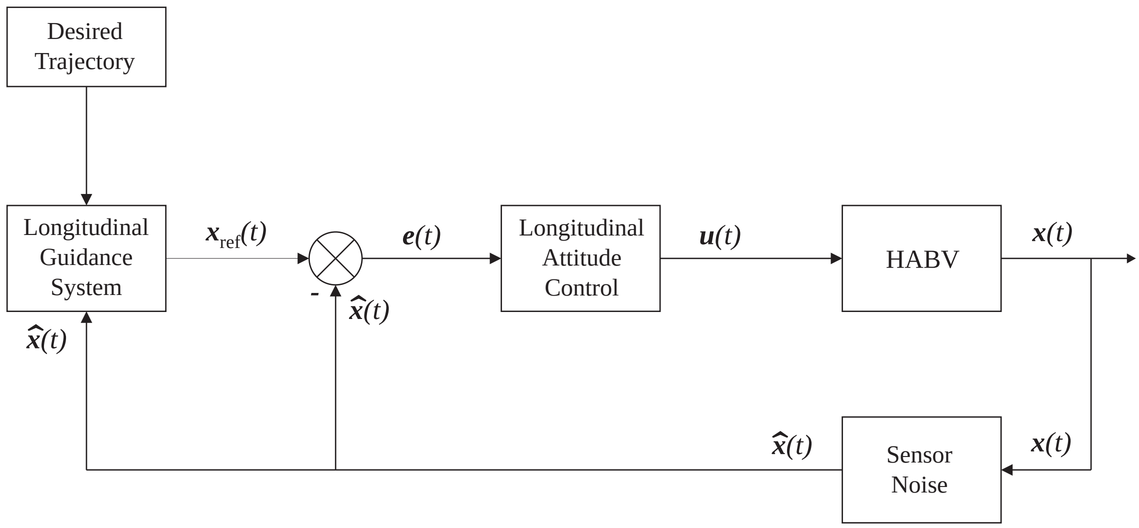


Figure 2.1: Longitudinal autopilot for the hypersonic air-breathing vehicle.

Hypersonic Air-Breathing Vehicle Model, (HABV):

The vehicle concept considered in this study is representative of an axisymmetric scramjet powered accelerator stage of a small scale launch vehicle configuration [194, 113]. Aerodynamic and propulsive modelling has been confined to the hypersonic operation of the vehicle. A full description of the hypersonic vehicle configuration and its flight simulation is provided in Chapter 3.

A simplification for the purpose of aerodynamic and propulsive analysis describes the flow paths as two-dimensional ducts. The six degree-of-freedom rigid-body motion of the vehicle is described about a spherical, rotating Earth, by the nonlinear set of differential equations describing the vehicle state dynamics:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{F}(t, \boldsymbol{x}, \boldsymbol{\Delta}, \boldsymbol{d}), \boldsymbol{M}(t, \boldsymbol{x}, \boldsymbol{\Delta}, \boldsymbol{d})) \tag{2.1}$$

The force $\boldsymbol{F}$ and moment $\boldsymbol{M}$ vectors are evaluated from surface pressure distributions along the exposed vehicle surfaces, arising from aerodynamic and propulsive effects. A numerical simulation of the vehicle's aerodynamic and propulsive behaviour describes the internal and external flow processing in terms of the atmospheric flight conditions, the vehicle geometry, the orientation of the vehicle, and the control actuator position. Parametric performance uncertainty, $\boldsymbol{\Delta}$, provides a stochastic time history of perturbations in the propulsive performance, mass properties, control effectiveness, and atmospheric conditions. Atmospheric turbulence provides a disturbance input $\boldsymbol{d} = [\Delta u, \Delta w]$, where Δu and Δw representative additive longitudinal and vertical turbulence velocities respectively. Though aeroelastic effects can have a significant influence on the vehicle performance, they have not been considered in this analysis.

The vehicle state $\boldsymbol{x}$, combines the vehicle position, attitude, angular velocities, translational velocities, mass, and control settings,

$$\boldsymbol{x} = \left[(R, \mu, \lambda), (\psi, \theta, \phi), (p, q, r), (u, v, w), m, (\theta_{\text{e}}, \theta_{\text{e,cmd}}, \nu_{\text{U}}, \nu_{\text{L}})\right]^{T}. \tag{2.2}$$

The inclusion of altitude in the state vector accounts for gradients in the atmospheric properties and gravity. Longitudinal flight control relies on the symmetric action of a rear wing-elevator combination, appearing in the state vector as an elevator angle θ_{e}. Fuel input settings $(\nu_{\text{U}}, \nu_{\text{L}})$, for the two scramjet engine modules, are included in the vehicle state description to account for the mass loss and subsequent variations in the vehicle mass properties. The subscripts U and L refer to the upper and lower engine modules respectively.

For control design, it is common practice to simplify the vehicle dynamics by linearizing the differential equations. This is generally done by retaining only the first-order terms from the Taylor series expansion of the non-linear equations of motion, relative to a non-equilibrium or steady-state flight condition. If the controller design is based on linear time invariant analysis, special care must be taken with the time varying aspects of the vehicle operation. In robust control analysis, parameter variations with time can be accounted for by augmenting the system model using a multiplicative uncertainty [91], such that the linear system can be considered time invariant over some interval and LTI control design can be used. In the present study, control design is not reliant on such a conventional representation of the vehicle. Instead, the full non-linear model is used to configure the controller such that simulated flight responses guide the development of an optimal control law.

Desired Trajectory:

Operation of the hypersonic scramjet depends on maintaining the vehicle within a narrow flight envelope. Trajectory planning and maintenance is thus a critical feature of the flight plan and autopilot role. The desired flight path may be sourced from an *a priori* solution to a global optimization problem to provide, for example, a near-minimum fuel trajectory, or it may be computed on-line during flight as a means to provide simultaneous maintenance of optimal engine performance and vehicle attitude. A feature of many hypersonic control studies is the development of an integrated trajectory management and control strategy [192, 191, 96].

Conventional launch trajectories are often expressed through a time history of velocity and altitude, so that stage separation can be scheduled. In this form, the role of longitudinal guidance is the simultaneous tracking of altitude by directing the flight path angle and the tracking of flight velocity via throttle control. Time based trajectory management for the hypersonic air-breathing vehicle is inappropriate. Due to the likely marginal thrust capabilities of the vehicle, its operation as an acceleration stage relies on the maintenance of optimal inlet conditions and maximal fuel settings. Since dynamic pressure constraints bound the hypersonic flight regime, a constant dynamic pressure trajectory is generally considered desirable. There are alternative specifications, such as a constant climb rate, constant ρV which roughly corresponds to constant thrust, constant ρV^2 corresponding to constant aerodynamic pressure, and $\rho^{1/2} V^{2\,1/2}$ corresponding to constant aerodynamic heating rate [194]. In this thesis, a constant dynamic pressure ($q_\infty = \frac{1}{2}\rho_\infty V_\infty^2$) trajectory is used to provide the nominal flight path as a mapping of velocity versus altitude, $h_{\text{ref}} = f(q_\infty, V_\infty)$. The role of the autopilot is therefore to be at the correct altitude for the flight velocity. Since the vehicle configuration used in this study does not support differential throttling for stability augmentation, the role of thrust modulation as a control variable is effectively removed. The constant dynamic pressure trajectory is plotted in Figure 2.2 for $q_\infty = 188\,\text{kPa}$. The flight profile describes an acceleration from 2500 m/s at 22.4 km altitude to 4900 m/s at an altitude of 31 km. It describes a gradual climb which allows sustained scramjet operation in the narrow air-breathing corridor. The “desired trajectory” block in Figure 2.1 takes the flight velocity and iteratively solves for the altitude, with reference to a standard atmosphere model. More detail on the atmosphere model is provided at the end of Chapter 3.

An estimation of the instantaneous trimmed condition is also provided as part of the trajectory data, though not strictly written in terms of the flight path. The trimmed condition for the elevator is expressed as a function of angle of attack α, using a least squares fit to the trim variation across the trajectory. Trimmed flight has been described by the elevator position which provides a zero net pitching moment on the vehicle. This state is not a steady-state flight condition, as an imbalanced lift force will rapidly disturb the trim condition. Using the vehicle configuration defined in Chapter 3, the trim equation is written as the following polynomial approximation, with the angles defined in radians.

$$\theta_{\text{e,trim}}(\alpha) = -117.4\alpha^3 + 0.1091\alpha^2 + 2.078\alpha - 0.0033 \tag{2.3}$$

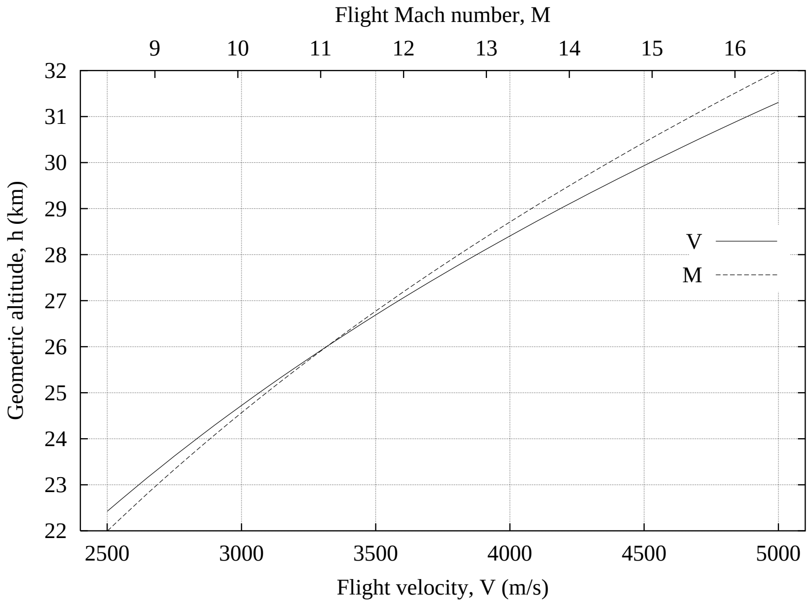


Figure 2.2: Hypersonic flight trajectory for a constant dynamic pressure $q_\infty = 188\,\text{kPa}$. The Mach number profile assumes a standard atmosphere, described in Section 3.6.

Figure 2.3 shows a sampling of the source data for the above expression. Each dashed line represents the variation of the trim elevator position versus angle of attack, for a specific flight condition. The thick solid line represents Equation 2.3. A consequence of the estimated trim condition being used by the control system, is a requirement of the controller to be robust against trim uncertainty. Vehicle performance variation will also contribute to errors in the trim estimate.

Longitudinal Guidance System(Outer loop):

Tracking the desired trajectory is the role of the longitudinal guidance system. The guidance law generates an angle of attack command, α_{cmd}, for the longitudinal attitude controller. It is based on the altitude error h_{err} and climb rate error $\dot{h}_{\text{err}}$, such that

$$\alpha_{\text{cmd}} = F_{\text{G}}(h_{\text{err}}, \dot{h}_{\text{err}}). \qquad (2.4)$$

Rather than having a preset nominal climb rate, the reference condition is estimated according to the current acceleration performance of the vehicle, with the assumption that the nominal trajectory is being followed. For simplicity, a constant gain guidance law is used with a bandwidth that provides the inner loop with sufficient time to trim the vehicle. The proportional and derivative guidance gains are configured to optimize the trajectory tracking performance. To be compatible with the capabilities of the inner loop, the guidance command and equivalent attitude error are bound by $\pm 3°$.

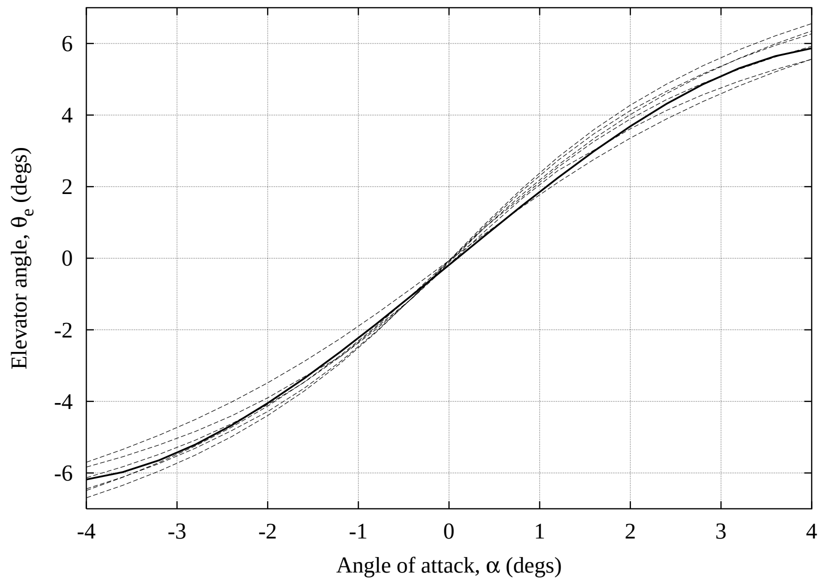


Figure 2.3: Variation in the elevator trim condition across the hypersonic flight envelope. Each dashed line represents a flight condition selected from the constant dynamic pressure flight trajectory. The solid line describes the approximation used by the autopilot.

Longitudinal Attitude Control (Inner loop):

Closed loop stability and attitude control is provided by a nonlinear feedback control law. It expresses a functional relationship between a subset of the vehicle state relative to the guidance commands, and the actuation command for the symmetric elevators. Due to the nozzle configuration used in this study, differential throttling of the engines for the purpose of stability augmentation or attitude control, is not a practical option. Results in Chapter 6 use both a linear constant gain feedback controller and a fuzzy rule base controller to express the elevator control law. Each controller can represented by the functional relationship,

$$u_{\mathrm{e}} = F_{\mathrm{att}}(\alpha_{\mathrm{err}}, \dot{\theta}, \theta_{\mathrm{e,err}}), \tag{2.5}$$

where the inputs are angle of attack error (α_{err}), pitch rate ($q = \dot{\theta}$), and the elevator trim error ($\theta_{\mathrm{e,err}}$). It is assumed that accurate full state information is available. In the Hyper-X testing program [52], an inertial measurement unit is used to supply accurate angle of attack data at a high bandwidth. Flush air data systems are also being considered, relying on accurate flow simulations to provide a functional relationship between the pressure differential amongst surface pressure ports and angle of attack.

The control command, u_{e}, describes the actuation rate for the symmetric rear wing-elevator combination, which is the only active stabilization and control device. Though there is no strict constraint applied to the steady-state attitude error during the design, a component of the objective function is the minimization of the settled response error. It is expected that the longitudinal attitude controller be able to maintain an angle of attack error less than 0.5 degrees. With the assumption that inertial measurements are available for specifying the vehicle angle of attack, perturbations from atmospheric turbulence are not included in the control input signal. The turbulence is introduced into the simulation as a freestream disturbance rather than additive noise for the control input.

2.3 Evolutionary Design for Robust Flight Control

As mentioned earlier, the basis of the control design approach is to use simulated flight responses to guide a parameter optimization procedure. The basic structure of the controllers is predetermined, see Chapters 4 and 5, and the free parameters are then optimized by a genetic algorithm, so that the simulated flight responses for a variety of initial conditions display desirable properties, such as long term stability, fast settling, disturbance rejection and broad range performance. The genetic algorithm is a zero-order search procedure, where the only information used to direct the search process is a performance measure, referred to as the objective function, computed from a set of simulations. Though the design procedure is essentially a brute force approach, it has been configured, in terms of the controller structure, the search algorithm, and the adaptive performance measure, to moderate the computation time required.

There are a number of advantages to designing the controller with an optimization tool and a performance metric abstracted from the randomly perturbed flight responses. Firstly, it relieves a common issue faced by many control design approaches, namely representing the vehicle mathematically in an appropriate form. The accuracy of the model is a function of available computing power and the knowledge of the vehicle physical properties and the processes governing the performance, rather than being bound by the structure of the control design procedure. In conventional design theories the system is typically assumed to be LTI and, in the case of robust control theory, uncertainty added to the system to account for system nonlinearities and variations with time. Representation of performance uncertainty is critical for the development of a robust control law. Much work in robust control theory is directed towards the development of compatible structured and unstructured uncertainty models. When the simulated flight responses are used, the inclusion of parametric uncertainty can describe the physical process leading to the variations in the vehicle performance, through the inclusion of appropriate simulation models.

Another advantage of the design approach is that the control law development is linked directly to the time history responses, allowing stability and performance measures to be easily quantified. The genetic algorithm does not need the components of the objective function to be the same throughout the design. They too can evolve with the controller design so that as the controlled flight responses improve, greater demands can be placed on the performance of the controller. The only constraint on the objective function with the design procedure implemented in this thesis, is that it be reduced to a scalar performance measure. Further supporting the evolutionary design procedure is the inherent robustness to noisy objective functions. The source of the noise is typically due to randomly sourced uncertainty models and sensor noise, but may also be generated by variation in the initial condition set used to assess the controller performance. Since the search is probabilistic, a significant level of noise in the objective function can be tolerated, while still allowing the genetic algorithm to successfully configure the control law.

Though the genetic algorithm is noted for its global search capabilities, it is also extremely opportunistic. Considerable care is therefore needed when defining objective functions, and when combining multiple and possibly conflicting design objectives. However, this is a feature which must be addressed in all optimal control theories. In problems where noncommensurate objectives are unavoidable, evolutionary algorithms are considered to be particularly suited since a set of solutions are processed in parallel. One means of dealing with such problems is to use a multi-objective genetic algorithm [77, 245] to obtain Pareto-optimal solutions.

One potential problem in an iterative design approach is the “curse of dimensionality”. As the number of design parameters increases there may be an exponential increase in the effort required to arrive at the solution. Though this can be mitigated by providing some structure to the design, it is important that a large number of design parameters can be dealt with. Evolutionary based search procedures are readily applied to problems of high dimension, and are able to rapidly extract useful designs in spite of the size of the problem. If the absolute global minimum or maximum of a complex multi-modal search space is required, then the computing effort remains considerable. However there are few algorithms capable of performing well on such functions and the notion of an efficient search procedure is still being established.

The focus of this effort is the design of an inner-loop attitude controller which would offer closed-loop vehicle stability, subject to system uncertainties, broad range performance variations, disturbances, sensor noise, and severe operational constraints. In the chapters that follow, a detailed description of the major areas of the research is provided. These include the hypersonic aerodynamics and propulsion modelling, flight simulation, control system configuration and design, and the construction of an evolutionary design tool.

CHAPTER 3

Simulation of Hypersonic Flight

A fundamental step in the development of an aircraft is a means of simulating the dynamic behaviour of the vehicle. The aviation pioneers of the Wright Brothers era set the standard by flying model aircraft and constructing simple wind tunnels. With well understood propulsion units, their principal intent was the thorough testing of vehicle aerodynamics and stability and control features. For the development of hypersonic technologies, the test flights of the X-15 aircraft allowed, amongst other things, the investigation of materials, flight systems and vehicle control requirements. The X-15 marked the first application of hypersonic theory and wind tunnel work to an actual hypersonic vehicle and, as a simulation tool itself, was extremely productive. It facilitated the transition to space flight, leading to the space shuttle orbiter.

As flight technologies have evolved, the simulation tools have become more specialized and less accessible. Today, flight access to hypersonic speeds is only available via rocket propulsion. The most familiar vehicle that regularly accesses the hypersonic regime is the space shuttle orbiter. Compared to the hypersonic air-breather there is a significant gap in the operating characteristics. The shuttle performance differences are due to its flight envelope, method of operation, and the vehicle configuration. Shown in Figure 3.1, is the ascent and reentry trajectories of the shuttle along with the ascent trajectory for a scramjet powered vehicle. In the ascent phase of the shuttle, high Mach numbers are reached only at high altitudes, and the severe thermal and pressure loading of an air-breathing vehicle are not duplicated. Reentry of the orbiter is as an unpowered glider and like the Apollo reentry, the descent involves deceleration at much higher altitudes than those needed for scramjet operation. The significance of this is not minor. By example, engineering estimates of aerodynamic heating show the expected stagnation point heating of a hypersonic air-breathing concept to be an order of magnitude greater than that for reentry heating of the shuttle [217, 6].

A further constraint to generating real flight test data is the cost of model production and the supporting launch vehicle needed to reach test conditions. The Hyper-X test program is a recent example of the demands of flight testing a hypersonic air-breathing vehicle. It uses a Pegasus launcher, released from carrier aircraft, to deliver the vehicle to

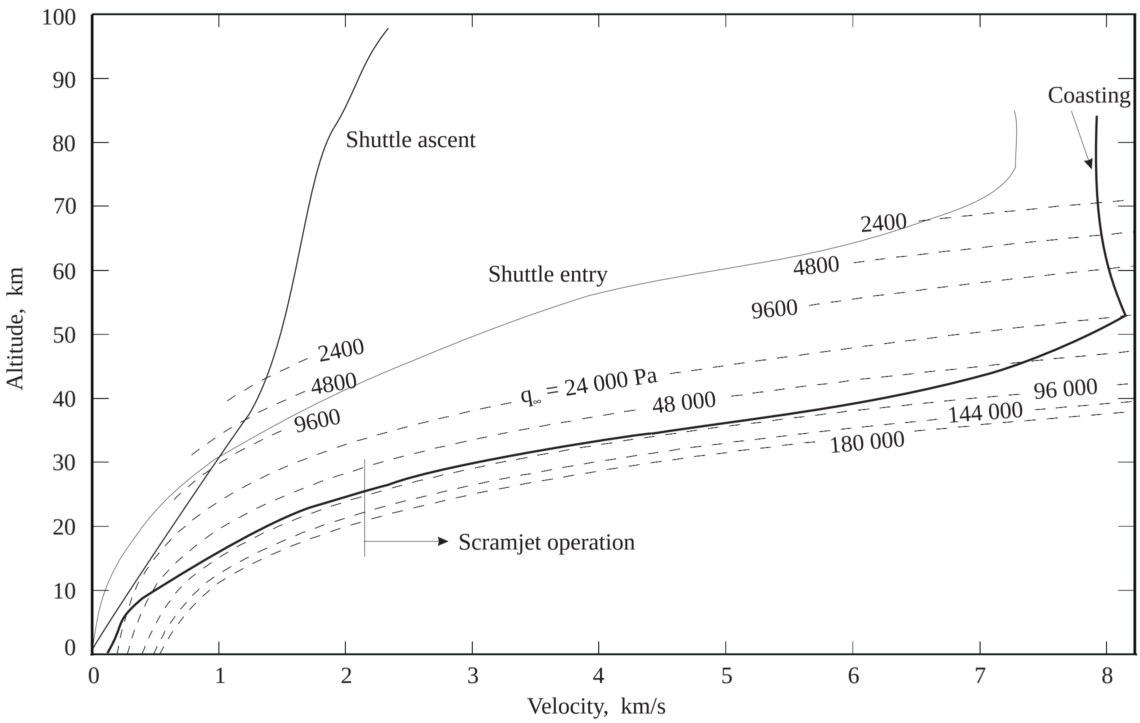


Figure 3.1: Comparison of a typical ascent trajectory for a scramjet powered flight vehicle and the ascent and reentry operation of the shuttle. The dashed lines are contours of constant freestream dynamic pressure $q = \frac{1}{2}\rho_\infty V_\infty^2$, using the atmospheric model from U.S. Standard Atmosphere, 1976 [1], and described in Section 3.6. (Source: Adapted from reference [29])

the hypersonic conditions needed for scramjet operation. The test vehicle is not a simple concept model, rather an advanced subscale (m) vehicle design, incorporating decades of vehicle and engine research in the configuration. The advanced configuration reflects the need for optimal airframe and engine coupling, to realize practical scramjet propulsion. Hyper-X has the support of an extensive ground based simulation program [151, 222, 66] which has allowed stability and control aspects to be well established. While a flight test program such as Hyper-X is not realizable for the majority of research groups, there is potential in less ambitious flight test projects which test simplified vehicle concepts or simple engine units attached to rocket launchers. We have already seen a number of programs of this nature: the X-15 rocket plane with an axisymmetric scramjet [45], Russian axisymmetric tests [120, 100], and the HyShot test program [170].

An implication of the divide between hypersonic rocketry and air-breathing hypersonics is a shift in the approach to vehicle design and analysis. Vehicles such as the shuttle orbiter and reentry probes have a relatively simple geometry exposed to the flow, in contrast to the complex internal and external flow structures of a scramjet powered vehicle. Integration of the engine with the airframe also means the usual division between aerodynamics and propulsion does not exist. Without the benefit of similarly performing vehicles, hypersonic air-breathing development has focused on ground-based testing facilities and the exploitation of computational modelling. These are complementary technologies as experimental facilities help validate computer models and in return, computer models aid the understanding of flow interactions observed in experiments.

Presently, impulse facilities such as shock tunnels and expansion tunnels [212, 5], are the most effective method of experimentally reproducing the flow conditions experienced by a scramjet. These facilities are capable of reproducing both the high temperatures and high pressures of hypersonic flight. The trade-off for reproduction of the hypersonic flight condition are short test times, of the order of 0.1 to 100 milliseconds. Experimenters must also contend with uncertainty in flow quality, a hostile testing environment, a limited testing range, scaling issues resulting from matching flow properties with a subscale flight model, long turn-around times, and the still developing capabilities of flow and force measurement techniques.

Computational modelling avoids many of the constraints imposed on experimentalists. Provided the computing power is available, detailed flow analysis of complex geometries is possible over the entire hypersonic trajectory. It is also possible to conduct flight dynamics simulations with numerical vehicle models or empirical performance data. The primary limitations of a computational approach is the availability of sufficient computing power for accurate simulation in a reasonable time scale, and the accuracy (and/or completeness) of the physical models used to represent the flow and its processing features. State of the art computation is a fully three dimensional real gas analysis of a complete vehicle [223, 153]. However, depending on the needs of the study, reduced order models [219, 12] may provide sufficient insight into the performance of a hypersonic vehicle. Existing vehicles are also making use of developments in computational analysis. One potentially fruitful area of research is the optimization of reentry vehicles. Uncertainty over the interaction of the ablative surface of reentry vehicles with hypersonic flows, has generated over-designed configurations which limit the potential payload return. The level of detail available in computational fluid dynamics (CFD) has promoted it as a valuable tool in the actual design of real components.

The potential of numerical analysis in aircraft design was demonstrated in the successful development of the Pegasus launch vehicle. Pegasus was designed as a small payload launcher through a joint venture between Orbital Science Corporation and the Hercules Aerospace Company [108]. Shown in Figure 3.2, Pegasus is a three-stage, winged space booster, configured for launching from a carrier aircraft. Setting the development of the vehicle apart was the sole use of computational aerodynamic and fluid-dynamic methods for the purpose of aerodynamic design and analysis [153]. Aiding the success of the design approach was the relatively simple geometry where the propulsion system is uncoupled from the airframe, accessible rocketry experience, and the availability of empirical data from the X-15 rocket plane to validate computational codes. A range of methods were used, from panel methods and other relatively simple engineering approaches, to the numerical solution of the Navier-Stokes equations. Rapid aerodynamic analysis techniques were used for the majority of the configuration, with more computationally expensive approaches left for localized interaction zones such as control surfaces.

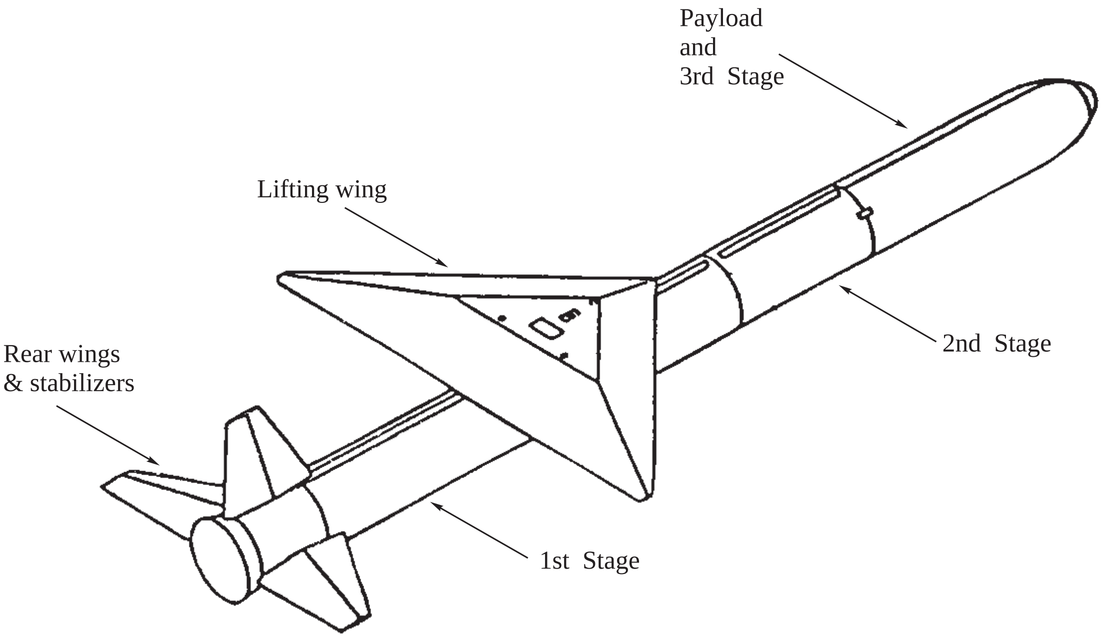


Figure 3.2: General configuration of a Pegasus vehicle - an air-launched solid-propellant space booster with wings - used for launching small payloads into orbit. (Source: Reference [153])

Pegasus is particularly relevant to this study due to its hypersonic flight capability and the similar target market it shares with the air-breathing launcher presented in this thesis. After first stage burnout, Pegasus reaches Mach 8.7 at 60 km altitude, a velocity of roughly 2.7 km/s. On the velocity-altitude map shown in Figure 3.1, the trajectory is similar to the shuttle ascent. Like the shuttle, the Pegasus second stage accelerates the vehicle through the velocity range of a scramjet vehicle, but at a much higher altitude.

Unlike the Pegasus vehicle, hypersonic air-breathing flight is not an extrapolation of existing technologies. After fifty years of scramjet research, the continuing focus of experimental studies is on component analysis, specifically the engine, rather than full vehicle simulation. To date, the generation of broad range performance information has been largely in the realm of computational techniques. The early evolution of concept geometries was supported by simple computational models which allowed rapid determination of the airframe performance. A collection of routines based on “non-interfering constant pressure finite-element analysis” [82] were used to satisfy preliminary design requirements for drag, lift, and moment coefficients, and some control derivatives [147, 159]. These relatively simple engineering approaches have been used on a range of hypersonic vehicles including the North American X-15 and the space shuttle. They have also found application in air-data calibration [117] and as a comparative tool for experimental pressure measurements for blunt body flows such as those around reentry vehicles [198]. For these cases, the application of Newtonian flow analysis [16, 6] is often suitable, as the shock shape follows closely that of the blunt body exposed to the flow. The scramjet vehicle has a propulsion system integrated with the airframe so, in addition to impact methods for the external airframe, the complete vehicle model must also include internal flow processing. See, for example [39].

For the purpose of this thesis, the role of the scramjet vehicle performance model is the computation of forces and moments which, together with the mass properties of the vehicle, allow the time integration of the flight dynamics. Being a complex function of shape and motion, the aerodynamic and propulsive modelling dominates the development of a numerical flight simulation. One of the approaches for encoding the vehicle operation is the assembly of an aerodynamic, propulsion and control coefficients database. In the case of longitudinal hypersonic flight, the database would be discretized with respect to flight speed, altitude, angle of attack and actuator position. An alternative approach is to have the aerodynamics and propulsion simulated as required by the flight dynamics integration routine. Simulation as required offers greater flexibility but at the expense of computation time. Presently the time required for full vehicle simulations through CFD represents a very large computational cost, prohibiting the fashioning of a complete system model for a flight simulator. It is possible however, to reduce the computational effort by representing the flow paths as two-dimensional or axisymmetric. For this thesis, a simplified geometric representation of the axisymmetric scramjet configuration is considered sufficient for longitudinal flight simulation and the assessment of flight control performance. Two-dimensional flow paths are featured, for which analytical and quasi-numerical methods are used to represent principal flow phenomena. It is understood that the practicality of simplified analysis is contrary to the expected reality where complex multidimensional flow effects are exploited, however, the exercise of trying out new control design ideas does not require such precise flow path simulation.

The remainder of this chapter details the numerical flight simulator, built to portray the operation of a hypersonic air-breathing launch concept. Following an introduction to the design specifications of the aircraft, models are developed to represent the physical properties of the vehicle, hypersonic aerodynamics and propulsion performance, the atmosphere, and the vehicle motion. Some details on the construction and operation of the longitudinal guidance and attitude control modules have been provided in Chapter 2 and are developed further in Chapters 4 and 5.

3.1 Simulator Overview

A numerical simulation environment called FACDS (Flight and Control Design Simulator) was constructed as a tool for the development and evaluation of control strategies for hypersonic aircraft. Incorporated within FADS are the following code modules:

- an instantaneous aerodynamics and propulsion vehicle performance model, described by one and two-dimensional quasi-numerical flow theories. Parametric uncertainty is also represented by randomized perturbations to the fuel centre of mass, engine performance and control effectiveness.
- an environment module described by a standard atmosphere model with turbulence and temperature perturbations, and a variation in the local gravity with altitude.
- a dynamics and kinematics module describing rigid-body 6 degree-of-freedom motion about a spherical, rotating Earth.
- methods for the numerical integration of the flight dynamics model.
- control modules for the generation of guidance and inner-loop commands.
- optimization procedures for control design, headed by a genetic algorithm, and supported by a Nelder-Mead simplex method.

Along with these primary modules there are numerous peripheral modules which, among other things, provide performance assessment from flight state histories [18].

A supporting guide to the hierarchy of the simulation modules is shown in Figure 3.3. The flight simulation component of FACDS is described by the flow diagram shown in Figure 3.4. FACDS integrates the dynamics of an aircraft HABV, in a discrete time simulation, partitioned by the sampling timesteps for the guidance loop and inner loop control, $(\Delta t_{\mathrm{g}}, \Delta t_{\mathrm{c}})$. The state vector $\boldsymbol{x}$, describes the vehicle position, orientation, attitude rates, velocities, mass, and actuator settings,

$$\boldsymbol{x} = \left[(R, \mu, \lambda), (\psi, \theta, \phi), (p, q, r), (u, v, w), m, \left(\theta_{\mathrm{e}}, \dot{\theta}_{\mathrm{e,cmd}}, \nu_U, \nu_L\right)\right]^T. \tag{3.1}$$

Initialization of the flight simulation requires the specification of the vehicle geometric and mass properties, the initial state vector $\boldsymbol{x}_0$, and the controller configuration. For each guidance update, the trajectory module provides the target altitude $h_{\mathrm{ref}} = f(q_\infty, V)$ given the aircraft velocity, and for this thesis, a predefined freestream dynamic pressure q_∞. This information is fed into the longitudinal guidance routine, along with state data, to return a reference attitude and a trim actuation position. The inner control loop represents the longitudinal attitude tracking and stabilization component of the flight control. Actuator

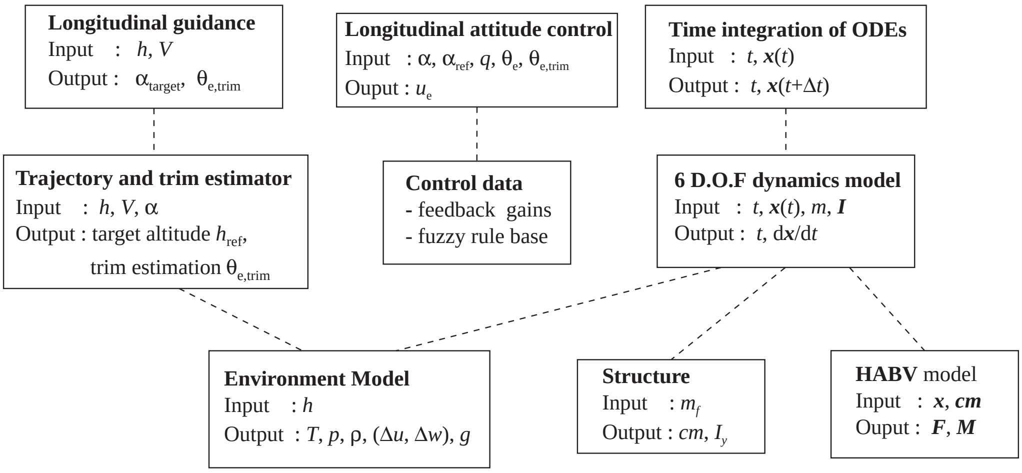


Figure 3.3: Hierarchy of code modules for HABV simulation. Higher level blocks make use of information from lower level blocks.

commands are describe by a control function dependent on the angle of attack error, pitch rate and the elevator trim error, $u_{\mathrm{e}} = f(\alpha_{\mathrm{err}}, q, \theta_{\mathrm{e,err}})$.

The flight dynamics are integrated between longitudinal controller updates, using a single step integrator over the integration timestep Δt, and working on the state vector $\boldsymbol{x}$. The block denoted by HABV represents a numerical simulation of the instantaneous aerodynamics and propulsion of the hypersonic air-breathing aircraft. No timescales are used to represent the flow processing dynamics, implying an instantaneous change in flow structures. With each call from the integrator to evaluate the state derivatives, HABV generates the net forces and moments acting on the vehicle. It also translates the fuel command settings, (ν_U, ν_L), into fuel flow rates which define the rate of mass loss of the vehicle. Though simplified models have been used within HABV, during the control design phase the aero-propulsive modelling represents 99 % of the computational effort. Other studies generally use equivalent analytical statements of the general performance characteristics [43], use an analytical or numerical vehicle model to establish a table of aerodynamic and control derivatives [39], or schedule a collection of linearized models.

3.2 Scramjet Vehicle Design

The scramjet vehicle concept studied for this thesis was an adaption of the axisymmetric configuration investigated at The University of Queensland (UQ). In Chapter 1 the vehicle was introduced as a potential acceleration stage in a launch vehicle concept for small payloads. Figure 1.5 (page 10) showed a possible configuration for the scramjet stage, incorporating many of the design features which were principally evolved from decades of scramjet engine research at NASA. In particular, the engine concept features round combustors, swept compression surfaces, a cut-back cowl, and a circumferential distribution

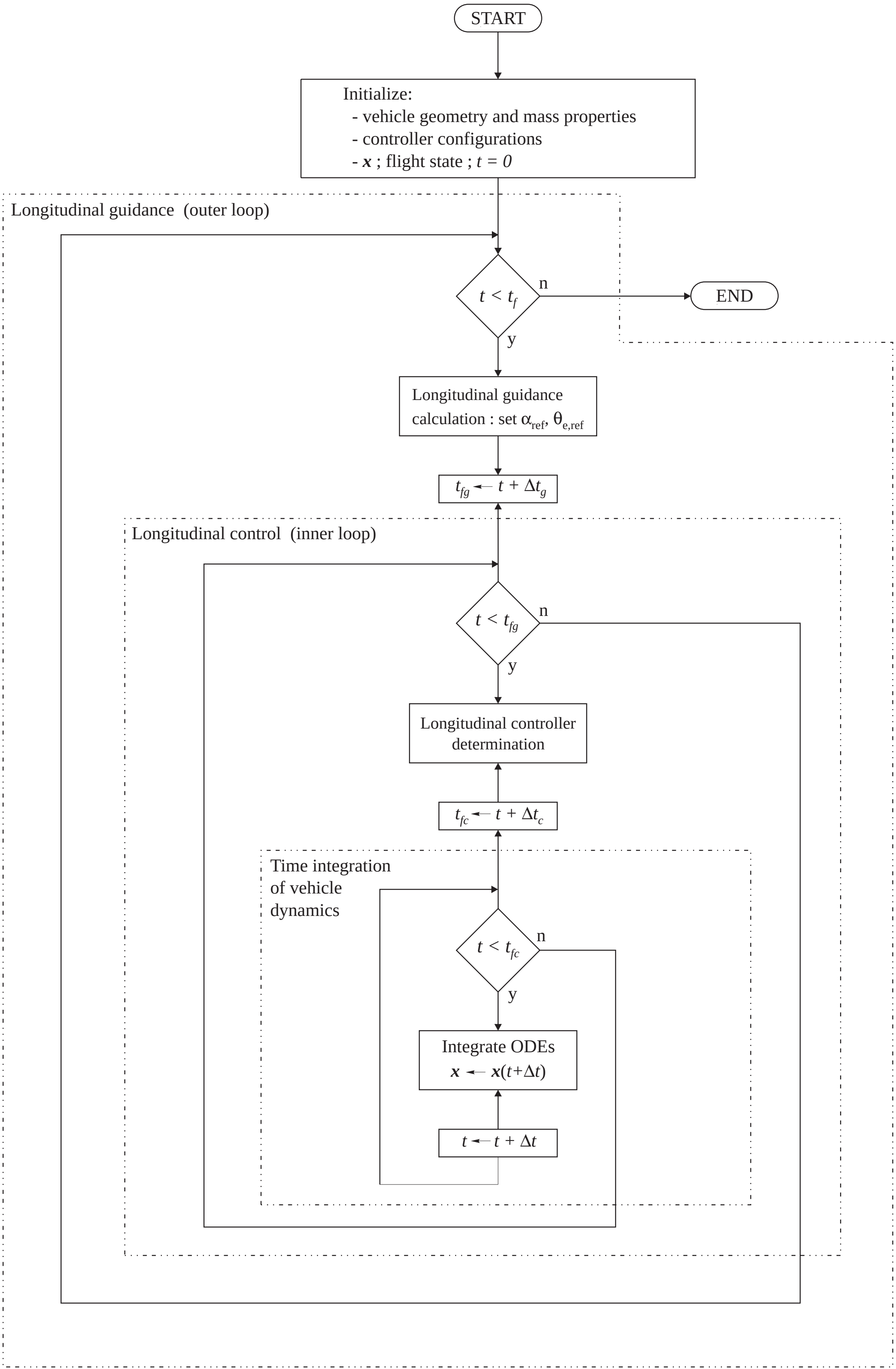


Figure 3.4: HABV flight simulation flow diagram.

of the engine modules around the vehicle axis of symmetry. Each engine module incorporates a forebody-inlet region, a combustor, and a nozzle or thrust generating expansion surface. Accurate representation of these features for the purpose of generating a numerical model, is a demanding task. The inevitable computational simplification must be traded against the importance of realistic engine operation which, amongst other aspects, will exploit multi-dimensional flow effects.

This thesis considers only the longitudinal performance of the scramjet powered stage. Its representation as a vehicle model within the flight simulator must facilitate the calculation of airframe and engine performance characteristics and the estimation of mass properties. The principal geometric simplification is the use of two-dimensional flow-paths, providing a box-section representation rather than the round body associated with an axisymmetric geometry. Importantly, the basic shape of the vehicle and engine are maintained, along with the operational dependencies on angle of attack and flight conditions. A fixed geometry is used with reference to a single flight condition, providing the best compromise for broad range operation. Acting as an accelerator, the vehicle is generally not at the design condition, resulting in significant variation in longitudinal performance, dependent on the engine and airframe flow processing. Consideration has also been given to the description of lifting and rear stabilizing surfaces, and the positioning of the payload and fuel, sufficient to reasonably represent the internal mass distribution.

3.2.1 Engine Specification

The level of integration of airframe and engine featured in scramjet vehicle designs have, for the axisymmetric configuration, earned the demonstrative title of a *flying engine* [105]. The implication of the extreme integration is that the engine geometry largely defines the dimensions of the vehicle. Here, a fixed geometry scramjet is described, using a nominal flight condition set at the high Mach number end of a typical hypersonic air-breathing trajectory. This represents a compromise for propulsive operation when performance is required over a range of Mach numbers. According to Stalker [203], the losses in net thrust for off-design operation are less when the propulsive duct is configured using a design Mach number at the maximum end of the range. The engine design condition was thus set at a flight Mach number of 15, at an altitude of 30 km, and with zero angle of attack. This places the vehicle at the edge of the air-breathing corridor, customarily defined by a dynamic pressure range of 24 to 100 kPa [105], see Figure 3.1. The design dynamic pressure of roughly 188 kPa, is recognized as exceeding the structural predictions of the past twenty years. Favour has been afforded to engine operation, in preference to the vehicle structural capability. Though lower dynamic pressures are preferable for designing a vehicle structure, the scramjet engine becomes less effective.

The key engine elements of a scramjet are the inlet, combustor, and nozzle. Figure 3.5 shows the arrangement of these elements for an axisymmetric scramjet. It is representative of a baseline concept where no effort has been made to optimize the exposed surfaces for broad range performance. The practical need for optimum broad range performance can also be augmented with a variable geometry engine. In particular, a variable inlet geometry can improve the efficiency of the inlet in capturing the freestream air. Henry and Anderson [98] showed a maximum performance increase of 16% with a variable geometry. When traded off with an associated penalty due to increased system complexity and increased weight, their conclusion was a preference for the fixed inlet scramjet. Interestingly, it is the three-dimensional features of the inlet which make a fixed inlet geometry feasible. Three-dimensional compression reduces the overall turning angle needed for broad range operation, and the swept inlets shown in Figure 1.5 aid inlet starting at low speeds and reduce boundary layer separation from the inlet wedge. Variable geometry has also been considered for the combustor and nozzle regions of the engine. A variable nozzle geometry has the added potential of acting as a control actuator. It is likely, however, that the only variable feature of the first generation scramjets would be through providing dual mode operation, where the engine is able to operate in both ramjet and scramjet modes [227, 50].

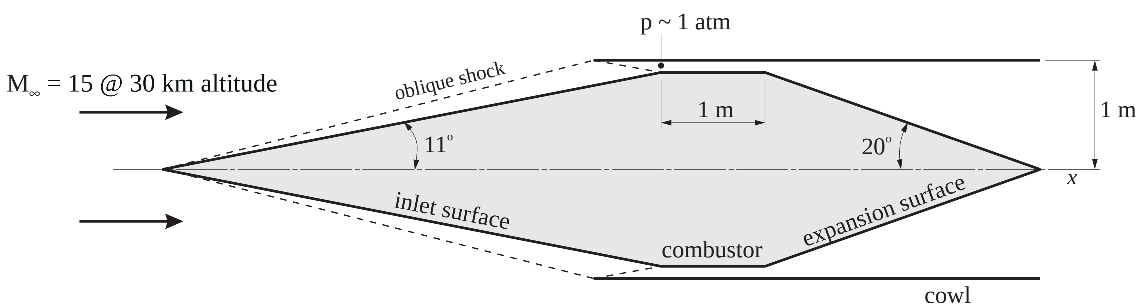


Figure 3.5: Geometric specification of the Mach 15 scramjet engine with two-dimensional flow paths. A compression ratio of 12.213 has been used

The purpose of the inlet is to direct the freestream flow into the combustor, whilst compressing the flow to a pressure and temperature desirable for combustion. If structural considerations were not a factor, the most effective method of doing this would be through isentropic compression of the freestream. Though providing shock free compression, such an inlet is prohibited by the low structural strength of a long sharp nose and by excessive viscous drag and heat transfer. Practical hypersonic inlet designs typically incorporate a combination of external and internal compression through oblique shocks, thereby providing half angles large enough to allow the construction of structures that can withstand the high dynamic loads. There is also a desire to control the inlet bow shock with changes in angle of attack and flight Mach number, especially in relation to the engine cowl lip [133]. The physical limitations are excessive wave drag if the bow shock is too far away from the vehicle, excessive local heating with shock impingement, and potentially harmful shock reflections propagating through the engine when the bow shock is swallowed into the cowl. Several inlet designs for the axisymmetric scramjet where investigated by Craddock [48]. A short multi-shock inlet with a bent cowl was favoured, due to its low drag and being the least prone to boundary layer separation. This also appears to be the configuration used for the scramjet flight tests performed in Russia during the 1990s, and discussed in Section 1.2.

The inlet geometry shown in Figure 3.5 is the simplest representation of a mixed external and internal compression system. It consists of a straight inlet surface orientated at 11° to the vehicle axis of symmetry x, and a cowl aligned parallel to x and positioned to provide a vehicle half height of 1 m. The shock produced by the the inlet wedge compresses the freestream flow and directs it along the inlet surface. At the design condition this shock intersects the cowl leading edge and the corresponding reflected shock redirects the flow uniformly into the combustor, with the shock being cancelled at the upstream corner of the combustor. The net result at the design condition is shock-free uniform combustor flow at a pressure of approximately one atmosphere. This is considered sufficient on the basis that, for a reasonable length combustor, the combustion kinetics are fast enough to bring the combustion composition nearly to its equilibrium state [121]. Inlet performance away from the design point is discussed in Section 3.3.1.

A constant area combustor sits downstream of the inlet, aligned parallel to the vehicle axis of symmetry. Experimental studies at The University of Queensland [229] utilized a 350 mm long combustor for tests on a Mach 7.6 Composite Scramjet Motor, figuring this was long enough to allow near complete mixing of the fuel with the incoming air. The length of 1 m used here, follows the recommendation by Kerrebrock [121], with consideration to the rate of energy release from the combustion process for an engine with a flight speed of Mach 15, and assuming a combustor inlet pressure of 1 atm. There is an alternative view supported through numerical studies by Craddock [48], that a short combustor is desirable in terms of limiting viscous losses, with the combustor and nozzle combination providing greater net thrust. This conclusion was also based on the use of an axisymmetric combustor rather than the arrangement of engine modules used here. Practical engine designs may also utilize relief through a diverging duct [50]. This is to counter the heat addition limit before steady flow breaks down in a constant area duct which, for low supersonic entry conditions, means very little heat can be added.

Expansion of the supersonic combustion products is the mechanism of thrust generation in the scramjet. This is achieved here by a straight 20° surface. The cowl has been extended axially to the extremity of the nozzle surface, primarily to simplify the nozzle flow simulation. Practical arrangements generally have the cowl terminated at a length sufficient to capture the expansion fan generated at the upstream corner of the nozzle. Like the inlet and combustor surfaces, significant performance gains are realizable through optimization of the expansion surface [112, 48].

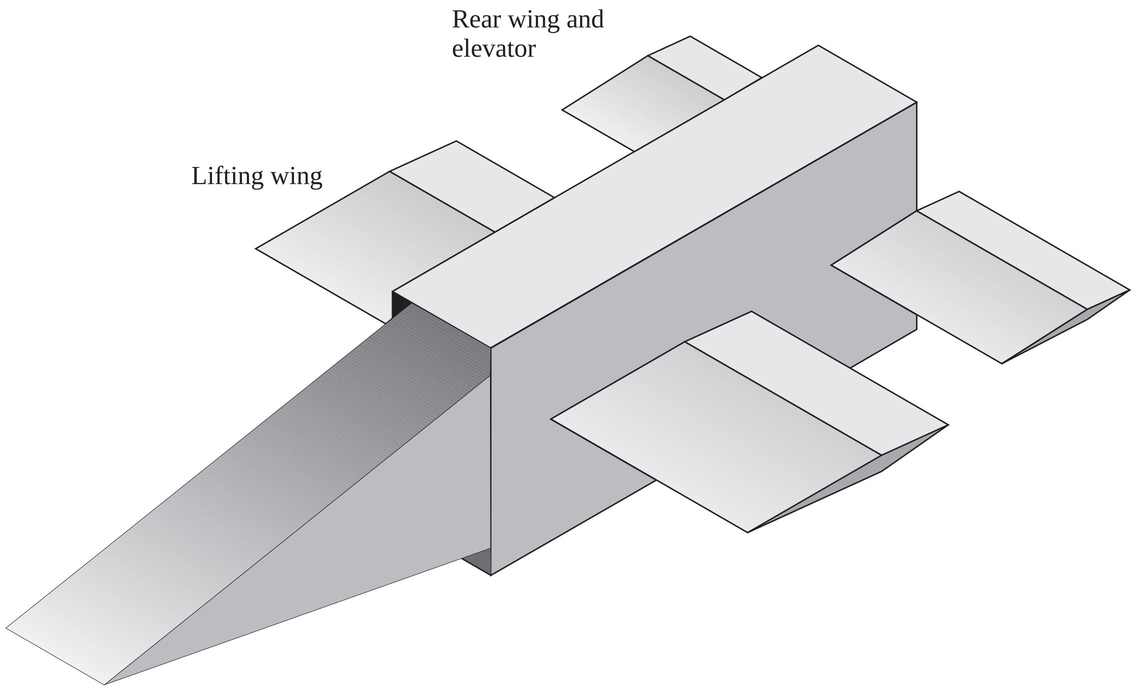
3.2.2 Overall Vehicle Configuration

The engine geometry of Figure 3.5 defines the primary flow paths necessary to determine the propulsive performance of the engine. It also describes the basic dimensions of the overall vehicle, in affect shaping and sizing the fuselage. To complete the physical description of the vehicle, the payload and fuel is positioned within the internal volume, structural densities are specified, and lifting wings and stabilizing surfaces are defined. Without access to any advanced scramjet vehicle designs, it is not possible to accurately represent the distribution of elements within the vehicle. For the structure, densities of advanced materials envisaged for hypersonic applications are used. Despite the conceptual approach, the mean vehicle density compares favourably with the Pegasus vehicle. It is worth repeating that assembling a practical hypersonic air-breathing vehicle is not simply a matter of adding the necessary flight components to the engine. Usage of the internal volume is critical for vehicle stability, and the necessary optimization of the airframe and propulsion system combination means the entire vehicle appears in the design equation from the beginning. In contrast, many traditional aircraft can be designed using the engine simply as a peripheral component to be added to the fuselage.

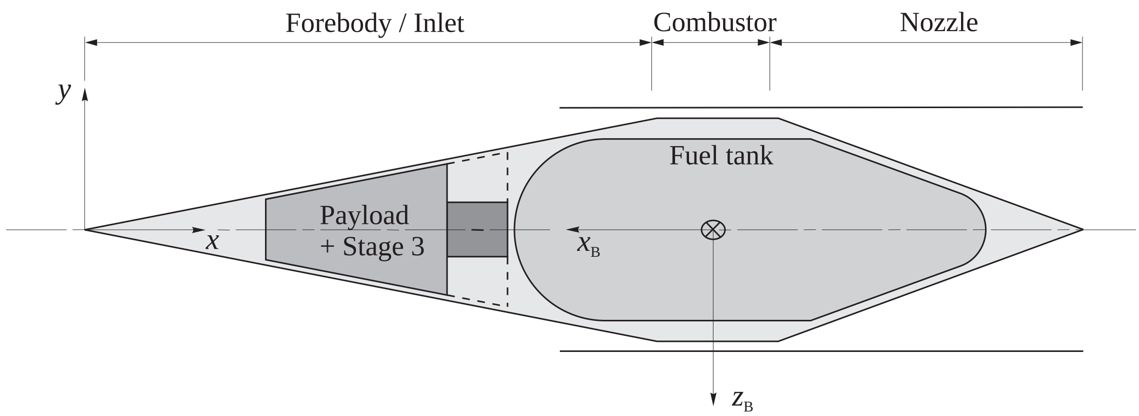
The physical layout and properties of the scramjet powered vehicle are summarized in Figure 3.6 and Table 3.1. In most respects, the simplification of the axisymmetric vehicle to that with two-dimensional flow paths means the vehicle is treated as a two-dimensional vehicle, with the vehicle depth set to 1 m. For convenience, two coordinate frames are used for defining the vehicle geometry. The leading edge of the vehicle body is used as a fixed reference for the initialization and storage of vehicle dimensions. Frame (x, y) as defined in Figure 3.6, also provides a fixed reference for element centres of pressure. These are updated relative to the vehicle centre of mass, as the mass of the vehicle changes during flight. The axes (x_B, z_B) represent the body-fixed coordinate frame F_B, used in the vehicle dynamics model. Frame F_B has its origin at the vehicle centre of mass, with axes aligned along the principal inertia axes of the vehicle.

Surface Description and Force Accounting:

The basic structure of the vehicle is an assemblage of uniform surface elements. Their geometry is specified by a general spatial distribution, an area, and vectors for the centre of area and an inward surface normal. The surface normals are maintained in terms of (x_B, z_B), providing a direct translation to body axial and normal forces. For all except the nozzle internal surfaces, a uniform surface pressure is assumed. Since the origin of



Overall vehicle shape



Sectional view of scramjet engine.

Figure 3.6: Overall physical layout of the scramjet vehicle.

Table 3.1: Physical definition of the Mach 15 two-dimensional scramjet vehicle. See Figures 3.6 and 3.5 for schematics of the vehicle assemblage.

Feature	Part description	Value
General	Overall length.	8.246 m
	Vehicle height.	2 m
	Inert mass.	2487 kg
	Maximum vehicle mass.	4972.0 kg
	Maximum pitch axis inertia.	11825 kgm^2
	Full fuel load centre of mass, (x).	5.2 m
	Surface element density.	30.0 kg/m^2
Inlet	Leading edge location, (x, y), see Figure 3.6.	(0,0) m
	Ramp angle.	$11°$
	Compression ratio, or inlet area ratio.	12.213
Combustor	Length.	1 m
	Combustor area, or height.	0.0819 m
Nozzle	Ramp angle.	$20°$
	Upstream height.	0.0819 m
Cowl	Design shock angle.	$14.319°$
	Leading edge location, (x, y).	$(3.918, \pm 1.0)$ m
Fuel	Ethane.	C_2H_6
	Liquid density.	544 kg/m^3
	Heating value or energy density, H.	47.484 MJ/kg-fuel
	Stoichiometric mixing ratio, mass basis.	0.0624 kg-fuel/kg-air
	Nominal combustion efficiency, $\eta_c/100\,\%$.	0.75
Tank	Maximum width.	1.56 m
	Storage capacity for ethane.	2485 kg
	Nominal centre of mass, (x).	5.41 m
Payload	Mass, including stage 3 motor.	500 kg
	Leading edge, (x).	1.5 m
	Axial location of mass centre.	+1.19 m
Lifting wing	Density.	25.0 kg/m^2
	Angle of attack.	$3°$
	Axial location.	5.794 m
	Area.	8.25 m^2
	Fore section half angle.	$3°$
	Aft section half angle.	$6°$
Rear wing	Density.	25.0 kg/m^2
	Half angle.	$3°$
	Axial location.	8.2 m
	Area.	3.0 m^2
Elevator	Half angle.	$6.0°$
	Axial length.	0.43 m
	Area.	1.5 m^2
	Actuation limit.	$20°$
	Acutation rate limit.	2.0 rad/s

F_B changes with fuel consumption, the centre of area vectors are maintained in terms of (x, y), and transferred to frame F_B as required, for the summation of element moments.

Mass Properties:

Overall, the physical properties of the vehicle are assembled in two parts, a fixed component containing all structural elements and fixed peripheral elements, and a variable fuel component. This allows an initial evaluation of the vehicle components to provide the inert values for mass, inertia, and centre of mass $(m_, I_y, cm)_{\text{fixed}}$. Included in the estimation are the main structural elements, payload, tank, and all external aerodynamic features. The vehicle mass is tracked through the fuel consumption rate, and a simple updating procedure combines the fuel properties with the inert elements to provide the overall vehicle centre of mass and inertia.

$$\begin{aligned} m &= m_{\text{fixed}} + m_{\text{fuel}} \\ cm_x &= \left(cm_{\text{fixed}} m_{\text{fixed}} + cm_{\text{fuel}} m_{\text{fuel}}\right)/m \\ I_y &= I_{\text{fixed}} + I_{\text{fuel}} + m_{\text{fuel}}(cm_x - cm_{\text{fuel}})^2 \end{aligned} \tag{3.2}$$

Table 3.1 contains the default physical properties for the Mach 15 scramjet vehicle, carrying a full fuel load. Since the fuel rate changes along the flight trajectory and with vehicle attitude, additional nominal fuel loads were defined along the trajectory, see Chapter 5. It is worth comparing the physical representation of the scramjet properties to those for the second and third stage combination of the Pegasus vehicle [109]. The scramjet's inert mass density is double that of the Pegasus vehicle, which may be supported by the argument that part of the weight saving from not having to carry oxidant for combustion, can allow a greater empty weight to improve vehicle ruggedness. Overall, the mean vehicle density is roughly equivalent to that for Pegasus.

Apart from the payload and fuel, each element of the vehicle is represented as a rigid panel of uniform thickness, with a mass per unit area reflecting the properties of the advanced materials that are expected to be used for such vehicles. Factored into the structural densities is a component associated with expected cooling requirements, which may take the form of ablative material or an active cooling mechanism. Inertia calculations for each structural element, including the wings and elevator, follow those for a thin plate of uniform density, see for example [27]. Without any deformable components in the dynamics model, elevator motion is not modelled as a contribution to the variation in vehicle inertia or center of mass. For the fuel and payload components, we assume a uniform distribution of their respective masses within the volume available to them. Further, the inertia calculation of the fuel is based on the volume of fuel remaining, assuming a constant storage density and a centre of mass fixed relative to the tank. Modelling uncertainty has been represented through a perturbation in the centre of mass of the fuel, see Section 3.5.

Fuel and Payload:

Modern rocket propelled launch systems rely on the combination of highly energetic and light propellants to achieve a high specific impulse. For the space shuttle this is realized with the combination of hydrogen and oxygen. Liquid hydrogen is the generally accepted fuel for high speed applications due to its high energy density, high combustion rate and high combustion temperature. However, due to the low molecular weight of hydrogen, external tanks are needed to provide the shuttle with sufficient storage to execute the launch. Such an arrangement is the antithesis of the integrated engine/airframe configuration of hypersonic air-breathing vehicles. Using hydrogen for the air-breathing launcher is therefore generally associated with vehicle concepts large enough to store the fuel internally. By example, the operational concept behind the NASA Hyper-X project, is a hydrogen fueled vehicle of approximately 61 m in length, compared to the 35 m long shuttle orbiter.

On a small scale launcher like that envisaged for the axisymmetric scramjet, the higher liquid density of hydrocarbons combined with reasonable specific impulse values, make them a suitable fuel. An additional advantage is the much simpler storage needs than the cryogenic storage needed for hydrogen. Because supersonic combustion timescales are of the order of milliseconds for a combustor of length 1 m, the critical parameters for selection of a hydrocarbon fuel are the ignition and reaction delay times at combustor conditions [160, 104, 229]. Ethane, being the fastest igniting of alkane hydrocarbons, has been considered suitable for small scale launch vehicle concepts [229, 160], and is used here.

As with any aircraft, the internal mass distribution of the fuel, payload, and ancillary equipment, is critical for vehicle performance. The most important issue to address is the location and variation of the vehicle centre of mass. Accurate representation of this is required to assess the inherent stability of the airframe, and to position and size stability augmentation devices and lifting surfaces. For hypersonic aircraft, though, the possible configuration options are limited due to the available space in a slim airframe, where the external surface area is necessarily minimized. Further, there are operational issues relating to delivering a payload to orbit, which constrain the internal arrangement. For the axisymmetric concept [204], final orbit insertion is expected to be achieved through a rocket motor and the positioning of this stage relative to the payload, as well as the primary fuel supply, is restrictive.

The maximum cross-sectional area is used to house the fuel tank, see Figure 3.6. This roughly coincides with the location of the vehicle centre of mass, thereby minimizing the influence of fuel consumption on the rotational behaviour of the vehicle. The maximum fuel capacity has been set at 2900 kg, providing roughly 290 seconds of engine operation. Though ethane is stored as an LPG (liquefied petroleum gas), the fuel tank mass uses the approximation for cryogenic storage of 10 kg per cubic metre of propellant [39]. Mass not needed for cryogenics is assumed available for additional peripheral components. Positioned forward of the fuel is the payload compartment which includes the stage three orbit insertion motor. The arrangement supports a possible scenario whereby the forebody is disposed of before releasing the payload for the final stage of inserting into orbit.

Lifting Wings and Stabilizing Surfaces:

Wings are used here to generate enough lift to allow the vehicle to travel a near-level flight trajectory at zero angle of attack, and to augment longitudinal stability. To limit the total wing area a lifting wing and rear stabilizing wing are configured similar to that used for the Pegasus vehicle, shown in Figure 3.2. The swept back design typical of high speed vehicles is not reproduced since two-dimensional flow paths are assumed throughout the vehicle. Operating along a constant dynamic pressure flight trajectory, the lifting wing requirements are relatively independent of flight speed. Though the lift generated decreases gradually as the vehicle accelerates and climbs, there is also a drop in the fuel load. Depending on the actual trajectory followed, there is the possibility of a net positive lift developing which would aid the gradual climb of the vehicle. With this in mind, the lifting wing was configured to provide sufficient lift with a full fuel load, at the low speed end of the flight trajectory.

Figure 3.7 shows the general configuration of the wing, defined by the parameter values recorded in Table 3.1. Axially, the wing is located using the division between the fore and aft sections of the wing. Tests showed the optimal axial position for a given wing size, to be relatively independent of flight velocity, based on the acceleration, attitude, and altitude hold performance of the vehicle for control fixed operation. The symmetric wedge sections are orientated relative to the vehicle body with an angle of attack, which, in combination with the wing surface area produces sufficient lift to approximately match the maximum weight flight condition.

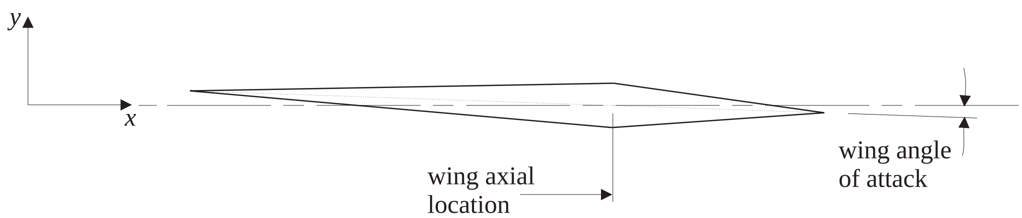


Figure 3.7: Lifting wing specification.

The principal stabilizing feature used in this thesis is a rear wing and elevator arrangement. A number of configurations were investigated for their capacity to control the vehicle. Figure 3.8 shows the arrangements and their mode of operation. In the case where the rear wing is fixed and the elevator a wedge profile, the control moment dead zone caused by shadow cast by the rear wing, significantly retards the flight control performance. To achieve stable flight the rear wedge arrangement must be prohibitively long. The two alternatives where symmetrically operated rear flaps, and an all-moving rear wing arrangement. These proved to be roughly equivalent in their flight control performance. The all-moving wing allows a lower surface area and is used for the simulations presented in this thesis. When sizing the rear wing surfaces, it was noted that successful flight control design required trim angles of roughly the same order as the angle of attack.

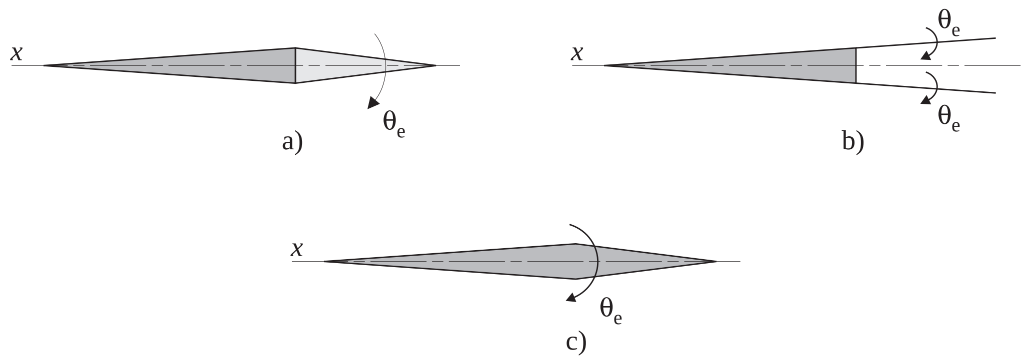


Figure 3.8: Possible arrangements for the rear wing and elevator combination: a) fixed rear wing with an elevator wedge, b) fixed rear wing with elevator flaps nominal set parallel to the wing surface, and c) an all-moving rear wing.

3.3 Airframe and Propulsion Models

The practical realization of a scramjet powered vehicle relies on a sound understanding of the real flow features encountered in hypersonic flight. High temperatures, viscous effects, boundary layer action, and three dimensional effects are all features of hypersonic flow, and all significantly impact on vehicle design and operation. As previously mentioned, numerical simulation of these features is very demanding and are not without uncertainty. The approach typically employed is to approximate the detailed mathematical function needed to obtain accurate performance characteristics. Simplified models have been used to provide performance insights on a range of studies, and are considered sufficient for a stability and control analysis of a concept vehicle.

An example of the level of modelling is the widely used collection of engineering methods representing the Aerodynamic Preliminary Analysis System (APAS) [159, 49, 60]. These impact-type finite element models simulate the external aerodynamics of the airframe, and have been used to estimate the forces and moments generated by various control actions on a range of hypersonic vehicles [147]. Simulating the engine flow requires additional analysis techniques, as in [12, 219, 39]. Armengaud et al. [12] used the approximation of one-dimensional thermodynamic equations to examine performance relationships for the scramjet. They considered the ideal gas model useful for exhibiting the general characteristics of scramjets, reporting a 20% difference in comparison to real gas characteristics. Tsukikawa *et al.* [219] applied a quasi-one-dimensional model for flow through the scramjet, aimed at determining the optimum configuration of the engine.

For this thesis the vehicle simulation task is simplified by describing the geometry with two-dimensional flow paths and by assuming an instantaneous representation of flow structures. A multi-domain quasi-numerical description of the vehicle can therefore be formulated using a combination of simple one-dimensional and two-dimensional gas flow models, whereby, for a given freestream condition, the net forces and moment relating to longitudinal vehicle motion can be evaluated. As the external aerodynamics is relatively straight forward, scramjet engine operation dominates the modeling requirements. Figure 3.9 schematically shows the basic engine processes, segmented by inlet, combustor, and nozzle regions. Freestream air is ingested into the inlet and is processed by a series of oblique shocks, raising its pressure and temperature. On entering the combustor the airflow continues to be supersonic. Fuel is added and combustion of the supersonic flowstream is represented by a simple heat addition model. The nozzle flow is then detailed with an expansion fan model capable of tracking the interaction of a finite number of waves.

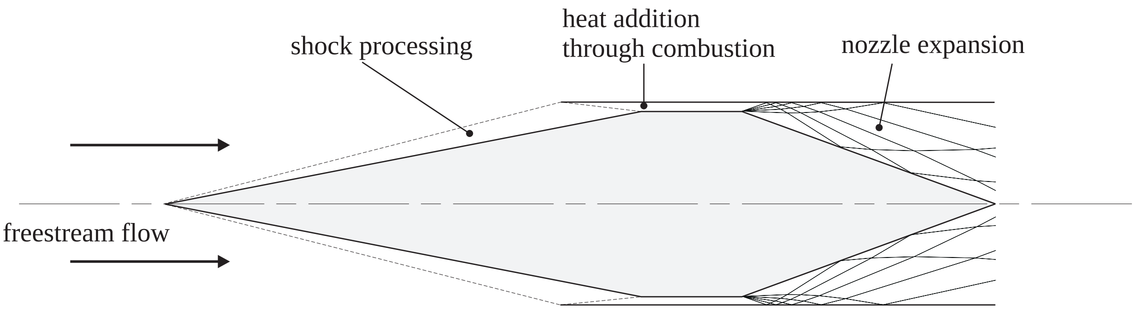


Figure 3.9: Processing of flow through the two-dimensional hypersonic scramjet.

Perhaps the most significant features not represented in this model are those relating to viscous effects. Computational experiments by Craddock [48] on a similarly simple axisymmetric configuration indicated that, even when the inlet, combustor, and nozzle surfaces were optimized, the viscous contribution to drag meant the scramjet engine did not produce a net propulsive thrust. The only redeeming conclusion was that the majority of the skin friction was associated with the axisymmetric combustor of the wrap-around configuration used. Significant improvement is available with the configuration depicted in Figure 1.5, which incorporates separate engine modules with cylindrical combustors. An engineering approach to skin friction such as that used by Chamitoff [39] could be used, but with skin friction forces being of similar magnitude as form drag, it would only serve to augment the vehicle drag to such a degree that it becomes a "net drag vehicle". Accordingly shear forces have been neglected in the propulsive and aerodynamic modelling of the scramjet vehicle in this thesis. Further critical features such as shock-boundary layer interaction, viscous-inviscid interactions, flow transition, mixing and combustion, and radiation, remain serious issues with respect to the performance of the propulsion system, particularly in terms of setting the maximum speed. They are however assumed to be secondary effects with respect to control. The small thrust margin achieved through neglecting these features is an accepted limitation of the simple geometry being used.

Figure 3.10 summarizes the aerodynamics and propulsive analysis of the scramjet vehicle. For a given vehicle state $\boldsymbol{x}$, and centre of mass $\boldsymbol{cm}$, the vehicle model returns the net aero-propulsive forces and moment from the contributions of the inlet (I), nozzle (N), cowl (C), lifting wing (LW) and rear wing (RW).

$$\boldsymbol{F}_{\text{aero-prop}} = \boldsymbol{F}_{\text{I}} + \boldsymbol{F}_{\text{N}} + \boldsymbol{F}_{\text{C}} + \boldsymbol{F}_{\text{LW}} + \boldsymbol{F}_{\text{RW}} \tag{3.3}$$

$$\boldsymbol{M}_{\text{aero-prop}} = \boldsymbol{M}_{\text{I}} + \boldsymbol{M}_{\text{N}} + \boldsymbol{M}_{\text{C}} + \boldsymbol{M}_{\text{LW}} + \boldsymbol{M}_{\text{RW}} \tag{3.4}$$

The environment module provides the atmospheric conditions (T, p, ρ) through a standard atmosphere model augmented by temperature perturbations and turbulence velocities $(\Delta u,\ \Delta w)$. In the following sections the application of the flow processing models is described.

3.3.1 Inlet Flow Processes

The inlet's function is to raise the pressure and density of the freestream air and direct the flowstream into the combustor. A uniform wedge and a cowl section perform this function, through the action of external and internal oblique shock waves. The shock configuration, and therefore the inlet performance is dependent on the freestream flight condition (M, h), defining the flight Mach number and altitude, and the vehicle angle of attack, α. At the zero angle of attack design condition of Mach 15 flight at an altitude of 30 km, the inlet processes the flow through two oblique shocks, as shown in Figure 3.11(a). The primary shock generated by the inlet wedge, redirects the freestream air parallel to the inlet surface, with the shock intersecting the leading edge of the cowl. The delivery of uniform flow to the combustor is provided via cancellation of the reflected shock at the combustor upstream corner. External flow over the cowl for this condition is simply the undisturbed freestream air.

For any non-zero angle of attack, the flow processing in the two engine modules is asymmetric. This can greatly influence the overall performance of the vehicle, not just

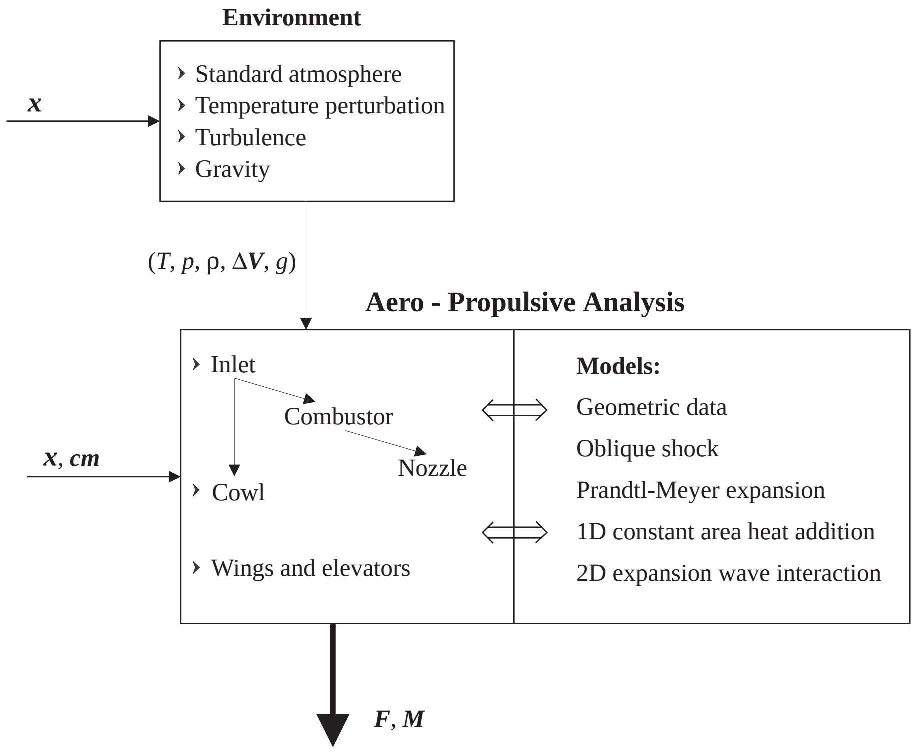


Figure 3.10: Scramjet aerodynamic and propulsion modelling summary. The arrows indicate the flow of data.

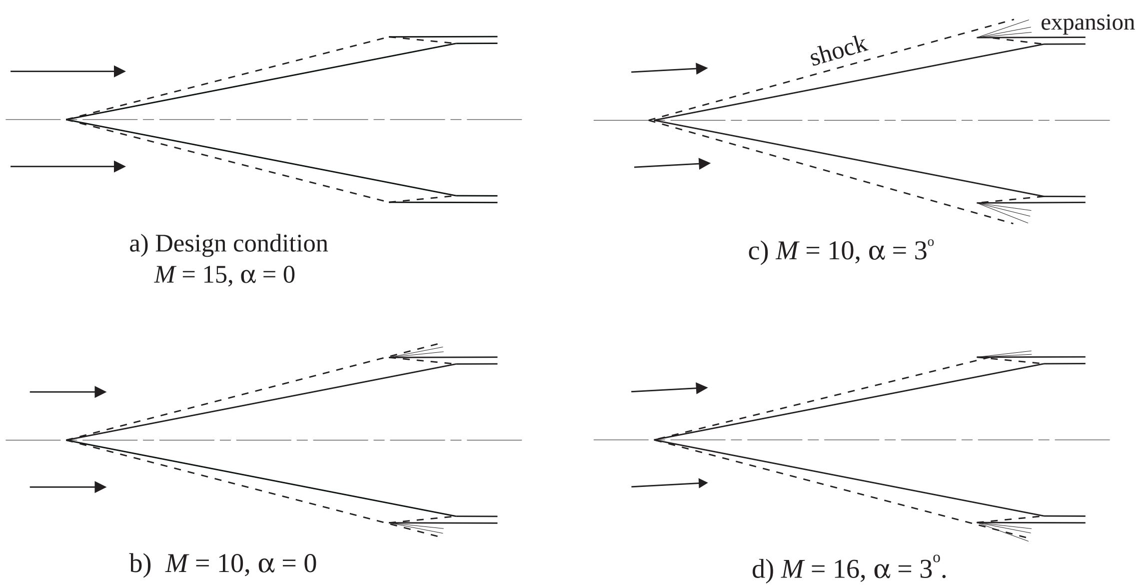


Figure 3.11: Schematic examples of the modelled flow processing through the scramjet inlet for various flight Mach numbers and angle of attack. Note that the vehicle and flow geometries are drawn as true shape.

in the flow conditions presented to the combustor, but also in the destabilizing moment generated by the pressure difference over the long inlet surfaces. The net effect is the application of an operational limit on the possible angle of attack of the vehicle. One clear boundary is to limit the angle of attack to a value less than the inlet wedge angle, thereby avoiding the effective shut down of one of the engine modules. However, the pitching moment generated by the inlet, rises rapidly with angle of attack α, and a practical operation limit of a just a few degrees is necessary. The generation of a stabilizing moment through wings, elevators, or other means, can thus be achieved without excessive thrust penalty. A similar operational tolerance could be expected for the Hyper-X style air-breathing vehicle described in Section 1.2. Although operating at a relatively high α, manipulation of vehicle attitude to maintain optimum engine performance as flight conditions change, is likely to be limited to a few degrees about the nominal condition.

The matching of the inlet shock structure to the inlet geometry as in Figure 3.11, represents inlet processing at a single precise flight condition. To simplify the off-design modelling, the basic two-shock inlet arrangement is assumed at all times. Approximating the inlet flow structure in this manner neglects any further shock interaction in the inlet and any follow on flow features downstream. Figure 3.11(c)-(d) shows a selection of off-design inlet flow structures. Noticeably there is little travel of the primary shock relative to the leading edge of the cowl. Despite the similarity of the flow structures in the Mach 10 examples, the pressure differential between the upstream combustor flows at $\alpha = 3°$ and $\alpha = 0°$ is around 50 % of that for the $\alpha = 0°$ condition, see Figure 3.12. The variation of inlet performance with flight Mach number and angle of attack is shown in this figure. If the vehicle is assumed to follow a constant dynamic pressure trajectory then the pressure downstream of the inlets would increase linearly with Mach number, for the two-shock model.

In Figure 3.13 a more accurate representation of the inlet flow structure simplified in Figure 3.11(d) is shown using a computational fluid dynamics (CFD) simulation [110, 111]. Although there are more waves in the combustor, the additional complexity makes little difference to the integrated pressure over the inlet surface, or to the average flow properties presented to the combustion. The shock interactions are important for the processes in the combustion, but these are considered an issue for engine designers, and beyond the scope of this study.

The external flow over the engine cowl varies according to the position of the primary shock relative to the cowl leading edge. It will generally result in either expansion of the freestream about a small angle, or expansion of the post shock flow through the inlet angle. No shock/expansion interaction has been modelled for flow over the cowl. The movement of the shock about the cowl leading edge is a critical feature, as step changes in pitching moment may result from changes in pressure over the external cowl surface.

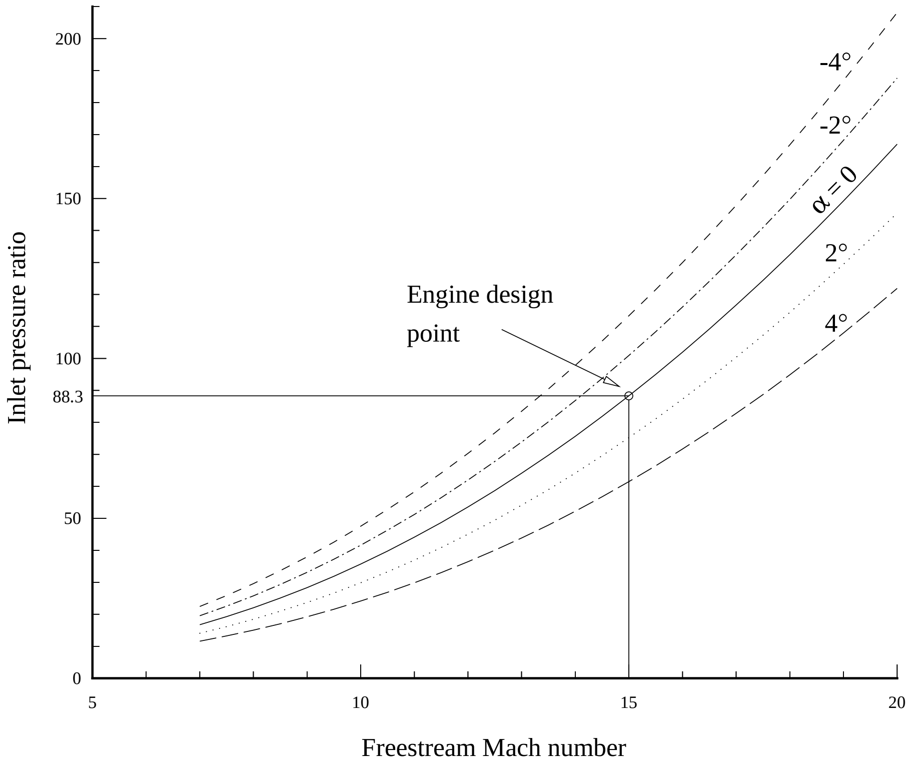


Figure 3.12: Inlet pressure ratio for the engine module on the upper half ($-$ve z_B) of the vehicle.

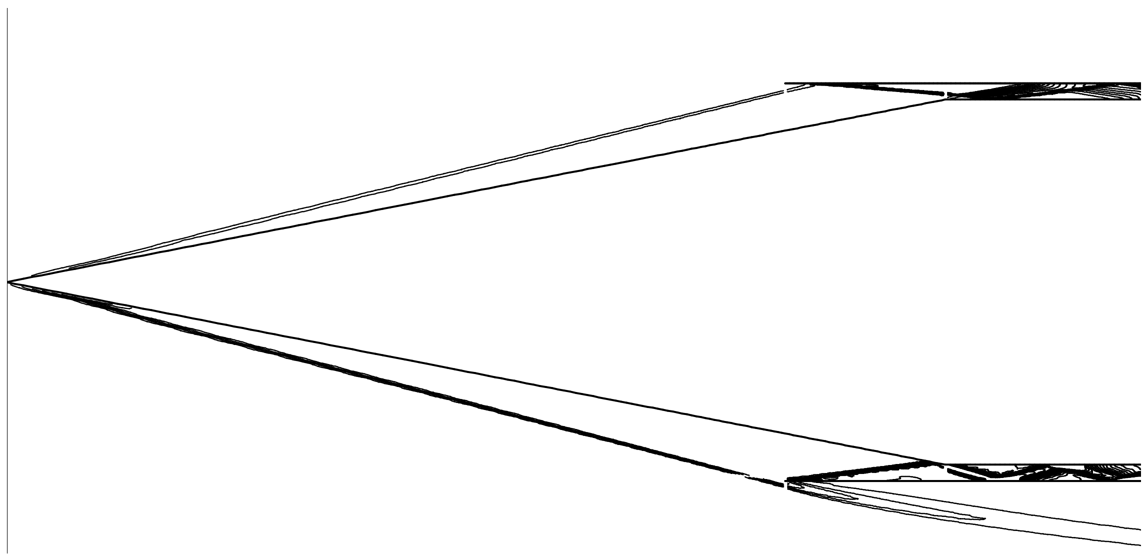


Figure 3.13: Pressure contours (in 5 kPa increments) from a computation fluid dynamics (CFD) simulation of the hypersonic inlet flow. The freestream conditions correspond to flight in a standard atmosphere at 4900 m/s and an altitude of 31.04 km, equating to a flight Mach number of 16.2. An angle of attack of 3° has been used.

It is also important for engine designers, as the local rise in pressure and heat transfer resulting from the condition where the bow shock generated from the inlet ramp strikes just inside the inlet, can cause structural problems. Further details regarding inlet design can be found in references [97, 98, 133].

Force and moment calculations for the inlet assume a constant pressure along the primary compression surface and the internal cowl section. A more detailed representation could be achieved by including the additional shock interaction associated with off-design operation. Again, the simple two-shock arrangement is considered sufficient to reasonably represent the inlet surface pressure and to capture the variation in combustor inlet conditions with flight condition. To simulate the flow arrangements of Figure 3.11, the inlet model makes use of the oblique shock relations and an expansion analysis for a calorically perfect gas. The theoretical foundations of these can be found in compressible flow texts, for example [134, 8]. Their solutions are summarized in the following sections.

Oblique Shock Analysis

A shock wave is generated whenever supersonic flow is turned onto itself. Provided the turning angle is less than a (Mach number dependent) maximum deflection angle, the shock will generally be an attached oblique shock wave. The notation used for the oblique shock procedure is shown in Figure 3.14. Using the θ-β-M relationship for oblique

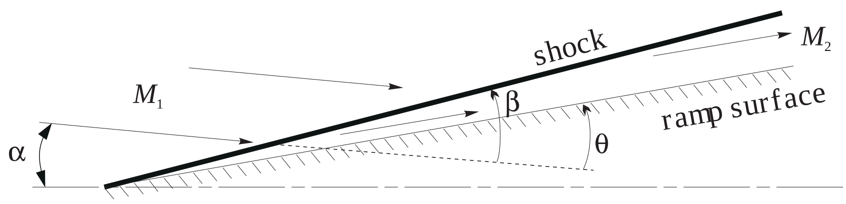


Figure 3.14: Oblique shock nomenclature.

shocks, the flow deflection angle θ is defined as a unique function of the upstream Mach number M_1 and the shock wave angle β,

$$\tan\theta = 2\cot\beta\left[\frac{M_1^2\sin^2\beta - 1}{M_1^2\left(\gamma + \cos 2\beta\right) + 2}\right]. \tag{3.5}$$

With θ and M_1 known, Equation 3.5 is solved iteratively to provide the shock angle β, relative to the freestream flow direction. The solution regime, as implemented, restricts the shock to the weaker of the two possible solutions available, providing supersonic downstream conditions. A strong shock solution would require some independent mechanism to increase the downstream pressure, such as choking of the combustor, and results in subsonic flow downstream of the shock. The bounds for the iterative solution of Equation 3.5

are provided by the high Mach number limit for small θ and β, and for the upper bound, a polynomial fit for the maximum shock angle achievable assuming a weak shock solution:

$$\begin{aligned}\beta_{\text{low}} &= \frac{\gamma+1}{2}\theta, \\ \beta_{\text{max}} &= \frac{1.5556}{M^3} - \frac{1.3844}{M^2} + \frac{0.0797}{M} + 1.1839.\end{aligned} \tag{3.6}$$

Having evaluated β, the conditions downstream are expressed by the oblique shock relations, where upstream and downstream conditions are denoted by subscripts 1 and 2, respectively.

Mach number normal to wave: $$M_{n2}^2 = \frac{M_{n_1}^2 + [2/(\gamma-1)]}{[2\gamma/(\gamma-1)]\,M_{n_1}^2 - 1} \tag{3.7}$$

Post shock Mach number: $$M_2 = \frac{M_{n_2}}{\sin(\beta-\theta)} \tag{3.8}$$

Pressure ratio: $$\frac{p_2}{p_1} = 1 + \frac{2\gamma}{\gamma+1}\left(M_1^2\sin^2\beta - 1\right) \tag{3.9}$$

Density ratio: $$\frac{\rho_2}{\rho 1} = \frac{(\gamma+1)\,M_1^2\sin^2\beta}{(\gamma-1)\,M_1^2\sin^2\beta + 2} \tag{3.10}$$

Temperature ratio: $$\frac{T_2}{T_1} = \frac{p_2}{p_1}\frac{\rho_1}{\rho_2} \tag{3.11}$$

An additional constraint on the model is associated with the maximum flow deflection angle θ_{max}, for a given Mach number. The form of the θ-β-M function in Equation 3.5 is such that if $\theta > \theta_{\text{max}}$, then no solution exists for a straight oblique shock wave. In that case a detached curved shock is required to process the flow and a more complex downstream flow field results. Using data drawn from Equation 3.5 a test for the existence of an oblique shock solution is expressed as a polynomial function, $\theta_{\text{max}} = f(1/M)$,

$$\theta_{\text{max}} = \frac{1.6137}{M^3} - \frac{2.418}{M^2} + \frac{0.0171}{M} + 0.7972. \tag{3.12}$$

An analytical expression for θ_{max} can also be derived by differentiating Equation 3.5 with respect to β, see [39]. Inlet design and vehicle operation for this study is such that departure from attached oblique shocks will only occur with operation well outside acceptable operating conditions.

Prandtl-Meyer Expansion Analysis

Prandtl-Meyer expansion describes the isentropic turning of a supersonic flow through an angle, for a calorically perfect gas. Expansion at a corner occurs through a *centered wave*,

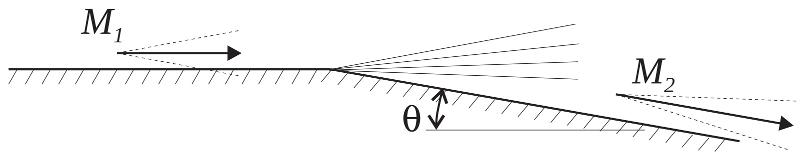


Figure 3.15: Nomenclature for expansion around a corner. The expansion fan angle is bounded by the upstream and downstream Mach angles, indicated by the dashed lines.

consisting of an infinite number of Mach waves which spread downstream, as sketched in Figure 3.15. The relationship between the expansion angle θ and the upstream and downstream Mach numbers is given by

$$\theta = \nu(M_2) - \nu(M_1), \tag{3.13}$$

where ν describes the Prandtl-Meyer function,

$$\nu(M) = \sqrt{\frac{\gamma+1}{\gamma-1}}\left[\tan^{-1}\sqrt{\frac{\gamma-1}{\gamma+1}(M^2-1)}\right] - \tan^{-1}\sqrt{M^2-1}. \tag{3.14}$$

Thus, knowing θ and M_1, $\nu(M_2)$ can be evaluated, and Equation 3.14 iteratively solved to provide the downstream Mach number M_2. To provide the limits for this numerical solution a set of polynomial functions expressing $M = f(\nu)$ were used, with ν defined in degrees. Two of these were provided by Fraser [78], accurate to four significant figures and covering the region $0 \leq \nu \leq 65°$ for $\gamma = 1.4$. For $0 \leq \nu \leq 5°$:

$$M = 1.0 + 7.932\times10^{-2}\nu^{2/3}(1 + 3.681\times10^{-2}\nu - 5.99\times10^{-3}\nu^2 + 5.719\times10^{-4}\nu^3), \tag{3.15}$$

and for $5° < \nu \leq 65°$:

$$\begin{aligned} M = {} & 1.071 + 3.968\times10^{-2}\nu - 4.615\times10^{-4}\nu^2 + 1.513\times10^{-5}\nu^3 \\ & -1.840\times10^{-7}\nu^4 + 1.186\times10^{-9}\nu^5 \end{aligned} \tag{3.16}$$

For Mach numbers greater than 4 (corresponding to $\nu > 65°$), another polynomial was assembled,

$$\frac{1}{M} = 0.6724 - 8.647\times10^{-3}\nu + 4.096\times10^{-5}\nu^2 - 1.088\times10^{-7}\nu^3. \tag{3.17}$$

The only downstream information needed for the external aerodynamics simulation is the Mach number and pressure. As the expanding flow is isentropic, the ratio of total and static pressure is

$$\frac{p_0}{p} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma-1}} . \tag{3.18}$$

So, for constant total pressure p_0, the pressure ratio across the expansion fan is expressed as

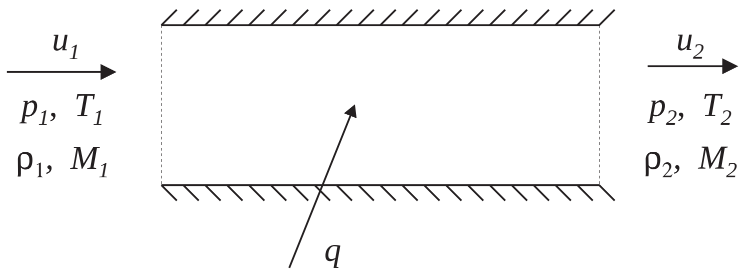
$$\frac{p_2}{p_1} = \left(\frac{2 + (\gamma - 1) M_1^2}{2 + (\gamma - 1) M_2^2}\right)^{\frac{\gamma}{\gamma-1}} . \tag{3.19}$$

The downstream surface pressure p_2 is assumed to be uniform over the entire surface, when applied to any external feature. Expansion within the thrust nozzle considers the wave reflections and is treated in more detail in Section 3.3.3.

3.3.2 Combustor Analysis

Achieving supersonic combustion in a scramjet engine must address a broad range of issues, including fuel injection, mixing, burning, chemical kinetics, shock interaction and boundary layer interaction. It continues to be the most researched feature of scramjet related technology [164, 50], however, even for the most advanced computational techniques, modelling the combustor flow processes is an extremely demanding task and, it is not without uncertainty [203]. Consequently, in pursuit of rapid analysis techniques, approaches of varying complexity have been used to provide general engine operating characteristics [219, 12, 39]. The basic processes of heat release in fuel-air combustion are fuel injection, fuel-air mixing, and chemical reaction. The HABV model avoids the details of the combustion process by describing combustion using a control volume approach to one-dimensional heat addition in a constant area duct. Heat is added directly to the flow in each scramjet duct without the addition of mass, assuming uniform flow across the ducts. The nomenclature for the analysis is shown in Figure 3.16, with subscripts 1 and 2 defining the upstream and downstream conditions respectively.

Without modelling the dynamic features of combustion, such as mixing rate and rate of combustion, the length of the combustor does not factor in the flow analysis. Also, by having the combustors parallel to the vehicle axis of symmetry, they do not contribute to the force and moment calculation. The purpose of the combustor model is therefore to evaluate the flow conditions (M, p, T, ρ), after combustion, thereby describing the flow presented to the nozzles. Despite the potential for high core flow temperatures in the combustor ($>$2000 K), the air is treated as a perfect gas. One of the limitations of the present modelling and flight trajectory chosen is that the simulated combustion temperatures exceed those generally desired for efficient engine operation [112].



Continuity: $\rho_1 u_1 = \rho_2 u_2$

Momentum : $p_1 + \rho_1 u_1^2 = p_2 + \rho_2 u_2^2$

Energy : $h_1 + \frac{u_1^2}{2} + q = h_2 + \frac{u_2^2}{2}$

Figure 3.16: One dimensional heat addition.

The fuel addition in the simulation model is controlled by the fuel to air equivalence ratio, ϕ. It defines the fuel/air mixing ratio f, in relation to the stoichiometric mixing ratio f_{st},

$$\phi = \frac{f}{f_{\text{st}}} \qquad \frac{\text{actual fuel/air}}{\text{stoichiometric fuel/air}} \tag{3.20}$$

Since the fuel addition is not actively controlled, a nominal equivalence ratio of one is maintained, representing a stoichiometric mix of fuel and air. Using ethane as a fuel, this occurs when hydrocarbon fuel molecules are mixed with just enough air such that all the hydrogen atoms form water vapour and all the carbon atoms form carbon dioxide. Such a combination usually results in the greatest liberation of sensible energy. The general stoichiometric equation for the combustion of hydrocarbon with air is as follows:

$$C_xH_y \; + \; \left(x + \frac{y}{4}\right)\left(O_2 + \frac{79}{21}N_2\right) \; \longrightarrow \; xCO_2 \; + \; \frac{y}{2}H_2O \; + \; \frac{79}{21}\left(x + \frac{y}{2}\right)N_2.$$

The stoichiometric mixing ratio on a mass basis is therefore given by the expression

$$f_{\text{st}} = \frac{36x + 3y}{103(4x + y)} \quad \frac{\text{kg Fuel}}{\text{kg Air}}. \tag{3.21}$$

For ethane C_2H_6, $f_{st} = 0.0624\frac{\text{kg Fuel}}{\text{kg Air}}$. So, given the equivalence ratio as a fuel input setting, the mass flow rate of fuel into the combustor can be evaluated.

$$\dot{m}_{\text{fuel}} = \phi f_{\text{st}} \dot{m}_{\text{air}} \tag{3.22}$$

The mass flow rate of air is defined relative to the upstream combustor conditions,

$$\dot{m}_{\text{air}} = \rho_1 A_{\text{C}} U_1,$$

where the combustor area A_{C} is equivalent to the height of the combustor for the simplified axisymmetric scramjet, and $U_1 = M_1\sqrt{\gamma R T_1}$, is the combustor upstream velocity.

Knowing the amount of fuel added to the flow, the conditions downstream are obtained by the application of the governing equations of continuity, momentum, and energy, to a control volume [8], see Figure 3.16. The amount of heat added per kilogram of air, q, is proportional to the fuel to air mass flow ratio and the heating value H (J/kg-fuel), of the fuel,

$$q = \eta_c H \frac{\dot{m}_{fuel}}{\dot{m}_{air}}. \tag{3.23}$$

where η_c represents the combustor efficiency, discussed further in Section 3.5.1. Applying the energy equation for a calorically perfect gas shows the heat addition q to directly change the total temperature of the flow,

$$q = c_p(T_{02} - T_{01}), \tag{3.24}$$

where c_p is the constant pressure specific heat and $T_{02} - T_{01}$ the increase in total temperature. The ratio of properties across the control volume are derived from the momentum and continuity expressions and the perfect gas equation of state.

$$\frac{p_2}{p_1} = \frac{1+\gamma M_1^2}{1+\gamma M_2^2} \tag{3.25}$$

$$\frac{T_2}{T_1} = \left(\frac{1+\gamma M_1^2}{1+\gamma M_2^2}\right)^2 \left(\frac{M_2}{M_1}\right)^2 \tag{3.26}$$

To find the downstream Mach number the isentropic flow relation describing the ratio of total and static temperatures is employed.

$$\frac{T_0}{T} = 1 + \frac{\gamma - 1}{2} M^2 \tag{3.27}$$

Combining Equations 3.26 and 3.27 provides a relationship for the ratio of total temperature,

$$\frac{T_{02}}{T_{01}} = f\left(M_1, M_2\right). \tag{3.28}$$

Equation 3.28 is iteratively solved to give the downstream Mach number, M_2. Heat ad-

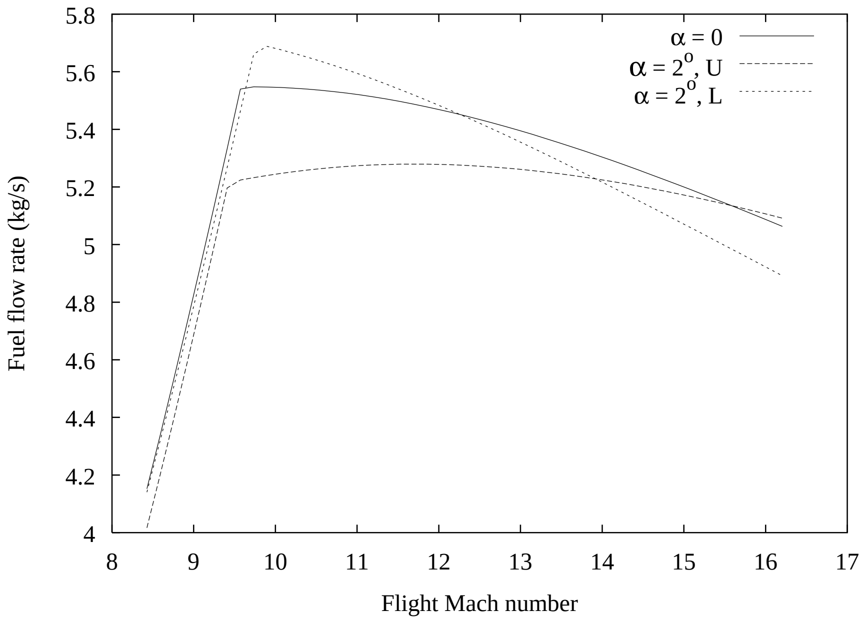


Figure 3.17: Fuel input during flight along the nominal trajectory for the upper (U) and lower (L) engine modules.

dition drives the Mach number towards 1, so with the assumption of supersonic flow throughout the combustor, the downstream Mach number is subject to the constraint $1 < M_2 < M_1$. Choking of the flow occurs when enough heat is added for the flow to become sonic. To prevent this occurrence, the fuel mass flow rate necessary for choking is monitored to provide an adaptive limit for the maximum heat addition. The choking limit is set conservatively by assuming 100% combustion efficiency. If the fuel setting breaches this limit the fuel mass flow rate is adjusted to 90% of the choke limit. Following the default flight trajectory discussed in Section 2.2, the fuel rate requires adjustment up to a flight Mach number of around Mach 10. Figure 3.17 compares the fuel input for the nominal condition $\alpha = 0$, to the operation of the upper and lower engine modules at angle of attack $\alpha = 2°$. The variation against flight Mach number reflects the changing air flow rate through the combustor, with a fixed fuel equivalence ratio setting of $\phi = 1$.

Active control of the fuel addition has not been used. Typically, fuel control would be warranted for trajectory maintenance and stability augmentation through differentially throttling the engine modules. However, the vehicle functions as an accelerator, and considering the marginal acceleration capabilities the simple geometry achieves, it was considered desirable to run the engines at their maximum settings. The lack of demand on the accelerating capability is reflected in the trajectory reference which correlates the flight velocity against altitude rather than time. In addition, the geometry of the nozzle does not allow effective use of differential throttling as a means of attitude control. The potential stabilizing moment generated by the nozzle thrust surface is diminished by the normal force acting on the cowl section.

3.3.3 Nozzle Analysis

Flow exiting the combustor enters the nozzle by expanding around a corner, increasing the Mach number downstream and lowering the pressure. Conventional wisdom includes a cowl geometry which extends far enough downstream to capture the expansion fan. The geometry used here simplifies the downstream interactions by confining the flow structure further, with an extended cowl section. To map the pressure profile along the cowl and thrust surfaces, a two-dimensional wave interaction model has been constructed. It is roughly equivalent in application to the characteristics method, which has been used in other studies to determine the thrust production in two dimensional scramjet [4]. Consistent with the simulation of the other scramjet elements, this model assumes a perfect gas and neglects viscous effects.

Figure 3.18 summarizes the features of the two-dimensional nozzle analysis. It shows the construction of a supersonic flow pattern using weak finite waves. An expansion fan originating from the corner, propagates across the air stream. What follows is a series of interaction zones as the fan reflects of the cowl surface and then the thrust surface. For low upstream Mach numbers this process may repeat itself within the length of the nozzle. With increasing flight speed the nozzle upstream Mach number increases, pushing the initial fan further downstream.

The expansion fan is represented by equal strength, weak, finite waves, where the wave strength δ represents the absolute flow deflection produced by each wave, or the total expansion angle divided by the number of waves. Individual waves are defined as either left or right running, relative to the upstream flow direction. Cells with uniform properties divide the flow and are referenced by the coordinates (m, n), see Figure 3.18, representing the number of right running waves m, and the number of left running waves n, crossed to arrive at the location. Governing the expanding flow are the flow deflection θ, and the Prandtl-Meyer function $\nu(M)$, previously expressed by Equations 3.13 and 3.14. Applying the argument that the strength of a weak wave is not affected by intersection with other waves [134], the flow properties (θ, ν) within each cell are expressed through the number of left and right running waves crossed to reach the cell.

$$\theta(m, n) = \theta_1 + \delta\,(m - n) \tag{3.29}$$

$$\nu(m, n) = \nu_1 + \delta\,(m + n) \tag{3.30}$$

The upstream conditions for this case are $\theta = 0$, and $\nu_1 = \nu(M_1)$ using Equation 3.14. Equations 3.29 and 3.30 then allow a complete mapping of the flow condition throughout the flow structure. The Mach number of the flow within each cell is evaluated using the polynomial functions described by Equations 3.15-3.17. Since the total pressure remains

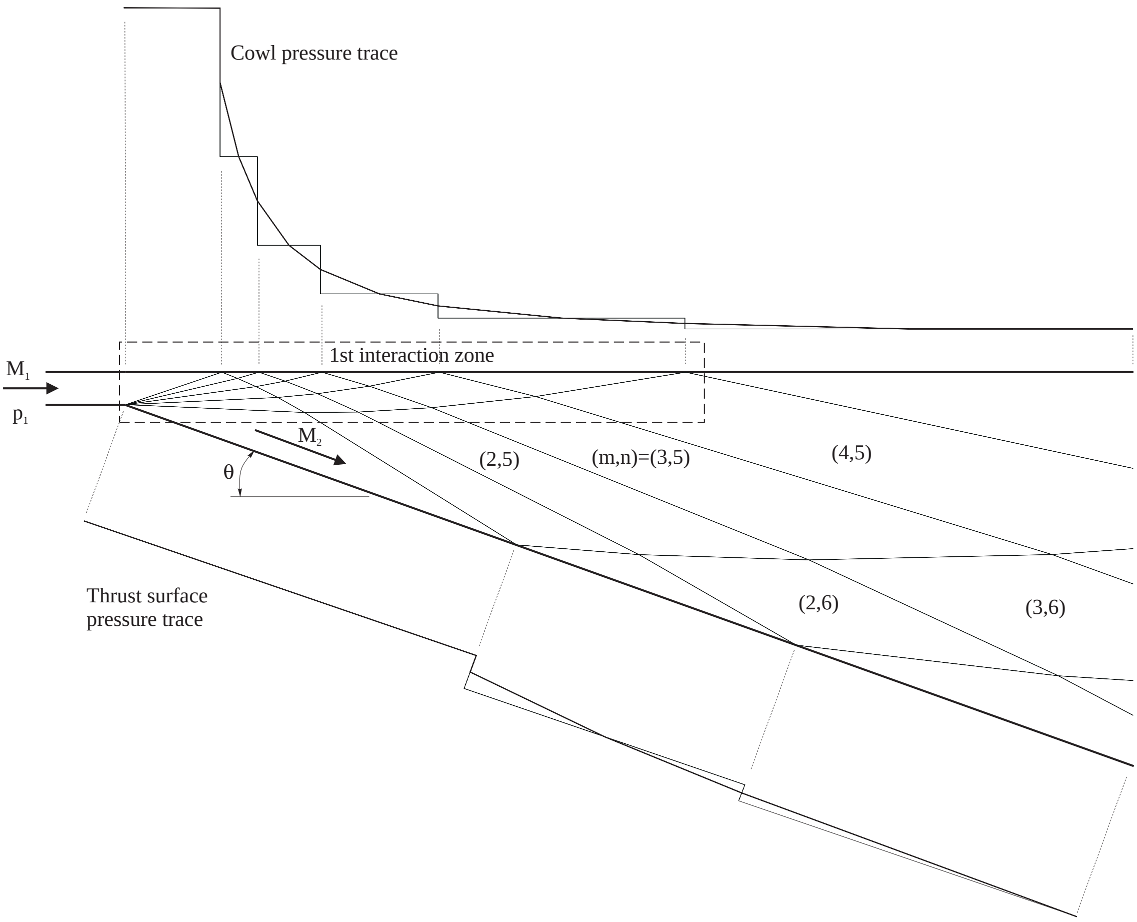


Figure 3.18: Wave interaction model for the expansion fan, showing a reduced number of waves to simplify the picture. Adjacent to each nozzle surface is an internal pressure profile showing the step changes in pressure coincident with wave reflection, and the transitions used to generate forces and moments. Also shown is the method of indexing regions within the expansion fan, using the number of right (m) and left (n) running waves crossed to reach the region.

constant through an isentropic expansion, the pressure for each cell is evaluated using the isentropic flow relation for the ratio of total and static pressure, see Equation 3.18.

Knowing the distribution of flow properties within the expansion, the surface pressure profiles are simply obtained by following the expansion fan geometry, starting at the corner. Each wave is orientated by averaging the local Mach lines in adjacent cells. For example, the angle ψ_r of the right running wave in between cells (2,5) and (3,5) is expressed as,

$$\psi_r(2:3,5) = \frac{1}{2}\left[(\theta-\mu)_{2,5} + (\theta-\mu)_{3,5}\right],$$

where $\mu_{m,n} = \sin^{-1}\frac{1}{M_{m,n}}$. By describing the wave angles in this manner, perfect reflection of the waves is not guaranteed. The quality of the interaction geometry improves with an increase in the number of waves used to subdivide the expansion fan.

Cells adjacent to the cowl and thrust surfaces provide the data necessary to build their respective pressure profiles. As the flow properties are assumed uniform within each cell, the step profiles shown in Figure 3.18 result. To smooth out the profile, linear transitions between the steps are used, which for a large enough number of waves provides an adequate approximation to a continuous expansion. The net axial and normal forces are formed by a summation along the surface S, where the surface normal relative to vehicle reference frame is $(n_x, 0, n_z)$.

$$F_x = \sum_{i=1}^{N-1} n_x \frac{P(i)+P(i+1)}{2}\Delta S_i \tag{3.31}$$

$$F_z = \sum_{i=1}^{N-1} n_z \frac{P(i)+P(i+1)}{2}\Delta S_i \tag{3.32}$$

The net pitching moment is likewise expressed as a summation, separated into components generated by the axial and normal forces for each surface segment.

$$M_y = \sum_{i=1}^{N-1}\left(\mathrm{sgn}(n_x)\int zP(z)\mathrm{d}z - \mathrm{sgn}(n_z)\int xP(x)\mathrm{d}x\right) \tag{3.33}$$

The integrals in Equation 3.33 are evaluated over each linear segment of the pressure profile. They are expressed in terms of their bounding points, referenced by the indices $(i, i+1)$:

$$\int xP(x)\mathrm{d}x = \frac{|\Delta x|}{6}\left(P_i(2x_i + x_{i+1}) + P_{i+1}(2x_{i+1} + x_i)\right) . \tag{3.34}$$

The expression for $\int zP(z)\mathrm{d}z$ is equivalent to Equation 3.34.

Figure 3.19 shows the convergence of the nozzle and cowl surface pressure profiles with an increasing number of waves within the expansion fan. A mid-trajectory flight condition has been used with $M_\infty = 11.7$ and $h = 26682\,\mathrm{m}$. The sudden changes in trends below a flight Mach number of 10 reflect the clipping of the fuel added to the combustor, to avoid conditions which could choke the flow. The nozzle calculation can represent a majority of the vehicle simulation time, so in cases such as designing the flight controller, the 10 wave model is considered sufficient. For the example shown in Figure 3.19 the error in the 10 wave approximation relative to the 20 wave case, is 1.1 % for F_x and 1.7 % for M_y.

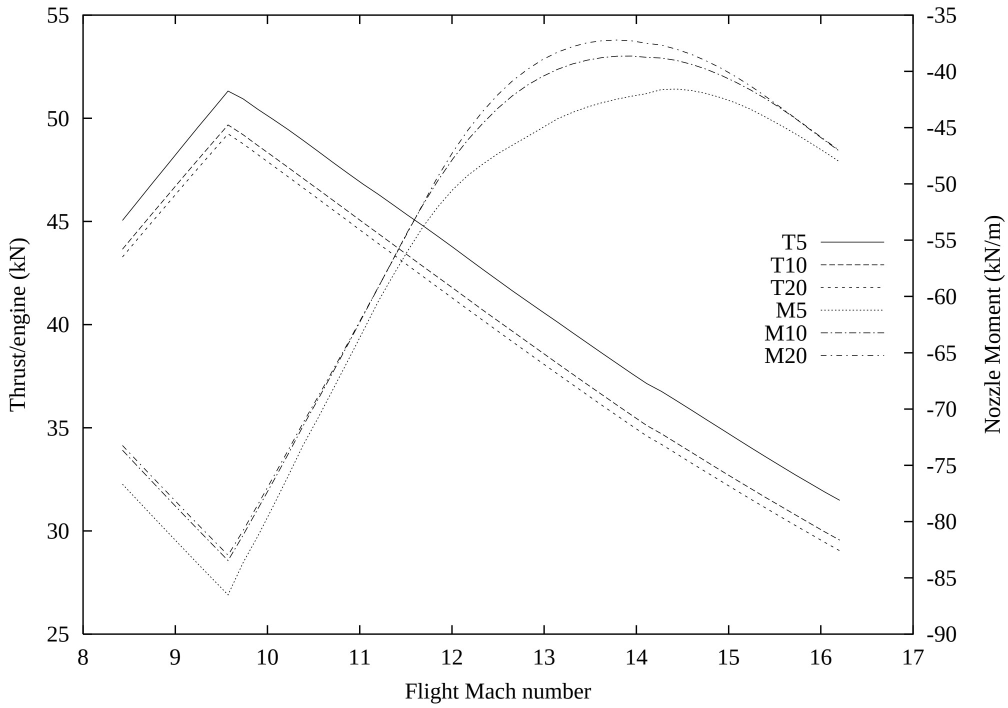


Figure 3.19: Thrust (T) and moment (M) evaluation for various expansion fan resolutions. The indexing (5,10,20) refers to the number of waves used to describe the expansion fan.

The effect of flight speed on the expansion fan structure is shown in Figure 3.20. As the vehicle accelerates the nozzle upstream Mach number increases, increasing the strength of the expansion, and spreading the interaction zones downstream. It was previously stated that the nozzle configuration is not conducive to generating stabilizing moments. Figure 3.21 shows the combined nozzle moments generated by a fixed fuel equivalence ratio setting and the difference in airflow through the modules with non-zero angles of attack. Up to the point where the net nozzle moment changes sign, the contributions from the internal cowl surface dominate the nozzle moment. So, despite the greater fuel and air flow rates in the lower engine module, the nozzles generate a destabilizing moment. A stabilizing effect is only produced when, with the increase in fuel addition to the upper module, the dominance of the cowl surfaces is reduced and the net nozzle moment is driven the thrust surfaces.

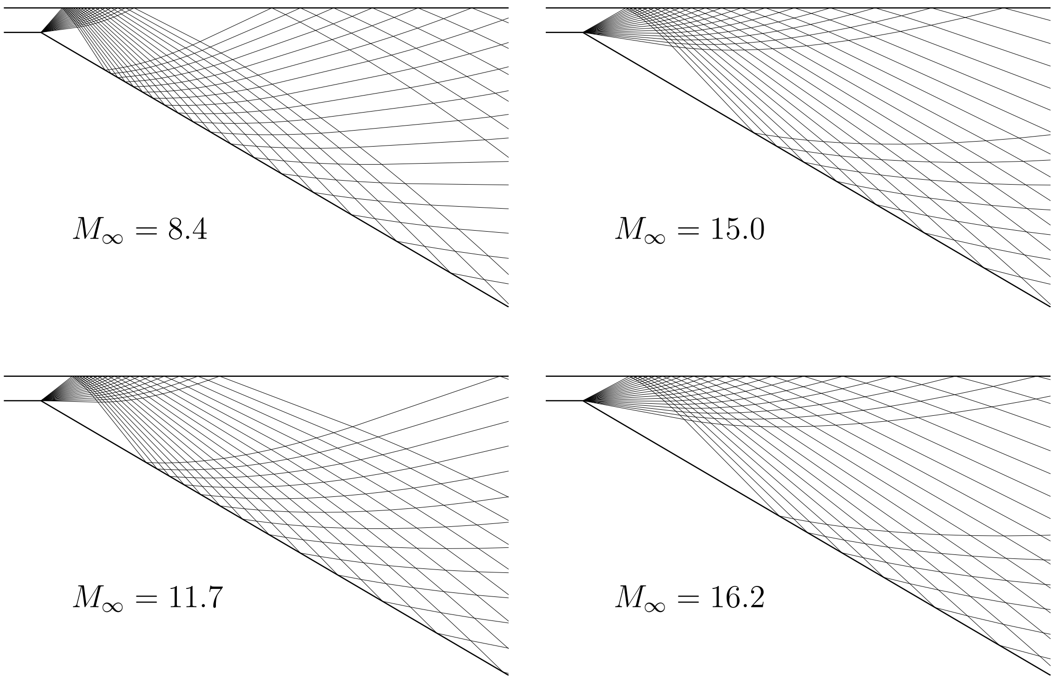


Figure 3.20: Nozzle flow variation with flight Mach number.

To check the implementation of the expansion fan interaction method, a series of CFD simulations were performed for the two-dimensional nozzle geometry, using a Navier-Stokes code [110, 111]. A Mach 15 flight condition was examined with the vehicle having a zero and a small non-zero angle of attack of $2°$. The axial and normal forces acting on the thrust surface were all within 0.6 % of the CFD values. For the normal force generated along the internal cowl surface, a 2 % difference was observed.

3.3.4 Lifting Wing and Elevator Analysis

Along with the cowl, the wings and elevator provide the only external aerodynamic analysis required for the vehicle simulation. Aerodynamic modelling of these components considers the surfaces as two-dimensional wedges, and applies an oblique shock or a Prandtl-Meyer expansion analysis [134], depending on the local flow turning angle. For the lifting wing, the flow structure is simply dependent on the vehicle angle of attack. In between the lifting wing and rear wing arrangement the flow is assumed to return to freestream conditions. Force and moment calculations for the all-moving rear wing use surface normals and area centres calculated from the actuation angle and the wing geometry. All surfaces are treated as having uniform pressure.

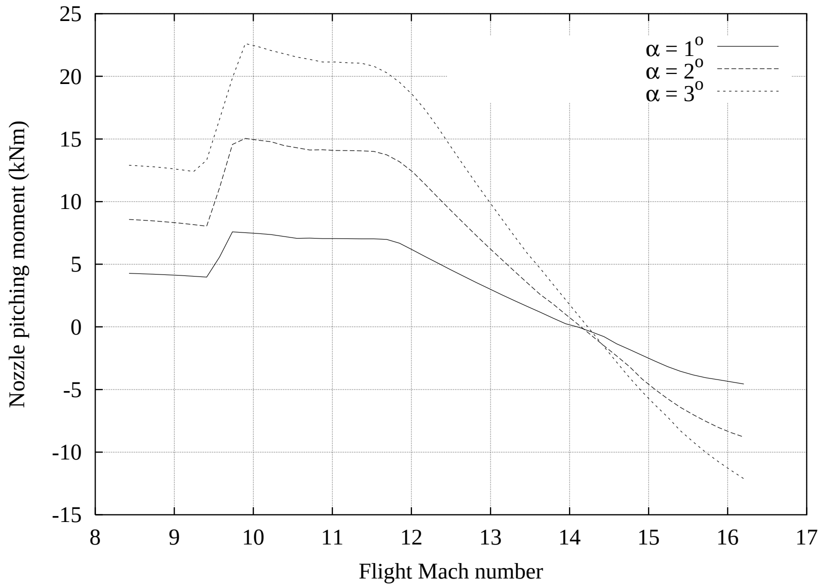


Figure 3.21: Stabilizing capability of the engine nozzles without active control of the fuel addition. With a fixed the fuel/air equivalence ratio, a differential fuel flow rate supply to the two engine modules is due to the differential air flow rate through the modules.

3.4 Vehicle Performance

Figure 3.22 shows the broad range performance of the scramjet vehicle in terms of the net thrust and specific impulse. There are several features worth noting. Firstly, the initial rise in thrust is due to the lessening threat of choking the combustor with the nominal fuel input, as the flight Mach number increases. Secondly, there is a substantial performance penalty for operating at a non-zero angle of attack. This makes the design of the vehicle geometry difficult, as a fixed geometry vehicle is unable to match the lifting requirements for the entire trajectory, and angle of attack perturbations will be required to track the desired trajectory. The final remark on Figure 3.22 refers to the relatively low specific impulse, which declines with increasing flight speed. Performance estimates for hydrocarbon fueled scramjets are generally provided for the low hypersonic flight conditions, see Figure 1.1, reflecting the limited Mach number range for which they are expected to be useful. The specific impulse measured for this scramjet is of the same order as that of modern rockets.

3.5 Performance Uncertainty

The hypersonic air-breathing vehicle model (HABV) presented so far represents a nominal performance assessment of the vehicle's aerodynamics, propulsion, structural components. It provides a deterministic, instantaneous description of the net forces and moments acting on the vehicle, as a function of the freestream conditions, the vehicle attitude, and

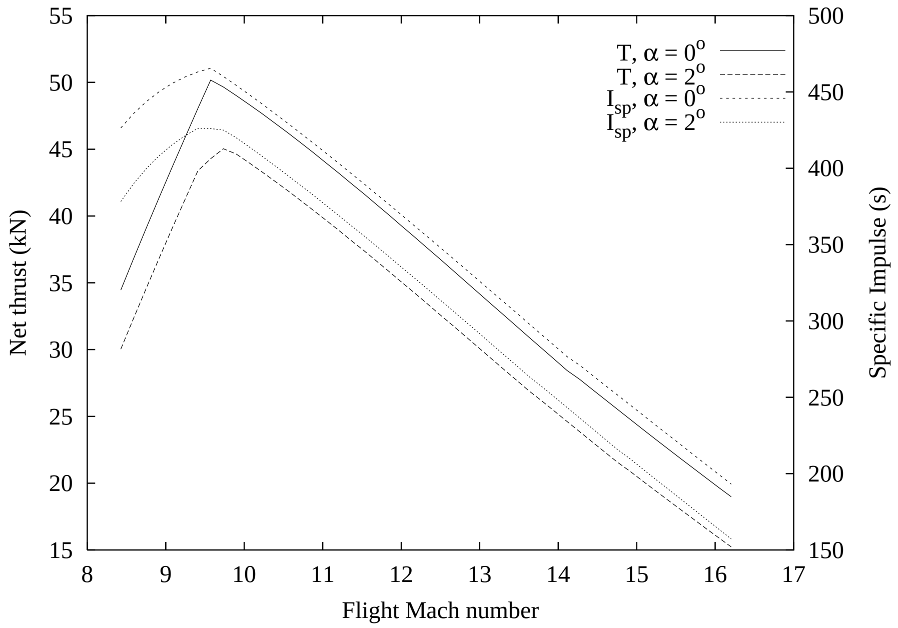


Figure 3.22: Net thrust and specific impulse performance of the engine along the default flight trajectory.

the control settings, applicable for the hypersonic flight trajectory. Unmodelled features and time dependent flow processes generate uncertainty in the vehicle performance.

It is important to examine the robustness of the flight controller in the presence of vehicle uncertainty. For this purpose, parametric uncertainty has been used to describe stochastic perturbations in the engine performance, control effectiveness, and the physical properties of the vehicle. Combustion efficiency, elevator surface pressure, and fuel centre of mass have been use to represent general performance variations. Each is implemented in the flight simulation as an uncertainty filter, based on a Nyquist frequency of 50 Hz. The low-pass filters are coded as difference equations with white noise of unit variance, $W_{0,1}$, providing the source signal. Aerodynamic and propulsive uncertainties are assumed to be driven by atmospheric turbulence. Uncertainty in the freestream conditions is discussed a little later in Section 3.6.

3.5.1 Combustor Efficiency

Engine flow processing uncertainty has be lumped into one parameter, using the efficiency of the combustion process. In the model described in Section 3.3.2, the efficiency represents a fraction of the available heat release of the fuel. A nominal combustor efficiency $\eta_{\text{nom}} = 0.75$, allows a reasonable thrust generation, though generating combustion temperatures 3000 - 4000 K, which exceed generally desirable values. This is perhaps an indication that the flight dynamic pressure is to high.

To evaluate the combustion efficiency variation, a randomized additive perturbation

$\Delta\eta$ is applied about the nominal efficiency. Using a low-pass first-order filter, the cut-off frequency is set to provide the same frequency content as the longitudinal turbulence model, see Section 3.6.2.

$$\begin{aligned} \eta_c &= \eta_{\text{nom}} + \Delta\eta \\ \Delta\eta &= 2.3525\times 10^{-3}(W_{0,1}[n] + W_{0,1}[n-1] + 0.99686\Delta\eta[n-1] \end{aligned} \tag{3.35}$$

The Nyquist frequency based on the nominal integration timestep, is significantly less than that associated with the transit time of flow through the combustor, which is roughly 0.0004 s. The first order filter equation above allows the possibility of low frequency engine surges and some higher frequency variations, with a maximum perturbation of $\pm 15\,\%$. Each combustor module is considered independently.

3.5.2 Elevator Surface Pressure

Control effectiveness has been represented by uncertainty in the elevator surface pressure, equivalent to $\pm 5\,\%$ variation about the nominal value. Again, with reference to the turbulence filter properties described in Section 3.6.2 the following filter equation is used,

$$\begin{aligned} P &= (1 + \Delta p)P_{\text{nom}} \\ \Delta P &= 7.8417\times 10^{-4}(W_{0,1}[n] + W_{0,1}[n-1]) + 0.99686\Delta P[n-1] \end{aligned} \tag{3.36}$$

Separate signal histories are kept for the upper and low surfaces of the elevator.

3.5.3 Fuel Centre of Mass

In addition to the variation in mass properties due to fuel consumption, the location of the center of mass of the fuel is allowed to fluctuate by ± 0.25 m. Since high frequency oscillations are unlikely to be present in the fuel sloshing behaviour, a second filter with a cut-off frequency of 2 Hz has been used.

$$\begin{aligned} cm_f &= cm_{\text{nom}} + \Delta cm \\ \Delta cm &= 2.1472\times 10^{-4}(W_{0,1}[n] + W_{0,1}[n-2]) + 0.43445 W_{0,1}[n-1] \\ &\quad +1.9556\Delta cm[n-1] - 0.95654\Delta cm[n-2] \end{aligned} \tag{3.37}$$

3.6 Environment Model

The environment model encompasses the description of the physical properties of the Earth and its atmosphere. Table 3.2 summarizes the defining physical parameters of the

Table 3.2: General definition of the simulation environment.

	Description	Value...
Earth	Radius, R_E.	6738.4 km
	Gravity at sea level, g_0.	9.81 m/s^2
	Rotation, ω^E.	7.29246×10^{-5} rad/s
Atmosphere	Sea level temperature.	288.15 K
	Sea level pressure, (1 atm).	101.325 kPa
	Sea level density.	1.225 kg/m^3
Gas properties	Ideal gas constant, R.	287 J/kgK
	Specific heat ratio, γ.	1.4

simulation environment. Altitude has been include in the system state to account for the atmospheric and gravity gradients. The local gravity g variation with altitude is simply described using the absolute altitude $h_a = h_G + R_E$, where h_G is the geometric height above the surface of the Earth whose radius is R_E.

$$g = g_0 \left(\frac{R_E}{h_a} \right)^2 \tag{3.38}$$

Atmospheric modelling describes the variation in temperature, pressure, and density with altitude. Relative to the rotating Earth, the atmosphere is assumed stationary. Disturbances in the velocity field, due to turbulence or wind, are applied uniformly to the vehicle. For a more detailed representation of the atmosphere, a complete inertial based atmosphere model could be applied [68].

3.6.1 Atmospheric Modelling

Natural variations in atmospheric properties exist as functions of altitude, longitude, latitude, time of day, season, and solar activities. As it is generally impractical to simulate these variations, a standard atmosphere is used to provide mean values of pressure, temperature, density, and other properties, as a function of altitude. Central to these models is a defined variation of temperature with altitude. Figure 3.23 shows the temperature profile for the U.S. Standard Atmosphere, 1976 [1]. It defines the temperature regions as being either isothermal or of constant gradient, up to a geometric altitude h_G, of 86 km. The geopotential altitude h, is used as a reference for temperature, simplifying the mathematics for defining pressure, by accounting for the variation of gravity with altitude [6]. The conversion between geometric and geopotential altitude is made with the following expression, where R_E represents the radius of Earth at the equator.

$$h = \left(\frac{R_E}{R_E + h_G} \right) h_G \tag{3.39}$$

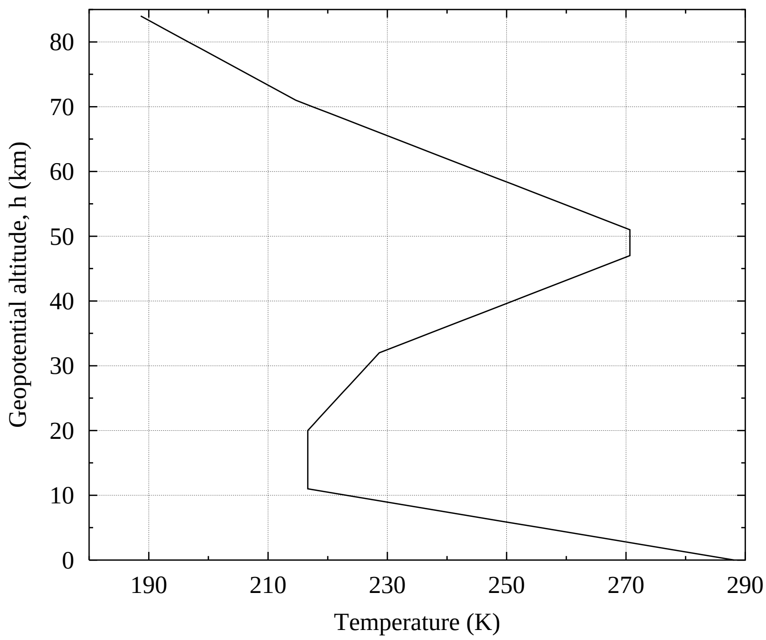


Figure 3.23: Temperature profile for U.S. Standard Atmosphere, 1976, up to $h_G = 86\,\text{km}$. See Table 3.3 for the supporting data.

Table 3.3: Standard atmosphere data, for the 7 fundamental layers up to $h_G = 86\,\text{km}$, of U.S. Standard Atmosphere, 1976. Subscripts 1 and 2 respectively refer to the lower and upper boundary of each layer.

h_1 (km)	h_2 (km)	T_1 (K)	$\mathrm{d}T/\mathrm{d}h$ (K/km)	p_1 (Pa)
0	11	288.15	-6.5	101.325×10^3
11	20	216.65	0.0	22.632×10^3
20	32	216.65	1.0	5474.9
32	47	228.65	2.8	868.02
47	51	270.65	0.0	110.91
51	71	270.65	-2.8	66.939
71	84.852	118.65	-2.0	3.9564

To predict the pressure and density given the temperature profile in Figure 3.23, the hydrostatic equation is used to construct a force balance on an element of fluid. Using the definition for altitude given by 3.39, the force balance gives the change in pressure $\mathrm{d}p$

across an element of fluid of height dh,

$$\mathrm{d}p = -\rho g_0 \,\mathrm{d}h, \tag{3.40}$$

where ρ is the density at altitude h and g_0 is the acceleration due to gravity at the surface of the Earth. Integration of Equation 3.40 over the isothermal and gradient temperature regions then provides the general expressions for atmospheric pressure for a perfect gas, where the subscript 1 refers to properties at the low altitude end of the relevant temperature region and the temperature gradient $a = \mathrm{d}T/\mathrm{d}h$ is in K/m:

$$\text{Isothermal regions:} \quad \frac{p}{p_1} = e^{-(g_0/RT)(h-h_1)} \tag{3.41}$$

$$\text{Gradient regions:} \quad \frac{p}{p_1} = \left(\frac{T}{T_1}\right)^{-g_0/aR} \tag{3.42}$$

$$\text{Perfect gas equation of state:} \quad p = \rho RT \tag{3.43}$$

The data to apply these equations is provided in Tables 3.2 and 3.3.

Uncertainty in the nominal atmospheric description is included by the addition of artificial noise to the atmosphere. Temperature variation is driven by turbulence, so the uncertainty is based on duplicating the frequency content of the turbulence functions defined in Section 3.6.2. Using the same format as the parametric uncertainty functions, the temperature variation ΔT is evaluated using difference equation to define a first order filter sourced with white noise of unit variance.

$$\begin{aligned} T &= (1+\Delta T)T_{\text{nom}} \\ \Delta T &= 3.921\times 10^{-4}(W_{0,1}[n] + W_{0,1}[n-1]) + 0.99686\Delta T[n-1] \end{aligned} \tag{3.44}$$

The intensity of the white noise input was chosen to provide coefficients that generated temperature variations up to $\pm 2\,\%$. Pressure is assumed to follow the standard atmosphere model and the density is provided by the perfect gas law.

3.6.2 Atmospheric Turbulence Model

For engineering purposes, the conventional approach to turbulence modelling as a stochastic process uses the Dryden spectra [150]. The Dryden spectra can be implemented as filters through which white noise of unit variance is passed. In a digital simulation the frequency content of the artificially generated noise is truncated by the Nyquist frequency, $\pi/\Delta t$ rad/s. The Nyquist frequency describes the maximum frequency which can be generated by a sampling time Δt. To account for the band-limited noise, the intensity of the input noise sequence is set to the inverse of the Nyquist frequency. The discrete time domain transfer functions for the longitudinal (F_u) and vertical (F_w) turbulence filters are therefore defined as follows:

$$F_u(s) = \sqrt{\frac{\pi}{\Delta t}}\sigma_u\sqrt{\frac{2V}{\pi L_u}}\left(\frac{1}{s+\frac{V}{L_u}}\right) \tag{3.45}$$

$$F_w(s) = \sqrt{\frac{\pi}{\Delta t}}\sigma_w\sqrt{\frac{3V}{\pi L_w}}\left(\frac{s+\frac{V}{\sqrt{3}L_{v,w}}}{\left(s+\frac{V}{L_w}\right)^2}\right) \tag{3.46}$$

Both filters are parameterized by the standard deviation of the turbulence, σ, and an integral scale length, L. Also featured in the above equations is the air relative vehicle velocity, V, excluding turbulence. The length scale determines the power distribution over the frequency range, while σ changes the power level without changing the relative distribution.

Since the numerical flight simulation is discretized by the integration time step, the longitudinal and vertical turbulence filters are implemented as difference equations. For a digital filter the output $y(k)$ at the kth sampling is defined in terms of the input $x(k)$ and the filter input/output history.

$$y[k] = \sum_{i=0}^{N} b_i x[k-i] - \sum_{i=1}^{M} a_i y[k-i] \tag{3.47}$$

In much the same as the s domain is used for continuous time systems, the z-domain is applied to discrete-time simulation. The s and z domains are related by

$$z = e^{sT}, \tag{3.48}$$

where T is the sampling period. To transfer the s-domain transfer function to the z-domain discrete equivalent, a simple conformal mapping between the two domains is provided by the bilinear transform, also known as Tustin's approximation to 3.48 [175, 233],

$$s \leftarrow \frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})}. \tag{3.49}$$

The bilinear transform is based on the Taylor series approximation of e^{Ts}. Following the substitution of 3.49 into the s-domain transfer functions, the resulting z-transform, H(z), can be written as a polynomial fraction.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{N} b_i z^{-i}}{\sum_{i=0}^{M} a_i z^{-i}} \tag{3.50}$$

From the definition of the z-transform of a number sequence, $z_{-i}Y(z)$ can be equated with $y[k-i]$. The difference equation of 3.47 can then simply be expressed by allowing $a_0 = 1$, and matching the coefficients with those in the z transfer function.

In the flight simulation code, atmospheric turbulence has been implemented using a single reference flight condition, thereby avoiding any consistency issues with switching between turbulence parameters. A mid-trajectory reference altitude of 25 km was used to source the filter parameters, using the severe turbulence data in [150]:

$$\begin{aligned} \sigma_u &= 4.34\,\text{m/s} & \sigma_w &= 3.34\,\text{m/s} \\ L_u &= 12000\,\text{m} & L_w &= 6560\,\text{m} \end{aligned}$$

The filters are discretized according to a nominal integration timestep $T = 0.01$ s. With unit variance white noise input data expressed as W, the difference equation for the longitudinal and vertical turbulence velocities are defined as follows:

$$\Delta u[n] = 0.1655\,(W[n] + W[n-1]) + 0.9971\Delta u[n-1] \tag{3.51}$$

$$\begin{aligned} \Delta w[n] = {} & 0.2105W[n] + 6.474\times 10^{-4}W[n-1] - 0.2098W[n-2] \\ & +1.98936\Delta w[n-1] - 0.98939\Delta w[n-2] \end{aligned} \tag{3.52}$$

White noise refers to a random process with a characteristic Gaussian distribution.

A sample turbulence velocity history is shown in Figure 3.24. For this example the standard deviations of Δu and Δw are 3.065 m/s and 3.42 m/s respectively. Given a long enough sequence, the simulated standard deviations approximately match the parameter values used to define the filters, as required.

3.7 Flight Dynamics

A flight simulator requires the coupling of a mathematical description of the vehicle's performance, with the dynamical and kinematical equations describing aircraft flight. The first half of this chapter presented the basis of the vehicle performance model, featuring a numerical aero-propulsive simulation and a description of the vehicle's physical prop-

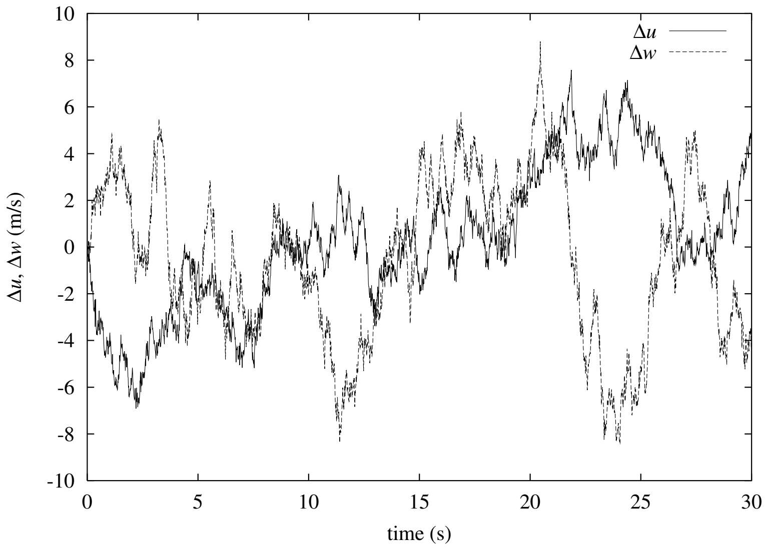


Figure 3.24: Simulated history of longitudinal and vertical turbulence velocities.

erties. Through the following sections, the kinematics and dynamics will be developed for the hypersonic flight of a rigid body aircraft. To capture features relevant to launching into orbit, a general six degree-of-freedom dynamic model is derived from the force and moment equations, for high speed flight about a spherical, rotating Earth [68]. The centre of the Earth is assumed to be fixed in inertial space with its atmosphere at rest relative to its surface.

3.7.1 Coordinate Reference Frames

Newton's law of motion $\boldsymbol{F} = m\boldsymbol{a}$ is defined relative to an inertial frame of reference. For convenience however, the equations of motion for atmospheric flight are generally written in terms of a non-inertial frame fixed to the aircraft. The mapping between the two frames is through a series of coordinate transformations which are based on describing the rotation of the circular Earth, and the vehicle attitude relative to a known reference. Figure 3.25 shows the reference frames used for the hypersonic flight simulator, following a conventional arrangement for aircraft simulators [68].

The Earth is assumed fixed in inertial space. Two Earth-fixed frames, F_{EC} and F_E, are used. The Earth-centre frame F_{EC} has its origin at the centre of the Earth, such that the Earth's rotation is given by an angular velocity ω^E about axis $O_{EC}z_{EC}$. Axes directions are further set by reference points on the Earth's axis and the equator - zero latitude (λ) and zero longitude (μ) for x_{EC} is used here. Earth-fixed surface frame F_E is also located by latitude and longitude (λ_E, μ_E), and arranged with $O_E z_E$ directed vertically down, while $O_E x_E$ and $O_E y_E$ are directed north and east respectively. It provides a reference point on Earth for the motion of the aircraft.

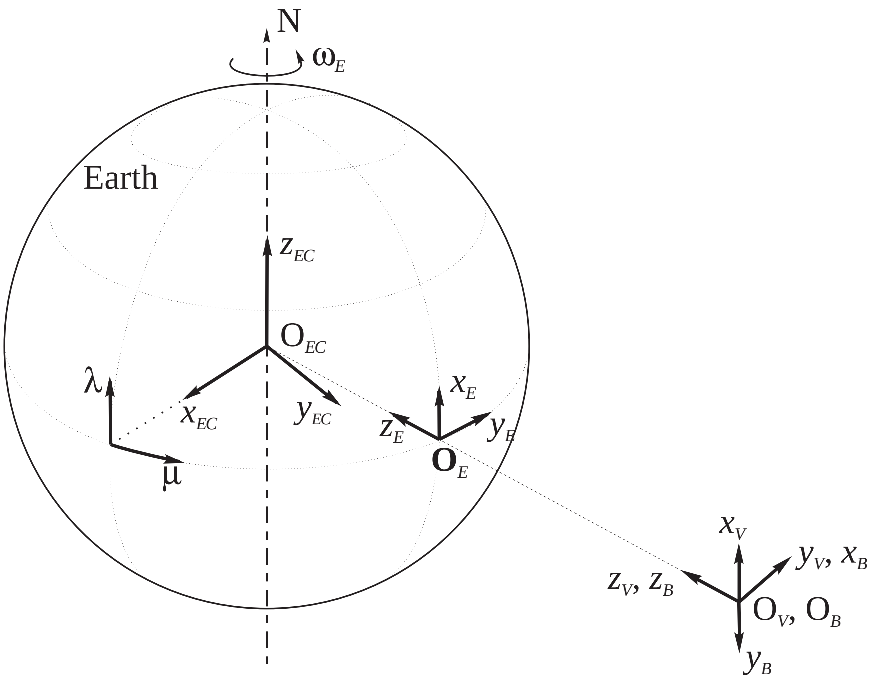


Figure 3.25: Reference frames used for hypersonic flight dynamics simulator.

Two vehicle based frames, F_V and F_B, are used. F_V, a vehicle carried vertical frame, accounts for the curvature of the Earth. Its origin O_V is attached to the vehicle at its centre of mass with axis $O_V z_V$ directed vertically down along the local gravity vector $\boldsymbol{g}$. Axes $O_V x_V$ and $O_V y_V$ are arranged similarly to frame F_E, describing northerly and easterly travel respectively. F_B, the body-fixed frame, is used as a reference for the final form of the force and moment equations. Wind axes could also be used as the body-fixed frame, but is inconvenient for the description of angular motion. Typically F_B is arranged to coincide with the principal axes of inertia of the flight vehicle, providing a simplification of the moment equations. In F_B coordinates the vehicle velocity relative to Earth is $\boldsymbol{v}_B = (u, v, w)$ and the angular velocity is $\boldsymbol{\omega}^B = (p, q, r)$. Here a superscript describes the reference frame the vector is measured relative to and a subscript is used to indicate the coordinate frame in which the vector components are written. Further, the angular velocity vector $\boldsymbol{\omega}$ generally represents the rotation relative to the inertial frame, of the frame of reference indicated by the superscript.

Transferring information between reference frames depends on their relative angular position and angular velocity. To define the orientation of the vehicle, Euler angles have been used. Euler angles describe a transformation that moves F_V into alignment with F_B, through a sequence of rotations (ψ, θ, ϕ). The transformation is described by the matrix

$\boldsymbol{L}_{BV}$, transferring the coordinates of vector $\boldsymbol{v}$ from frame F_V to frame F_B, $\boldsymbol{v}_B = \boldsymbol{L}_{BV}\boldsymbol{v}_V$.

$$\boldsymbol{L}_{BV} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{3.53}$$

The relative angular velocity between the two vehicle-based frames is based on the Euler angles, and is written here in terms of body coordinates.

$$\boldsymbol{\omega}_B^B - \boldsymbol{\omega}_B^V = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi}\sin\theta \\ \dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi \\ \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi \end{bmatrix} \tag{3.54}$$

Equation 3.54 provides a path to defining the Euler rates $(\dot{\phi}, \dot{\theta}, \dot{\psi})$, and tracking the attitude of the vehicle, by further development of $\boldsymbol{\omega}_B^V$. Starting with the Earth's rotation based on a sidereal day, $\boldsymbol{\omega}^E$ can be expressed in the various reference frames of Figure 3.25:

$$\boldsymbol{\omega}_{EC}^E = \begin{bmatrix} 0 \\ 0 \\ \omega^E \end{bmatrix}; \qquad \boldsymbol{\omega}_E^E = \begin{bmatrix} \cos\lambda_E \\ 0 \\ -\sin\lambda_E \end{bmatrix}\omega^E; \qquad \boldsymbol{\omega}_V^E = \begin{bmatrix} \cos\lambda \\ 0 \\ -\sin\lambda \end{bmatrix}\omega^E \tag{3.55}$$

Reference frame F_V rotates according to the curvature of the Earth, and as such is dependent on the rate at which the vehicle is travelling across the surface, $\left(\dot{\lambda}, \dot{\mu}\right)$. The angular velocity of frame F_V relative to the inertial frame is therefore written using $\boldsymbol{\omega}_V^E$ and the relative motion between frames F_E and F_V.

$$\boldsymbol{\omega}_V^V = \begin{bmatrix} \left(\omega^E + \dot{\mu}\right)\cos\lambda \\ -\dot{\lambda} \\ -\left(\omega^E + \dot{\mu}\right)\sin\lambda \end{bmatrix} \tag{3.56}$$

To be compatible with the body angular rates $\boldsymbol{\omega}_B^B$, the left hand side of Equation 3.54 is transformed through a transformation, $\boldsymbol{\omega}_B^V = \boldsymbol{L}_{BV}\boldsymbol{\omega}_V^V$.

For the simplification to a flat Earth model where $\boldsymbol{\omega}^V$ and $\boldsymbol{\omega}^E$ are neglected, such that $\boldsymbol{\omega}_B^B - \boldsymbol{\omega}_B^V \equiv \boldsymbol{\omega}_B^B$, and $[P, Q, R]^{\mathrm{T}} = [p, q, r]^{\mathrm{T}}$. In terms of the reference frames, ignoring the rotation of the Earth leaves F_E as the inertial frame, and ignoring the curvature of the Earth makes F_V equivalent to F_E.

3.7.2 General Equations of Motion

Using the reference frames of Figure 3.25, the general equations of motion for a rigid-body flight vehicle are developed. This is presented in two sections. The first deals with the application of Newton's laws of motion to the vehicle, providing the general force and moment equations. To complete the equation set, the inertial acceleration of the vehicle centre of mass is then derived. Much of the complexity in these equations results from the transformation from inertial to body-fixed coordinates, with the inclusion of the rotation of the Earth through $\boldsymbol{\omega}^E$, and its curvature through $\boldsymbol{\omega}^V$.

The primary assumption in this dynamic model is that of a rigid body vehicle. For the force expression, further simplification results from neglecting the momentum of the fuel through the engine and potential operational mass losses such as ablation. The magnitude of these terms are small compared to the momentum change imparted to the air flowing through the engine. In the development of the moment equations, the contributions from moving aerodynamic surfaces have been neglected.

Dynamics - force and moment equations

In Figure 3.26, an elemental mass dm, moving within an inertial reference frame, is acted upon by a force $d\boldsymbol{f}$. The following general force and moment equations result from applying Newton's laws of motion to the flight vehicle.

$$\text{force equation :} \qquad d\boldsymbol{f} = \dot{\boldsymbol{v}}\, dm \tag{3.57}$$

$$\text{moment equation :} \qquad \boldsymbol{r} \times d\boldsymbol{f} = \boldsymbol{r} \times \dot{\boldsymbol{v}}\, dm \tag{3.58}$$

Defining the vehicle mass centre C by the expression $m\boldsymbol{r}_C = \int \boldsymbol{r}\, dm$, the integration of Equation 3.57 becomes $\boldsymbol{f} = m\boldsymbol{a}_C$, where $\boldsymbol{a}_C$ is the inertial acceleration of the vehicle mass centre. Transferring the components from the inertial frame to the body-fixed frame then provides the force equation in the desired form.

$$\boldsymbol{f}_B = m\boldsymbol{a}_{C_B} \tag{3.59}$$

Force vector $\boldsymbol{f}_B$ is the resultant of all externally applied forces acting on the vehicle, including airframe aerodynamics, propulsive, control, and gravitational forces. The acceleration vector $\boldsymbol{a}_{C_B}$, describes the acceleration of the vehicle mass centre relative to F_I, and is developed further in the following section.

The moment equation is also simplified by defining C as the mass centre, and refer-

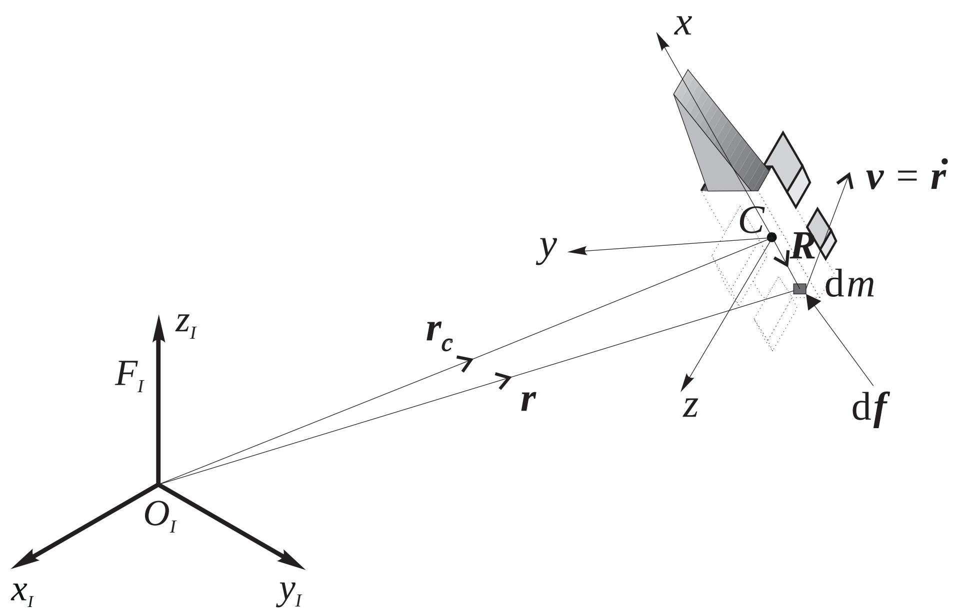


Figure 3.26: Nomenclature for the application of Newton's Law to an element of a body.

encing the integral over the vehicle to the moving point C,

$$\boldsymbol{M}_I = \dot{\boldsymbol{h}}_I, \tag{3.60}$$

where $\boldsymbol{M}_I = \int \boldsymbol{R}_I \times d\boldsymbol{f}_I$ is the resultant external moment about C, and

$$\boldsymbol{h}_I = \int \boldsymbol{R}_I \times \boldsymbol{v}_I \, dm, \tag{3.61}$$

is the resultant angular momentum about C. Again it is desired to express the components of (3.60) in terms of body-fixed coordinates rather than inertial coordinates. This transformation is achieved through the expansion

$$\boldsymbol{M}_B = L_{BI}\boldsymbol{M}_I = \dot{\boldsymbol{h}}_B + \tilde{\boldsymbol{\omega}}_B^B \boldsymbol{h}_B \tag{3.62}$$

where $\boldsymbol{M}_B$ represents the aerodynamic, propulsive, and control moments applied to the vehicle, and $\boldsymbol{h}_B = \boldsymbol{L}_{BI}\boldsymbol{h}_I$, is the transformation of (3.61). Neglecting deformation components - elevator motion, fuel sloshing, and elastic deformation, for example - h_B becomes

$$\boldsymbol{h}_B = \boldsymbol{I}_B \boldsymbol{\omega}_B, \tag{3.63}$$

where

$$\begin{aligned}\boldsymbol{I}_B &= -\int \tilde{\boldsymbol{R}}_B \tilde{\boldsymbol{R}}_B \, dm \\ &= \begin{bmatrix} I_x & -I_{zx} & -I_{zx} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{zx} & -I_{yz} & I_z \end{bmatrix}. \end{aligned} \tag{3.64}$$

The moments of inertia and products of inertia defined in matrix $\boldsymbol{I}_B$ (3.64) respectively take the form $I_x = \int (y^2 + z^2)\; dm$ and $I_{xy} = \int xy \, dm$. The rotation of the Earth though not explicitly appearing in 3.62 and 3.63, occurs implicitly in $\boldsymbol{\omega}^B$.

Inertial acceleration

The reference frames used to derive the equations of motion move relative to inertial space, including an acceleration of the origin and a rotation. To define the position, inertial velocity, and inertial acceleration of the vehicle parallel to F_B, we are therefore required to deal with the arbitrary motion of these frames relative to inertial space. Figure 3.27 shows the framework for developing the necessary expressions. The flight vehicle with mass centre C is represented as a point moving within the arbitrarily moving Earth reference frame F_E. To simplify the notation, the origin of F_E is written as O rather than the explicit O_E.

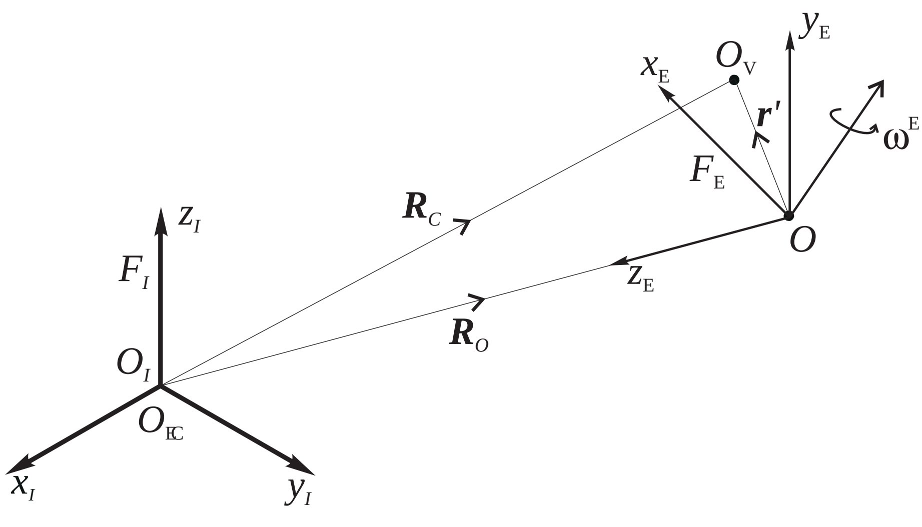


Figure 3.27: Moving reference frame, with reference to Figure 3.25

For two frames moving relative to each other, the expression $\boldsymbol{n}_b = \boldsymbol{L}_{ba}\boldsymbol{n}_a$ describes the transformation of the vector $\boldsymbol{n}$ - observable in both frames - from frame F$_a$ to F$_b$. Assuming frame F_a is fixed and frame F$_b$ moves relative to it, then the components of the derivative of the vector in the moving frame are transformed to F_b using

$$\boldsymbol{L}_{ba}\dot{\boldsymbol{n}}_a = \dot{\boldsymbol{n}}_b + \boldsymbol{\omega}^b \times \boldsymbol{n}_b, \tag{3.65}$$

where $\boldsymbol{\omega}^b$ describes the relative angular velocity between the two frames. This expression was used to form Equation 3.62. Applying (3.65) to the derivative of the vehicle position vector, provides the following expression for the components parallel to the axes of F_E, of the inertial velocity of O_V.

$$\begin{aligned} \boldsymbol{v}_{C_E} &= L_{EI}\boldsymbol{v}_I = L_{EI}\left(\boldsymbol{v}_{O_I} + \dot{\boldsymbol{r}}'_I\right) \\ &= \boldsymbol{v}_{O_E} + \dot{\boldsymbol{r}}'_E + \boldsymbol{\omega}^E_E \times \boldsymbol{r}'_E \end{aligned} \tag{3.66}$$

Differentiating $\boldsymbol{v}_I$ and using (3.66), the components of inertial acceleration parallel to F_E are found:

$$\begin{aligned} \boldsymbol{a}_{C_E} &= \boldsymbol{L}_{EI}\dot{\boldsymbol{v}}_I = \dot{\boldsymbol{v}}_{C_E} + \tilde{\boldsymbol{\omega}}^E_E\boldsymbol{v}_{C_E} \\ &= \boldsymbol{a}_{O_E} + \ddot{\boldsymbol{r}}'_E + \dot{\tilde{\boldsymbol{\omega}}}^E_E\boldsymbol{r}'_E + 2\tilde{\boldsymbol{\omega}}^E_E\dot{\boldsymbol{r}}'_E + \tilde{\boldsymbol{\omega}}^E_E\tilde{\boldsymbol{\omega}}^E_E\boldsymbol{r}'_E, \end{aligned} \tag{3.67}$$

where $\boldsymbol{a}_{O_E} = \dot{\boldsymbol{v}}_{O_E} + \tilde{\boldsymbol{\omega}}^E_E\boldsymbol{v}_{O_E}$, the acceleration O relative to F_I. The matrix equivalent to the vector product has been used, as indicated by the tilde accent. For example, $2\boldsymbol{\omega}^E_E \times \dot{\boldsymbol{r}}'_E \equiv 2\tilde{\boldsymbol{\omega}}^E_E\dot{\boldsymbol{r}}'_E$.

The form of Equation 3.67 can be considered a general expression for the inertial acceleration of a point within a moving reference frame. In the case of flight vehicle simulation, the Earth-fixed surface frame is the moving frame, and the vehicle centre of mass the moving point within that frame. The terms of 3.67 may then be defined as follows:

- $\boldsymbol{a}_{O_E}$: inertial acceleration of the origin of F_E. Following the assumption that the Earth's axis is fixed in inertial space, and that $\dot{\boldsymbol{\omega}}^E = 0$, this term is the centripetal acceleration associated with the Earth's rotation, $\tilde{\boldsymbol{\omega}}^E\tilde{\boldsymbol{\omega}}^E\boldsymbol{R}_E$. The maximum value for $\boldsymbol{a}_{O_E}$ occurs at the equator, and at less than 1% of gravity, this term is neglected here.
- $\ddot{\boldsymbol{r}}'_E$: $\dot{\boldsymbol{v}}^E_{C_E}$, the acceleration of the vehicle mass centre in the moving frame F_E.
- $\dot{\tilde{\boldsymbol{\omega}}}^E_E\boldsymbol{r}'_E$: the tangential acceleration of frame F_E is zero as the Earth spins at a constant rate, $\dot{\tilde{\boldsymbol{\omega}}}^E_E = 0$.
- $2\tilde{\boldsymbol{\omega}}^E_E\dot{\boldsymbol{r}}'_E$: the Coriolis acceleration due to motion of the vehicle within the moving frame F_E. It is dependent on the magnitude and direction of the vehicle velocity, and at orbital speed is around 10% of gravity.

$\tilde{\boldsymbol{\omega}}_E^E \tilde{\boldsymbol{\omega}}_E^E \boldsymbol{r}'_E$: the apparent acceleration of the vehicle mass centre due to the angular velocity of the moving frame F_E. Being less than $\boldsymbol{a}_{O_E}$, this term is also neglected. If the simulation was concerned with accurate navigation or positioning, then this term would not be negligible.

Transforming Equation 3.67 into the moving body fixed frame F_B, makes use of the property $\tilde{\boldsymbol{\omega}}_a = \boldsymbol{L}_{ab}\tilde{\boldsymbol{\omega}}_b\boldsymbol{L}_{ba}$, and again uses (3.65). In this case, the angular velocity in (3.65) represents the angular velocity of F_B relative to F_E, $(\boldsymbol{\omega}_B^B - \boldsymbol{\omega}_B^E)$. The inertial acceleration of the vehicle is thus written with components parallel to F_B:

$$\begin{aligned} \boldsymbol{a}_{C_B} &= \boldsymbol{L}_{BE}\boldsymbol{a}_{C_E} = \boldsymbol{L}_{BE}\left(\dot{\boldsymbol{v}}_{C_E}^E + 2\tilde{\boldsymbol{\omega}}_E^E \boldsymbol{v}_{C_E}^E\right) \\ &= \dot{\boldsymbol{v}}_B^E + \left(\tilde{\boldsymbol{\omega}}_B^B + \tilde{\boldsymbol{\omega}}_B^E\right)\boldsymbol{v}_{C_B}^E, \end{aligned} \tag{3.68}$$

where $\boldsymbol{v}_{C_B}^E$ is the flight velocity of the vehicle relative to the Earth,

$$\boldsymbol{v}_{C_B}^E = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad \text{for a stationary atmosphere,} \tag{3.69}$$

and $\boldsymbol{\omega}_B^B$ and $\boldsymbol{\omega}_B^E$ are respectively the angular velocities of frames F_B and F_E, relative to inertial space

$$\boldsymbol{\omega}_B^B = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad \boldsymbol{\omega}_B^E = \begin{bmatrix} p_B^E \\ q_B^E \\ r_B^E \end{bmatrix} = \boldsymbol{L}_{BV}\boldsymbol{\omega}_V^E . \tag{3.70}$$

3.7.3 System of Equations

The complete system of dynamic and kinematic equations for hypersonic flight simulation are assembled here. They are presented in a form compatible with numerical integration, for the purpose of tracking flight velocity, vehicle position, and vehicle orientation. The dynamics state vector describes the vehicle position, attitude, angular velocity, and flight velocity.

$$\boldsymbol{x}_d = [(R,\mu,\lambda),(\psi,\theta,\phi),(p,q,r),(u,v,w)]^T \tag{3.71}$$

For the combined trajectory and attitude motions, a body-axes form of the equations has been used.

Force equations - for motion of the vehicle centre of mass

The force equations are described in terms of body coordinates by Equation 3.59. Here, $\boldsymbol{f}_B$ is the sum of the aero-propulsive force vector $[F_x, F_y, F_z]^T$ and the gravity vector $\boldsymbol{g}_B$. Following the transformation $\boldsymbol{g}_B = L_{BV}\boldsymbol{g}_V$, with L_{BV} defined by (3.53), the force vector becomes

$$\boldsymbol{f}_B = \begin{bmatrix} F_x - mg\sin\theta \\ F_y + mg\cos\theta\sin\phi \\ F_z + mg\cos\theta\cos\phi \end{bmatrix}. \tag{3.72}$$

Combining the force and acceleration expressions, the vehicle inertial acceleration components in F_B are

$$\begin{aligned} \dot{u} &= (F_x - mg\sin\theta)/m - \left(q + q_B^E\right)w + \left(r + r_B^E\right)v, \\ \dot{v} &= (F_y + mg\cos\theta\sin\phi)/m - \left(r + r_B^E\right)u + \left(p + p_B^E\right)w, \\ \dot{w} &= (F_z + mg\cos\theta\cos\phi)/m - \left(p + p_B^E\right)v + \left(q + q_B^E\right)u. \end{aligned} \tag{3.73}$$

Integration of these equations provides the vehicle velocity, $\boldsymbol{v}_B = [u, v, w]^T$, in body coordinates.

Moment equations - rotational motion about the centre of mass

The simplest form of the moment equations given by expressions (3.62) and (3.63), comes from using principal axes for F_B with a plane of symmetry aligned along C_{XZ}. A diagonal inertia matrix is thus provided. If the time derivative terms of inertia are neglected, the moment equations can be arranged as,

$$\begin{aligned} \dot{p} &= (M_x + (I_y - I_z)\,qr)/I_x, \\ \dot{q} &= (M_y + (I_z - I_x)\,rp)/I_y, \\ \dot{r} &= (M_z + (I_x - I_y)\,pq)/I_z, \end{aligned} \tag{3.74}$$

where the vehicle net aerodynamic moments are given by $[M_x, M_y, M_z]^T$. Integration of (3.74) provides the vehicle angular rates, $\boldsymbol{\omega}_B = [p, q, r]^T$.

Vehicle attitude

Recalling that the orientation of the vehicle relative to the local vertical frame F_V is given by the Euler angle sequence (ψ, θ, ϕ). The matrix equation for integrating the Euler rates and therefore tracking the angular position, is found from Equation 3.54, requiring the relative angular velocity $[P, Q, R]$ between the two body frames F_V and F_B, as input.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{3.75}$$

Relative angular velocity

Expressions for the relative angular velocity are formed from considering $\left(\boldsymbol{\omega}_B^B - \boldsymbol{\omega}_B^V\right)$ using Equations 3.54 and 3.56,

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} - L_{BV} \begin{bmatrix} \left(\omega^E + \dot{\mu}\right)\cos\lambda \\ -\dot{\lambda} \\ -\left(\omega^E + \dot{\mu}\right)\sin\lambda \end{bmatrix} . \tag{3.76}$$

Earth's angular velocity

The vehicle acceleration in body coordinates (3.68) requires the angular velocity components associated with the Earth's rotation, and expressed in body coordinates,

$$\boldsymbol{\omega}_B^E = \begin{bmatrix} p_B^E \\ q_B^E \\ r_B^E \end{bmatrix} = L_{BV} \begin{bmatrix} \cos\lambda \\ 0 \\ -\sin\lambda \end{bmatrix} \omega^E . \tag{3.77}$$

Vehicle position

Spherical polar coordinates are used to locate the vehicle centre of mass relative to the Earth. ($\mathcal{R}$, λ, μ), represent geocentric radius, latitude, and longitude respectively. Their derivatives are related to the movement $\boldsymbol{v}_V^E$, of frame F_V relative to the Earth, giving

$$\begin{aligned} \dot{\mathcal{R}} &= -v_{V_z}^E, \\ \dot{\mu} &= \frac{1}{\mathcal{R}\cos\lambda} v_{V_y}^E, \\ \dot{\lambda} &= \frac{1}{\mathcal{R}} v_{V_x}^E. \end{aligned} \tag{3.78}$$

To provide the components of $\boldsymbol{v}_V^E$, the velocity vector $\boldsymbol{v}_B^E = [u, v, w]^T$ for a still atmosphere, is transferred to the frame F_V.

$$\boldsymbol{v}_V^E = \boldsymbol{L}_{VB} \boldsymbol{v}_B^E, \tag{3.79}$$

where $\boldsymbol{L}_{VB} = \boldsymbol{L}_{BV}^T$, the transpose of the orientation matrix used to rotate frame F_V to the body axes, defined by (3.53).

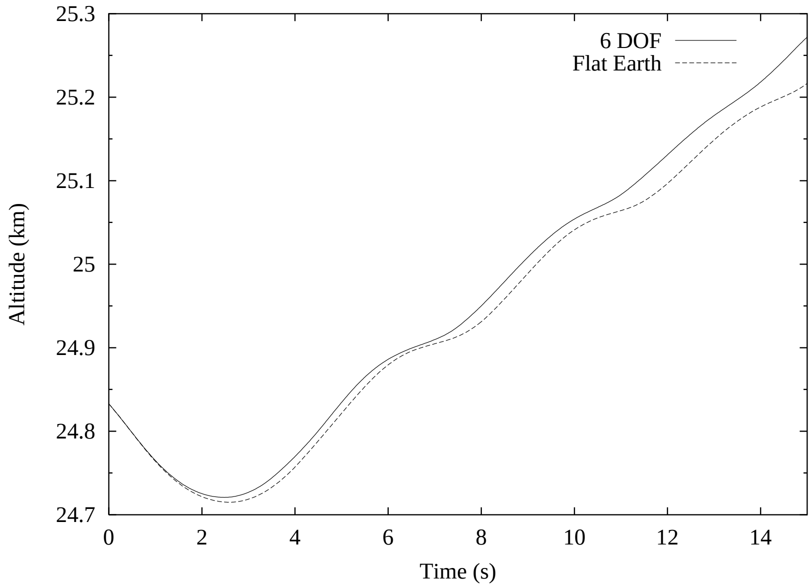


Figure 3.28: Altitude response comparison between the general six degree-of-freedom model and the simplified longitudinal equations.

Longitudinal Flat-Earth Flight Dynamics

To check the derivation of the general six degree-of-freedom dynamic model, the longitudinal dynamics were separately derived for a flat Earth system. These equations can also be produced by the following simplifications to the general equations:

$$\begin{aligned}
\text{Attitude:} \quad & \psi = \frac{\pi}{2}, \phi = 0 \\
\text{Angular rates:} \quad & p = r = 0 \quad ; \quad p_B^E = r_B^E = 0 \\
\text{Forces and moments:} \quad & Z = 0 \quad ; \quad L = N = 0
\end{aligned}$$

The simplification of the flight dynamics in this manner noticeably changes the altitude response characteristics of the vehicle. By example, the simulated altitude responses shown in Figure 3.28 where produced during a flight guidance simulation. The difference is due to the Coriolis acceleration component which appears in the six degree-of-freedom equations, and contributes to altitude gain. This does not appear in the flat Earth model.

3.8 Control Actuator Dynamics

In this hypersonic vehicle study, a rear wing/elevator combination provides the means for active control. The elevator action is specified by an inner loop attitude controller providing an angular rate command, $\dot{\theta}_{\text{e,cmd}}$. An instantaneous change in the angular rate following the command has been assumed. Depending on the quality of the control function, this simplification may contribute to controller sensitivity to high frequency noise generated in the simulation by performance uncertainty and signal noise.

Fuel settings are not actively controlled, but are considered part of the vehicle input vector for the purpose of tracking the rate of mass loss. Changes in the fuel rate to each combustor occur independently as a function of the fixed nominal fuel/air equivalence ratio and a variable mass flow rate of air through the engine modules due to changes in flight condition and vehicle attitude. These changes are assumed to be instantaneous.

3.9 Numerical Integration

The flight simulation history is provided by numerical integration of the flight dynamics equations of motion. In the flow structure shown in Figure 3.4 the integrator is used as a single step procedure for the integration interval Δt. The initial value problem is thus written for the interval $t = [t_i, t_i + \Delta t]$.

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(t, \boldsymbol{x}) \quad \text{for } t \in [t_i, t_i + \Delta t] \\ \boldsymbol{x}(t_i) &= \boldsymbol{x}_i \end{aligned} \tag{3.80}$$

where the state vector $\boldsymbol{x}$ is described by Equation 3.1. One step methods tested for the flight simulation task included a fixed timestep Runge Kutta, and the predictor corrector method of Heun [65].

The fixed timestep Runge Kutta scheme is a fourth order explicit Runge Kutta procedure. An advantage of the scheme is a fixed local error order $O((\Delta t)^4)$, but this is offset by requiring the calculation of four functional values with each integration step. Like all explicit one-step schemes, the fixed timestep Runge also suffers from the need for very small stepsize for solution convergence when dealing with stiff equations. A system of differential equations are said to be stiff if the component functions exhibit very different growth behaviours. For the scramjet flight simulation, altitude and angle of attack have potentially large growth rates relative to the rest of the vehicle states, and are therefore sources of stiffness in the flight dynamics equations. There is also the potential for discontinuities in aerodynamics and propulsion modelling, which can lead to step changes in the vehicle pitching moment.

A stiff set of equations implies that over the time period of interest, there are important features characterized by a much small timescale. To capture these features, an adaptive stepsize Runge Kutta scheme can be employed. However, experiments with a 4th order scheme showed discontinuities in the system model can result in severe reductions in the integration timestep without much gain in overall accuracy, as the algorithm attempts to precisely follow the derivative function. Since the vehicle flow processes are simulated with each functional evaluation, there is a significant computational gain by to be realized by a procedure with fewer function calls, while not subject to the failings of an explicit routine.

The predictor-corrector method of Heun combines explicit and implicit formulae, making it suitable for integrating a system of stiff differential equations. With two corrector iterations, Heun's method has three function calls per integration step, with a fixed local error order $O((\Delta t)^4)$. Over the interval $t = [t_i, t_{i+1}]$, the path from x_i to x_{i+1} follows two intermediate evaluations.

Starting with $\boldsymbol{x}_i = \boldsymbol{x}(t_i)$, the first value $\boldsymbol{x}_{i+1}^{(0)}$ is determined by the explicit Euler method. The implicit update for $\boldsymbol{x}_{i+1}$ is provided by the trapezoidal rule for evaluating the integral $\int_{t_i}^{t_{i+1}} \boldsymbol{f}(t, \boldsymbol{x}(t))\mathrm{d}t$, and is solved here with two iterations. The predictor and corrector steps are thus summarized:

$$\begin{aligned}\text{Predictor:} \quad & \boldsymbol{x}_{i+1}^{(0)} = \boldsymbol{x}_i + \tfrac{\Delta t}{2}\boldsymbol{f}(t_i, \boldsymbol{x}_i) \\ \text{Corrector:} \quad & \boldsymbol{x}_{i+1}^{(\nu+1)} = \boldsymbol{x}_i + \tfrac{\Delta t}{2}\left(\boldsymbol{f}(t_i, x_i) + \boldsymbol{f}(t_{i+1}, \boldsymbol{x}_{i+1}^{(\nu)})\right), \text{ for } \nu = 0, 1\end{aligned} \tag{3.81}$$

After the corrector iterations,

$$\boldsymbol{x}_{i+1} := \boldsymbol{x}_{i+1}^{(2)}. \tag{3.82}$$

To minimize the corrector iteration error, the integration step size Δt, should be chosen according to the Lipschitz condition [65],

$$K = \Delta t L_i \leq 0.20\,, \tag{3.83}$$

where L_i is the local Lipschitz constant for $\boldsymbol{x}_i \in [\boldsymbol{x}_i, \boldsymbol{x}_{i+1}]$,

$$L_i = \max_{1 \leq r,k \leq n} \left|\frac{\partial f_r}{\partial x_k}\right|. \tag{3.84}$$

Applying the Lipschitz condition to the integration of the scramjet vehicle dynamics suggested a timestep of 0.00005 s is required. This value is consistent with the stepsize reduction that the adaptive Runge Kutta scheme undergoes. Due to the computational effort required in evaluating the state derivatives, it was necessary to sacrifice integration accuracy by using a stepsize ($\Delta t = 0.01$ s), whereby $\Delta t L_i \sim 30$. Experiments showed the simulated response for the Heun scheme was equivalent to the Runge Kutta scheme with the same timestep. The results also indicate that satisfying the Lipschitz condition was not required to accurately generate the flight response history. Consequently, for this thesis Heun's method with a timestep of 0.01 s was used for all flight simulations needed by the control design procedure.

3.10 Concluding Remarks

This chapter presented the simulation tools by which hypersonic flight simulation of an air-breathing launch vehicle concept is possible. Featured amongst the simulator components are the estimated vehicle geometry and mass distribution, a vehicle aero-propulsive response model, a high speed flight dynamics model, and a system for integrating the flight dynamics to generate a flight history. The implementation of the flight simulation software is available as FORTRAN code in a supporting technical report [18].

Flight simulation is a major component of the flight control design problem. For the design procedure applied in this thesis, the control law is evolved on the basis of the performance of simulated flight responses. The design and performance of the two control loops (longitudinal control and guidance), are the topics of the following chapters.

CHAPTER 4

Control System Design Tools

The two defining features of a control design approach are the functional representation of the control law and the algorithm by which the control function is designed. Closed-loop control design requires the capturing of system behaviour through a model derived from knowledge about the process to be controlled. Historically, control theory has been developed using linear time invariant models of the system and providing an analytical solution to the control design problem. Within this framework the system knowledge is generally described by transfer functions, frequency response functions, or state space representations. The manipulation of system properties in the frequency domain and continuous time domain has provided the basis for much of the control theory being used today. In the frequency domain, measures such as phase and gain margin are used to configure the control law. Such measures are also used to examine the closed loop stability of controlled system and to provide robustness guarantees. In the continuous time domain, performance measures such as percent overshoot, steady-state error response and response time are applied.

An alternative to the conventional analytical description is the heuristically derived rule-based control. The use of rule base systems is generally related to process control situations, where the action to be taken is reliant on a combination of events of systems states. For ill-defined industrial processes, conventional control methods may fail, either due to the inability in obtaining an analytical system model or because controller input information is imprecise. Uncertainty in the system model used to design the control has, over recent times, led to the development of robust control theories within the continuous time domain. Fuzzy control (FC), as a specific implementation of rule-based control, has also emerged as an alternative to conventional control systems, particularly for complex ill-defined problems where it is difficult to form precise mathematical statements of the system performance. In the realm of process control applications, fuzzy control provides a convenient means of converting a linguistic control strategy based on expert knowledge, into an automatic control strategy [143].

Fuzzy control is considered part of the intelligent control field since it emulates the human decision making process. It has been argued that this constitutes an intelligent system rather than simply an intelligent designer [11]. Together with neural networks, fuzzy control has received increasing attention as a method for the implementation of truly nonlinear control laws. According to the representation theorem of Kosko [124], any continuous nonlinear function can be described by a finite set of fuzzy variables, values and rules. In flight control, nonlinearities are typically represented in a discrete manner, through gain scheduling and mode switching, for example. The skilled human pilot is an excellent example of a highly nonlinear control strategy, which can be duplicated using fuzzy logic control or neural networks [207, 135]. However, due to the difficulty in validating neural network and fuzzy control systems, they are often used to augment more conventional control approaches. Applications include flight control law design through gain scheduling [135], incorporating intelligent behaviour in the outer loop trajectory maintenance [207], and on-line learning [208]. The aircraft carrier landing problem presented by Steinberg [206, 207] shows the performance enhancement FC offers by being able to represent pilot actions and knowledge as an automatic control strategy.

The decision of which control methodology to apply to a problem rests with the unique demands of the system to be controlled. It is generally thought that fuzzy control is left to problems such as process control, which are not readily dealt with by conventional analytic approaches. There is, however, no systematic procedure for the analysis of a system to assess the viability of applying a fuzzy control strategy. Advocates of fuzzy control do not see this as the only area of application [141]. In closed-loop control there are two major classes of applications for fuzzy control: (i) supervision of closed loop operation thereby complementing and extending conventional control algorithms, and (ii) the direct realization of closed loop operation, replacing the conventional control system.

In terms of the hypersonic flight control problem, there have been limited studies on the application of fuzzy control. Christian [43] reported the application of a fuzzy logic controller for the regulation of the acceleration of a hypersonic interceptor. Robustness against large aerodynamic parameter variation was shown. Zhou *et al.* [244] used a hypersonic transporter concept and applied a fuzzy logic based control system to provide longitudinal stability and attitude command tracking. Zhou based the development of the controller on the behaviour of a human pilot. Favourable comparisons were made with a linear proportional-derivative feedback controller and robustness of the FC to variations in flight condition were shown. Despite the simplicity of the system models used in these studies, the non-linear FC laws offer promising performance and appear to be robust in the presence of system uncertainty.

For this thesis, the flight control approach is determined by two constraints. The first is the desire to configure the control law without reducing the vehicle model to an analytical function. The design is thus based on capturing the control function from simulated flight responses, covering the full nonlinear operating characteristics of the vehicle. The second constraint was the desire to use a single control function that provided good performance over a broad range of operation conditions of the vehicle. A fuzzy logic rule base controller was considered due to the reported robustness fuzzy control offers against system uncertainty, and the capacity to represent a complex nonlinear control law. Despite the simplicity of the rule base format there are numerous control parameters which define both global and local features of the control surface.

Though procedures for the design of fuzzy controllers have been developed for special cases, there is no systematic procedure for the design of an FC. The construction of fuzzy control systems is generally a trial and error process. While the trial and error process can be circumvented by first developing a linear controller using conventional techniques and fine-tuning the fuzzy equivalent, the computing power available today makes it possible to automate the trial and error design using a "brute force" approach. The design procedure considered for this thesis uses a genetic algorithm to provide a numerical optimization of the control surface. Such an approach presents the possibility of generating novel solutions to the control problem which may not be reachable from the linear controller. The design procedure is configured as a black box design process, where the only interaction the design algorithm has with the control problem, is the supply of a parameter set and the receipt of a performance measure. The core of the design procedure is a parameter optimization problem based on a genetic search algorithm, with simulated flight responses providing the performance measure needed to direct the search.

Genetic algorithms belong to a collection of zero-order algorithms based on the global search and optimizing capabilities of natural and biological systems. Due to their search robustness over highly dimensional and complex search spaces, they have been widely applied as a way to automate the learning of fuzzy control rules. To configure the flight controller of this thesis, a real-coded genetic algorithm was developed.

This chapter introduces the fuzzy controller and the genetic algorithm. The structural components of the FC are defined and the construction of the control law is discussed with reference to the flight control problem. A general introduction to evolutionary optimization is provided and the implementation of the real-coded genetic algorithm, as used in this study is presented. Further discussion on the genetic algorithm is provided in Appendix A, where standard test functions are use to examine the performance of a modified mutation operator.

4.1 Fuzzy Logic Control

Fuzzy logic models the logic of perception and, in so doing, provides an abstraction of human reasoning. It is a recently defined term promoted by the marketing appeal it offers in favour of the more conventional descriptors of fuzziness, such as “vagueness” or “multivalence”. In contrast to the binary logic which, stated by Aristotle as *everything must either be or not be*, fuzzy logic is inspired by the philosophy of Buddha [125], building on the expression “A and not-A”. The fuzzy controller is built from a collection of *if-then* style rules which are applied in parallel using the mathematics of fuzzy logic.

The principles of multivalued logic were worked on by logicians in the 1920’s, forming the mathematical foundations of fuzzy logic. Further development led to the first fuzzy sets being drawn in 1937 [125]. At the time they were referred to as vague sets. Lotfi Zadeh, a mathematical theorist, was the first to describe vague sets as fuzzy. His seminal papers on the linguistic approach and system analysis using the theory of fuzzy sets [240, 241, 242], were the motivation for the development of fuzzy control. Control applications have since dominated the practical application of fuzzy logic. The ability of fuzzy logic to interpret human operation and reasoning has been particularly attractive in the field of process control where traditional automatic control strategies are outperformed by human operators. Mamdani and his research colleagues pioneered fuzzy logic control applications. Their work during the 1970s on the design of a fuzzy controller for a steam engine [140], was the first practical application of fuzzy logic.

Fuzzy control in its simplest form is described by a set of rules which provide a functional relationship for the set of actions given a set of states. Human language forms the basis of the rules, allowing actions to be taken according to vague descriptions and a reasoning model. The inherent vagueness of fuzzy sets allows the convenient use of linguistic rules in an heuristically defined automatic control strategy. The combination of rule-based systems with fuzzy parameters and fuzzy reasoning, as used by Mamdani, was the framework for subsequent fuzzy control applications. Fuzzy theory has since been applied to a broad range of problems, in a variety of forms: cement kiln control [102], automatic train operation [237], and consumer products such as washing machines. There have also been developments in fuzzy hardware such as fuzzy memory devices and fuzzy computers, promoting the effective utilization of fuzzy control [131].

The relatively recent appeal of fuzzy control (FC) is largely the result of a marketing inspired consumer demand. The washing machine market is a typical example [126, 125], where new sensors have been configured to provide similar information to that used by humans, and fuzzy logic therefore provides a more user friendly interface. Another reason for the appeal of FC are the benefits afforded by upgrading linear controllers to nonlinear algorithms. An additional field of application is for complex, ill-defined systems where analytical or experimental models are limited. In many of these applications, controller design can be performed by writing rules directly.

The potential of fuzzy control has been premised on the basis that fuzzy controllers provide greater robustness than conventional control and that they are more appropriate to the control of nonlinear processes [114]. Research offering support for these superior features is, however, not conclusive. Excluding adaptive forms of fuzzy control, the control law is essentially a static nonlinearity. The possibility that it is more robust to parameter variations relies on being able to recognize the parameter variations and being able to encode them in the rule base. In this manner it is possible to provide seamless transfer between control parameters. In a study by Kortmann [123], a comparison was made between a classical proportional-derivative (PD) controller and a PD-like fuzzy controller, applied to an unstable vertical take-off and landing plane model. The fuzzy controller tested proved to be particularly sensitive to unfiltered noise if the sampling time was too small. It is possible that the inference method used by Kortmann contributed to the observed poor noise robustness, due to the introduction of nonlinearities in the control surface. With regards to robustness against parameter variations, Kortmann's FC was shown to be superior to a conventional PD controller.

If the basic input/output mapping of the fuzzy controller is based on PD control, then its superiority over the linear case for nonlinear systems is reliant on the nonlinearities being a function of the inputs used. Often this requires the use of reference signals in addition to the conventional error and derivative inputs. For example to capture the performance nonlinearities with the vehicle attitude it could be necessary to include the angle of attack in addition to the error, as an input. It is possible that nonlinearities expressed in terms of the state error and its derivative can offer improved performance over linear controllers. This leads to one of the main application areas for fuzzy control, namely the enhancement of linear control laws by using fuzzy logic to separately manipulate different regions of the state space, thereby generating a nonlinear control law. The parameterization of the fuzzy controller allows independent manipulation of the global characteristics generated by the complete rule base and the localized features associated with individual rules.

One of the issues which has tempered the general acceptance of fuzzy logic control is the assurance of stability. Being a nonlinear controller, it is difficult to obtain general results for stability analysis and design [63]. Stability proofs for fuzzy control have been limited to simple proportional-derivative style controllers and for processes which are themselves stable [139]. Stability of FC has therefore generally been addressed through prototype testing. A compelling argument presented by Mamdani [141] is that prototype testing is more important in terms of assessing controller performance. The suggestion is that mathematical stability is not a necessary and sufficient requirement for controller acceptance. The basis for Mamdani's position is that stability proofs require a mathematical model of the process, which may not be available. With regard to stability features of the flight controller in this thesis, extensive flight simulations are performed and thesis include several disturbances and uncertainty features. In a similar manner, flight control law evaluation for the Hyper-X research vehicle [52] performed numerous parametric variations and Monte Carlo analyses with nonlinear flight simulations.

The fuzzy approach to system modelling is distinguished by linguistic variables in place of numerical variables, the use of conditional statements to characterize simple relations between variables, and the characterization of complex relations by fuzzy algorithms [242]. Fuzzy control actions are determined using a knowledge base built from system knowledge (including constraints), fuzzy rules, and an inference mechanism to evaluate the rules. The following sections provide the detail of some key concepts of fuzzy systems, using the configuration of an attitude control law as an example. Further discussion on fuzzy system theory applied to fuzzy control is available from numerous publications, reflecting the now widespread interests in fuzzy systems. The following references were particularly useful in providing an overview to fuzzy control, [125, 131, 132, 114, 63, 58].

4.1.1 Fuzzy Sets and Fuzzy Variables

Classical sets are based on bivalent logic, where crisp boundaries allow an element to either belong or not belong to the set. The membership $\mu_A(x)$ of a classical set A, being a subset of the universe X, is defined by

$$\mu_A(x) = \begin{cases} 1 & \text{iff } x \in A \\ 0 & \text{iff } x \ni A \end{cases} \tag{4.1}$$

These sets are referred to as crisp sets, allowing the membership function to take on only two values, 1 or 0, according to cases where x does or does not belong to A.

Fuzzy sets are a generalization of the ordinary set. Formally defined by Zadeh [240] in 1965, fuzzy sets allow the possibility of degrees of membership, defined by a characteristic function that can take on any value in the interval $[0, 1]$. In contrast to the crisp sets defined by Equation 4.1, the value of $\mu_A(x)$ at x represents the grade of membership of x in A. To differentiate the membership function from probability ideas, it has been referred to as a “possibility” function. A functional representation for fuzzy sets, $\mu_A(x) = f(x)$, is often a convenient way to define the membership function. In control applications function forms typically take the form of triangular, trapezoidal, or Gaussian functions. The simplest definition is available with symmetric triangular or Gaussian functions, which are parameterized by a central location a and a half-width b.

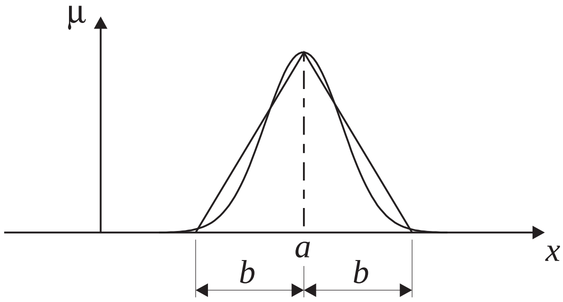


Figure 4.1: Triangular and Gaussian membership functions.

$$\text{triangular:}\quad \mu(x) = \begin{cases} 1 - \frac{|x-a|}{b} & \text{iff } |x-a| \leq b \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

$$\text{Gaussian:}\quad \mu(x) = \exp\left(-\frac{9}{2}\left(\frac{x-a}{b}\right)^2\right) \tag{4.3}$$

For the Gaussian set, the width is assumed to be 3 standard deviations. Fuzzy sets were designed to represent the ambiguity associated with classifying elements into classes. They can be used to represent vague concepts such large and small. Consider, for example, the variable α_{err} representing the angle of attack error internal to the flight controller. The condition statement such as "α_{err} is large", implies a decision to be made according to the degree that value of α_{err} is large.

In system modelling or control, fuzzy sets are used to discretize a fuzzy variable across its domain. The fuzzy variable for α_{err} represents a linguistic interpretation of the error variable. A possible discretization of the α_{err} fuzzy variable is shown in Figure 4.2. Fuzzy sets abbreviated by NL, NS, ZE, PS, and P characterize the error as being negative large, negative small, zero, positive small or positive large. In FC with a continuous variables, the quantization levels of the input variables expresses the sensitivity of the controller to the observed variable. To reduce the sensitivity of the controller to noise it is necessary to have sufficiently wide membership functions [131]. The partitioning is traded off against the benefits of additional degrees of freedom available through finer partitions. For all cases considered in this thesis, an odd number of partitions have been used to discretize the fuzzy variables.

There is considerable scope for fine-tuning the shape of the membership functions to match the variable description, accounting for known nonlinearities, or to enhance fine-tuning. The higher the density of fuzzy sets the more complex the control surface which can be configured. Using nonlinear functions to define the set membership results in a nonlinear interpolation between rules. Though this induced non-linear characteristic may

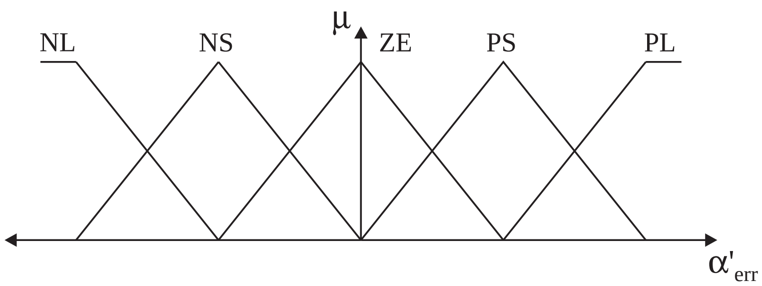
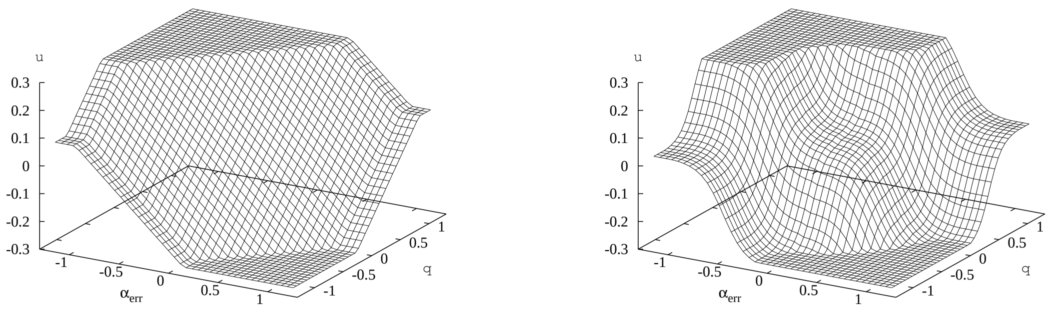


Figure 4.2: Fuzzy sets partitioning angle of attack error.



(a) (b)

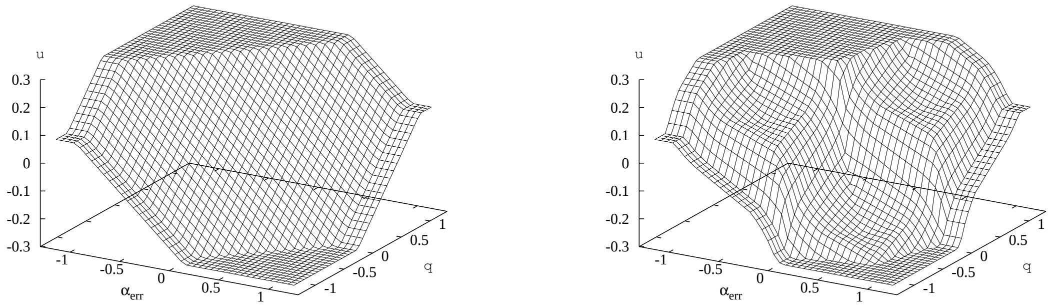
Figure 4.3: Membership function impact on a proportional derivative rule set, with each variable partitioned by three fuzzy sets. (a) Triangular membership functions produce a linear interpolation between the rules, and (b) Gaussian membership functions producing a nonlinear interpolation.

provide the desired control surface, it is considered to be not in accordance with the basic idea of a fuzzy controller. The nonlinearity should be defined by the fuzzy rules, which depend upon the number and distribution of fuzzy sets [114]. Figure 4.3 shows the impact of a nonlinear interpolation on the control surface of a linear PD controller.

4.1.2 Fuzzy Rules and Reasoning

The basis for reasoning with fuzzy logic is a collection of fuzzy propositions. A fuzzy proposition takes the form "x is A", where x is a variable and A is a linguistic variable represented by a fuzzy set. For example, "x is *large*" inquires to what degree x fits the description *large*. The evaluation of a fuzzy proposition measures the degree of membership of x to the linguistic variable, using a membership function, $\mu_A(x)$, which is continuous over the domain of A.

Fuzzy propositions are combined using logical connectives such as "and" and "or". General forms of these are represented by triangular norms (T-norms) and triangular conorms (T-conorms or S-norms) [114, 132], respectively. The following set operators hbtp



(a) **and** $\equiv$ product operator (b) **and** $\equiv$ min operator

Figure 4.4: Control surfaces generated by two common ”and” connectives for a linear PD equivalent fuzzy controller. The normalize input variables were discretized by three partitions.

were originally proposed by Zadeh [240].

$$\boldsymbol{and} \quad \equiv \mu_{A_1 \cap A_2}(x) \quad = \min\{\mu_{A_1}(x), \mu_{A_2}(x)\} \tag{4.4}$$

$$\boldsymbol{or} \quad \equiv \mu_{A_1 \cup A_2}(x) \quad = \max\{\mu_{A_1}(x), \mu_{A_2}(x)\} \tag{4.5}$$

Another commonly used T-norm has the **and** connective represented by a product,

$$\mu_{A_1 \cap A_2} = \mu_{A_1}(x) * \mu_{A_2}(x). \tag{4.6}$$

One or more fuzzy propositional statements can be used in the premise of a fuzzy rule. For the general case where the premise is constructed using variables from different domains, the premise for two inputs may be written as:

$$p: \quad x_1 \text{ is } A_1 \ \boldsymbol{and} \ x_2 \text{ is } A_2 \tag{4.7}$$

The evaluation of the proposition, set by the definition of the **and** connective in 4.7, is nontrivial. In Figure 4.4 the effect of the ”min” and ”product” operators on the controller surface is demonstrated. The controller rules were based on a proportional-derivative linear attitude controller with three partitions per input variable. Nonlinearities generated by the minimum operator are not adjustable by the design process. The finer the partition of the input space, the greater the frequency of nonlinear features on the control surface. It is interesting to note that, of all the possible definitions for the **and** connective, only the *product* operator doesn’t introduce nonlinearities in the control surface [114, 131].

Reasoning with fuzzy logic requires each fuzzy rule to be written as an implication, an *if-then* statement, where the antecedent and the consequent are formed by fuzzy propositions. Fuzzy rules provide a way of expressing control policy and domain knowledge by characterizing a dependency between system variables or features. To describe a multiinput multi-output (MIMO) rule, multiple propositions in the condition and consequence statements are used, with each consequent proposition treated independently. Two types of rules are used in fuzzy control: Mamdani rules and Sugeno rules. Mamdani fuzzy rules originate from the first reported applications of fuzzy control [142]. For a MIMO system with N_X inputs and N_Y outputs, Mamdani rules have the following general form:

$$r_k: \quad \boldsymbol{if}\ x_1 \text{ is } A_{1,k}\ \boldsymbol{and} \cdots x_{N_X} \text{ is } A_{N_X,k}\ \boldsymbol{then}\ y_1 \text{ is } B_{1,k}, \cdots, y_{N_Y} \text{ is } B_{N_Y,k}$$

The output of the rule, expressed here in terms of a fuzzy set, can be a constant numerical consequent. This provides greater freedom in generating the control surface and a simpler rule base evaluation. An alternative format for fuzzy rules which is often used, was devised by Takagi and Sugeno [215] for control of a model car. In the so called TS rules, the rule consequents are written as functions of the controller inputs.

$$\begin{aligned} r_k: \quad \boldsymbol{if} \quad & x_1 \text{ is } A_{1,k}\ \boldsymbol{and} \cdots x_{N_X} \text{ is } A_{N_X,k}\ \boldsymbol{then} \\ & y_1 = f_{1,k}(x_1, \ldots, x_{N_X}), \cdots, y_{N_Y} = f_{N_Y,k}(x1, \cdots, x_{N_X}) \end{aligned}$$

Using Sugeno rules provides a mechanism for interpolating between a set of control functions, effectively providing a fuzzy gain scheduling technique. The main drawback of conventional gain schedulers is the potential for abrupt changes in the control parameters and the need for accurate linear time-invariant models Sugeno rules represent a possible solution to the problems by using a fuzzy reasoning mechanism to determine the control parameters. The rule condition values could also be used to capture nonlinearities with respect to the flight condition and vehicle attitude, while the rule consequents are based on linear controllers designed with conventional control design techniques.

The processing of a fuzzy rule base requires the aggregation of the set of fuzzy rules using the sentence connective *also*. All rules are evaluated in parallel, with the aggregation procedure providing a means to generate a single output description. Along with the method of implication which generates the output for a single rule, the connective *also* has a substantial influence on the quality of a fuzzy model. Lee [132] reports on a study of fuzzy implication and aggregation methods and assembles appropriate pairs for fuzzy control. Of the approaches, those commonly applied are Mamdani's R_c (min implication) and Larsen's R_p (product implication) combined with the union operator for the connective "also". A simplified inference mechanism uses numerical constants in the consequents, allowing the single result to be obtained arithmetically, thus bypassing the complexity of dealing with fuzzy sets in the consequence.

4.1.3 Fuzzy Controller Operation

In fuzzy control as used in this thesis, the control law is evaluated by the parallel action of a set of Mamdani fuzzy rules which, in general, describe a nonlinear mapping of inputs to outputs. Also, the application of fuzzy control has used a simplified fuzzy inference method, because of the computational savings and the flexibility it offers in defining the control surface. The general fuzzy rule using the simplified inference method is described as follows, for a multiple-input single-output controller,

$$\boldsymbol{if}\ x_1 \text{ is } A_{i,1}\ \boldsymbol{and}\ x_2 \text{ is } A_{i,2}\ \boldsymbol{and}\ \ldots\ x_n \text{ is } A_{i,n}\ \boldsymbol{then}\ y \text{ is } w_i$$

where $A_{i,j}$ defines the membership function of the j-th input of the i-th rule, and w_i is the scalar output value for the i-th rule. The procedure for evaluating the numerical output, using the simplified inference method, is described by the following steps:

Step 1: Input variables are scaled, mapping the physical values of the state variables onto a normalized domain. The normalization factors can be equivalent to the inverse of the gains used in conventional linear controllers. Each fuzzy proposition x_j is $A_{i,j}$ used in the premise of the rules is evaluated by matching the appropriate membership function to the the input. The membership degrees $\mu_{A_{i,j}}(x_j^*)$ of the j-th input are calculated, where x_j^* represents the scaled numerical value of the j-th input.

Step 2: The firing strength, or degree of fulfilment (DOF) β_i of rule i, is evaluated using the appropriate T-norm for the **and** connective in the condition statement. Results in this thesis have used the product operator with all conditional statements of equal importance,

$$\beta_i = \prod_{j=1}^{n} \mu_{A_{i,j}}(x_j^*). \tag{4.8}$$

Step 3: The calculation of the output from the rule base combines the implication relation for each rule, as defined by the the *if-then* relationship, and the aggregation of the rules with the **also** connective. Using the simplified inference method, the control command u, is evaluated by a weighted average of the rule outputs based on the firing strength of the rules,

$$u = k_u \; \frac{\sum_{i=1}^{N_r} \beta_i \, y_i}{\sum_{i=1}^{N_r} \beta_i} \tag{4.9}$$

where k_u is the scaling factor used to transfer the normalized output variable to a physical control command.

Though the preceding steps are applied to Mamdani fuzzy rules, an equivalent expression can also be applied to Sugeno type rules. The method is considered a special case of the product-sum-gravity method [81, 114], where the terms *product*, *sum*, and *gravity* respectively refer to the product operator for the condition statement, the sum operator for combining the output fuzzy sets from each rule, and the centre-of-gravity method to find the numerical output from the aggregated rule outputs.

4.1.4 Designing the Fuzzy Controller

The fuzzy controller (FC) can be represented in functional form, in the same manner as a conventional control law:

$$\boldsymbol{u}(k) = \boldsymbol{k}_u F(\boldsymbol{k}_e \boldsymbol{e}(k), \boldsymbol{k}_x \boldsymbol{x}(k)) \tag{4.10}$$

where the controller output $\boldsymbol{u}$ at some sampling instant k, is expressed as a nonlinear function F of the system state $\boldsymbol{x}$ and the state error $\boldsymbol{e}$. The scaling factors $\boldsymbol{k}_u$, $\boldsymbol{k}_e$, and $\boldsymbol{k}_x$ have a similar role to the gains in conventional controllers. In the flight control problem of this thesis, the FC is applied in a regulatory manner for the maintenance of vehicle stability and attitude. The basic structure of its operation is defined in Figure 4.5. Transformation of the control input signals ($\alpha_{\text{err}},\ q, \theta_{\text{e,err}}$) to the elevator actuation command $\dot{\theta}_{\text{e,cmd}}$ is achieved via the parallel processing of an array of rules, according to the steps set out in the previous section, and using the knowledge base defined by the data base and rule base. The data base describes the storage of input variable definitions and the array of outputs. The rule base defines the structure of each rule using the various combinations of condition statements available through the partitioning of the input space.

Where conventional control systems have a control algorithm based on a set of analytical equations, FC's are knowledge based system. They require a means of knowledge representation (a set of rules), a reasoning strategy (a definition of how the rules are processed), and a means of acquiring the knowledge. The four principal means of knowledge acquisition are expert human knowledge, based on operator control actions, based on a fuzzy model of the process, and learning based on experience [131, 63]. The overall design requires the specification of many parameters and, as a trial and error process, can be very time consuming. Without an expert to provide the knowledge, a learning or adaptive process is required. One approach to simplify the development of the fuzzy controller is to first establish a linear controller using conventional analytic design methods and to then fine tune the fuzzy equivalent. Such an approach has been used by Ying [239]. The

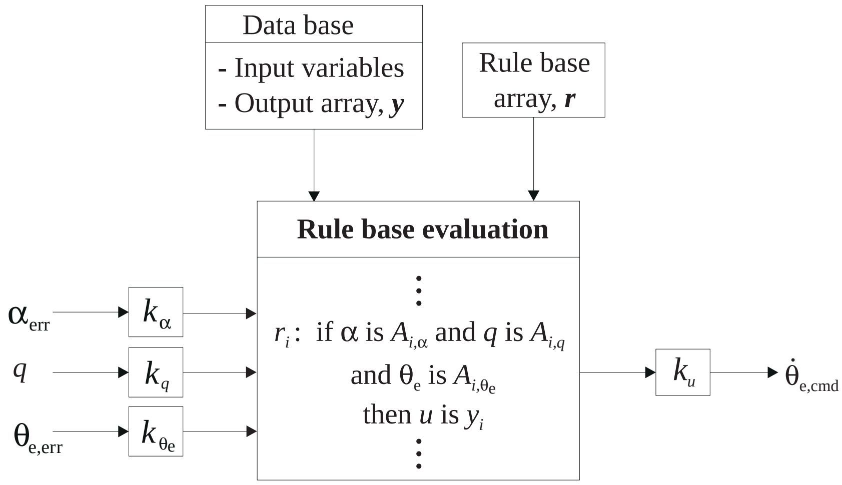


Figure 4.5: Structure and operation of the inner-loop fuzzy controller.

.

approach allows the transfer of local stability analysis from the linear case to the tuned nonlinear fuzzy controller.

A popular approach over the last decade has been the coupling of the search capabilities of evolutionary algorithms to the automated design of the fuzzy controller. The knowledge acquisition process is transformed into a numerical optimization procedure, where the evolution of control parameters is directed towards providing optimal performance relative to desired performance characteristics. Other naturally inspired algorithms, such as simulated annealing [44], have also been applied to the design of fuzzy control [107]. Genetic algorithms (GAs), as used in this thesis, are zero-order search procedures based on the mechanics of natural genetics. They differ from most other search methods used for optimization, in that the search works with a population of possible solutions and the transition rules are probabilistic rather than deterministic. As a direct consequence of these differences, GAs are capable of performing a global search on large complex problems which may be characterized by a stochastic cost function. In addition to the design of fuzzy controllers, GAs have also been used in rule discovery systems [90], to search a stochastic robustness cost function in robust control design [144], in combination with gradient-based optimization for robust control design of multivariable systems [169], and applied to the optimization of various aerospace control systems [128].

A summary of the application of genetic algorithms to offline and online design of fuzzy control system is provided by Linkens and Nyongesa [136]. The work contained in this thesis has been developed concurrently with the appearance in the literature of applications combining genetic algorithms and fuzzy control. Many propose the simultaneous design of membership functions and the rule sets [103, 224], providing a general knowledge acquisition procedure. However, the present design approach is focused on the learning of rule consequent values $\boldsymbol{y}$, assuming a fixed rule base structure and a fixed set of input variable definitions. With each rule having an independent consequent value, the design allows manipulation of the local features of the control function throughout the full range of the input variables. Experiments were also conducted on the additional design freedom of tuning the input scaling parameters, thereby affecting global features of the control function.

In designing the fuzzy controller, it is desirable for the rule base to display the properties of continuity, and completeness [131]. Continuity follows the general preference for smooth control surfaces, and depends on exposing the design procedure to as many simulation examples as is practicable, ensuring full coverage of the input space and consideration of the range of possible variations in system performance. Completeness infers a proper control action for every state of the system within the bounds of the input variables. The realization of completeness is attainable through the appropriate discretization of the input space and the formation of the rule base.

Evolutionary design of a controller is often characterized as a *brute force* approach, as computational power is exploited in place of more conventional analytical approaches. The main contributor to the often considerable computational effort, is the coupling of the large number of function evaluations required in the search for *good* solutions, and the evaluation of the objective function used to classify the quality of the control solution. Computational power continues to increase however, and with the application of parallel computation, the impact of large populations can be mitigated by the simultaneous evaluation of individuals in the population. The remaining hurdle is then the desire for a realistic representation of the design environment. Uncertainty models may cover the robustness requirements of the controller, but their worth relies on a base model of sufficient complexity.

Part of the experiment of this thesis is the use of relatively small populations and generations. The main reason is the large cost associated with performing many flight simulations for each evaluation of the objective function. Using the full non-linear vehicle dynamics with a vehicle performance simulation model computed in-line with the dynamics integrator, the control design approach is well deserving of the *brute force* title. The applicability of intelligence based techniques to the design is thus dependent on the optimization performance of the genetic algorithm. Application of the GA to test functions as shown in Appendix A, indicated that high quality solutions are found with a relatively small scale search. In designing the flight controller, the population size and generations required are strongly related to the criteria used to evaluate control performance. Chapter 5 provides a complete description of the design setup and the objective function evaluation. The remainder of this chapter describes the genetic algorithm used to design the flight controllers and a simplex method which was used for fine-tuning a linear fuzzy controller.

4.2 Evolutionary Design

Evolution, as defined by Darwin [51], is a process of gradual change driven by natural selection, where natural selection is a process which selects ultimately for reproductive success. Modern evolutionary theory has been established since the 1930's, through the synthesis of Darwin's theory of evolution and Mendelian genetics [152]. Though Mendel's experiments on pea plants in 1865 had little impact at the time, they would eventually profit him the title of *the father of genetics*. The robustness, efficiency, and flexibility of biological systems inspired the development of *evolutionary algorithms*, or heuristic search techniques in the form of directed probabilistic procedures. In general, methods which simulate evolution are characterized by a population-based search approach that relies on selection and random variation. They are well suited to search through large and complex solution spaces, such as those associated with designing control strategies, financial market predictions, and function optimization. Another similarly inspired algorithm is *simulated annealing* [44, 197], which uses random processes to help guide a search for minimal energy states. On many nonlinear optimization problems classical techniques such as gradient descent, deterministic hill climbing, and purely random search, have proven unsatisfactory.

Evolutionary computation can be traced back to the late 1950's, through the works of Box [33], Friedberg [80], Bremermann [34], and others. The majority of applications today draw from the three main algorithmic approaches: evolution strategies, evolutionary programming, and genetic algorithms. These areas have been developed almost independently, each for a specific application and each emphasizing different features as being necessary for a successful evolutionary process.

Evolutionary strategies, developed in Germany by Rechenberg [179] and Schwefel [193], began as numerical optimizers for both continuous and discrete problems. They are self-adaptive with deterministic, extinctive selection, and use normally distributed mutation as the main operator. The self-adaptive feature controls the strategy parameters for the mutation probability density functions.

Evolutionary programming was introduced by Fogel [76, 75] as a technique for searching through a space of small finite state machines, with the aim to predict environmental changes by creating an artificial intelligence. Also self-adaptive, evolutionary programming used a selection scheme which was probabilistically extinctive and employed mutation as the only operator.

Genetic algorithms were formulated by Holland [101] in the 1960's and further developed by Holland and colleagues [55] at the University of Michigan. Originally devised as a means to model the adaptive processes as it occurs in nature, Holland also recognized the potential of incorporating the mechanisms of natural adaption into an adaptive search algorithm. Genetic algorithms introduced a population-based algorithm, with probabilistic selection as a form of natural selection, and reproduction through crossover and mutation. Of the genetics-inspired operators, crossover is the main operator with mutation being considered a background operator.

Darwinian evolution appears as an optimization process and it is this nature which is generally exploited in the application of evolutionary algorithms. Comparative performance of different evolutionary algorithms indicate that one type is not universally preferable to others, often leaving the choice of algorithm to personal preference. For the work in this thesis, a genetic algorithm approach was chosen to design parameters of a fuzzy flight controller, see Section 4.1.4. Like other general iterative nondeterministic algorithms - simulated annealing [197] and tabu search [189, 59], for example - the genetic algorithm is computationally simple and easy to implement, yet has proven to be robust and effective in producing high quality solutions for large, complex problems. They exhibit hill climbing capability, show asymptotic convergence to an optimal solution, and are able to exploit domain specific heuristic information to bias the search [189]. Being a blind search however, a stopping criteria must be supplied to indicate a sufficiently evolved solution.

What follows is a general introduction to the application and performance of a genetic algorithm. This is then extended to the structure and operation of the GA used in the controller design task of this thesis.

4.2.1 A General Description of Genetic Algorithms

There is no strict definition for structure and operation genetic algorithms, however, it is generally accepted that a population of individuals is evolved through the selection of individuals for mating according to fitness, and the creation of new offspring by crossover and random mutations. The general aim of the GA is to improve the fitness of individuals across generations. The basic structure and operation of a simple genetic algorithm, as defined by Goldberg [83] and others, is summarized in Figure 4.6. The general purpose of the algorithm is: for a function of k variables, $f(x_1, \ldots, x_k) \; : \; \mathbb{R}^k \rightarrow \mathbb{R}$, evolve a population of individuals with the aim of maximizing f. The length of the search is typically constrained by a maximum number of generations. For each generation, parents are selected to mate and properties from mating pairs are recombined through crossover and mutation. Parents are replaced by their children with the proviso that the best individual so far, according to the relative objective function value, is copied into the next generation. This is the so-called elitist strategy.

The beginning of GA research is considered to be the publication of Holland's book, *Adaption in Natural and Artificial Systems* [101], in 1975. Holland provided theoreti-

```
begin genetic algorithm (par,F_obj)
{
  t := 0;
  init P(t);                        % randomly initialize the population
  evaluate (P(t), F_obj);           % evaluate the fitness of all initial individuals
  while (t ≤ T) do
    t := t + 1;                     % index the generation counter
    P'(t) := select P(t);           % select parents for reproduction
    recombine P'(t);                % recombine the genes of selected parents
    mutate P'(t);                   % perturb the mated population
    evaluate (P'(t), F_obj);        % evaluate the new population
    P := elite (P, P'(t));          % transfer the new population
  end
end
}
```

Figure 4.6: Structure and operation of a simple genetic algorithm.

cal and empirical proof of the capacity of genetic algorithms to robustly search complex spaces. Using a binary alphabet to encode information, Holland's theory used a building block referred to as schema, a set of genes representing a partial solution to a problem. The idea is that *good* solutions to a problem are generated by discovering, emphasizing, and recombining good building blocks of a solution in a highly parallel manner. Schemata were used to define subsets of similar chromosomes, representing hyperplanes in an n-dimensional space, where n is the number of genes in an individual. Schemata are a pattern matching devices, or templates, used to explore similarities among chromosomes. However, they are not explicitly dealt with in GA operation. The performance of a genetic algorithm is expressed by the growth equation, see [83], which relates the effect of selection, crossover, and mutation on the number and type of schema processed. It indicates that selection increases, exponentially, the sampling rates of above-average schemata, but does not introduce new information (schemata). Crossover enables structured, yet randomized information exchange allowing new schemata to be introduced, while mutation introduces greater variability into the population. The efficient operation GAs relies on the exchange of information in a highly parallel manner, referred to as implicit parallelism. Without extra processing requirements, the solution space is searched through a simultaneous search effort in many hyperplanes.

The representation of problem information rests with: (i) the choice of alphabet used to encode information within an individual chromosome; and (ii) the distribution of information amongst a population of individuals. Both the encoding and distribution issues are fundamental to the operation of the genetic algorithm.

Ever since Holland's work on the schema theorem proving the operation of GAs, a binary alphabet has been favoured [83]. From the perspective of a genetic algorithm being an algorithm that processes schemata, the binary alphabet ensures the relevance of short low-order schemata and provides the maximum schema processing per bit of information of any coding [101]. A side effect of an alphabet with low cardinality is that, for parameter optimization problems, large string lengths are required to encode the problem information. For example, for a function with 100 variables requiring minimization with precision to two decimal places between -10.00 and 10.00, a string length of 1100 bits is needed. Performance, in terms of the search time required, can be relatively poor for problems of this size, however, improvements in fine local tuning and processing time can be achieved through smart operators or using Gray coding [200]. Gray coding uses a binary alphabet, with a representation scheme that ensures adjacent integers differ by one digit only.

Since the properties of Holland's schema theorem are not limited to binary strings, other alphabets have been used, but proof of performance is generally reliant on empirical results. Though contradicting the idea of low cardinality being optimal, the benefits offered by a floating point encoding scheme for continuous variables problems are many. By a near-direct mapping of variables in the chromosome, the GA is moved closer to the problem space, thus removing the abstract nature of binary encoding. Because precision depends on that available in the computing machine, real-coded genetic operators offer fine-tuning capabilities, and a computational saving is obtained when compared to a binary encoded GA. There is also support for an improvement in search robustness when using real-valued vectors within the GA [154].

The other representation issue asks the question of how to distribute problem information amongst the individuals of a population. There is generally a choice between two approaches: the Pitt approach and the Michigan approach. The Pitt approach gained its name from De Jong and his students from the University of Pittsburg, who used it to code parameter values in individuals [56]. Each individual in the population is encoded with all the parameters of a possible solution. The alternative is to have each member of the population representing a single parameter or subset of parameters (say a single rule for a controller) and the entire population forming a complete solution to the problem. This approach was introduced by Holland at the University of Michigan as a model for classifier systems and subsequently became know as the Michigan approach. For all applications in this thesis the Pitt approach [56] was used to code parameter values in individuals. For the controller design procedure this has each individual representing an entire rule set so that each generation consists of a population of possible rule sets.

Numerous variations of the simple algorithm shown in Figure 4.6 are possible. Applications are generally described through chromosomal representation, selection schemes, population and generation control, and genetic operators. Other biological inspired features can also be included in the algorithm, but have not been examined in this work: for example, virus infection [130], and age structure [129].

Despite their description as a general purpose algorithm, they remain, as do all search algorithms, subject to the “no free lunch” (NFL) theorem for optimization [235]. According to the NFL theorem, an algorithm that does particularly well on average for one class of problems, must do worse on average over the remaining problems. The implication is that for optimal performance the definition of the GA must be tuned to the specific problem. There are a number of implementation issues which influence the efficiency and efficacy of the global search. A typical issue addressed while formatting the algorithm is the potential for premature convergence of the population to a sub-optimal solution. Obvious causes that could be suggested include: insufficient population, insufficient generations, and function characteristics such as many local minimum or plateaus. More important, however, are algorithmic issues such as the information encoding scheme, selection mechanism, and the genetic operators.

Premature convergence occurs when rapid convergence early in the evolutionary process allows the population to be dominated by better than average individuals which then stagnate the search at a less than optimal solution. Most attempts to improve the convergence of GAs have looked at the selection process, which directs the search across generations [83]. However, since premature convergence is related directly to the diversity of individuals within a population, perturbation operators also play a role in the efficacy of the search. Further discussion on this topic is included in Appendix A.

Another implementation issue relates to the performance benefits available with fine-local tuning of the solution space. To achieve fine-local tuning when using a GA, a number of options are available. Holland recognized that local search required higher order schemata while the driving force of a GA was the processing of short, low-order schemata. This prompted the suggestion that GAs be used to perform the initial search and then employ a local search technique on the best individuals. Integration of two schemes can be such that a simplex method [166] may be used inline with a genetic algorithm to provide faster convergence rates along with fine-tuning capabilities [238]. Grefenstette [89] also noted the usefulness of invoking a local search technique once high performance regions of the search space have been identified by the GA. Also, it is possible to improve the fine-tuning capabilities of a genetic algorithm through smart genetic operators with mutation typically targeted for this task. The formation of a fine-tuning feature is dependent on the chromosome alphabet and the encoding scheme. For a binary encoded chromosome fine-tuning can be initiated through limiting available mutations to the least significant digits in genes as the search proceeds [155]. When using a floating-point encoding where each gene is a real-number, a fine-tuning mutation operator can be formed by limiting the magnitude of possible mutations, indexed against the iteration (or generation) number. An adaptive operator operator of this form was formulated for genetic algorithms by Michalewicz [154], and referred to as non-uniform mutation. A modification of the operator was necessary for this thesis, and is discussed in detail in Appendix A.

It is worth noting at this point, that the real valued capabilities of the genetic algorithm have existed for some time in evolutionary strategies and evolutionary programming [24]. For example, the fine-tuning mutation function of evolutionary strategies is controlled by the standard deviation of the randomly distributed additive mutation. Chromosomes in evolutionary strategies consist of a pair of vectors where one defines the search space and the other a vector of standard deviations used by the mutation operator. The possibilities for the acceleration of optimization through reducing the variance of a Gaussian mutator function have been discussed by Atmar [15]. Since gross evolutionary optimization generally occurs quite quickly, reducing the effect of unconstrained variation across generations accelerates the optimization process.

4.2.2 Real-Coded Genetic Algorithm (RCGA)

The genetic algorithm implemented in this thesis is based on the simple genetic algorithm structure presented by Goldberg [83], and employing real-valued chromosome representation. Real-coding for genetic algorithms refers to the representation of an individuals' chromosome as an array of floating-point values. The length of the chromosome is therefore the same as the length of the solution vector to a problem, so that each gene represents a variable of the problem. Since the controller design requires many expensive flight simulations to evaluate the performance of each potential control solution, it is desirable to rapidly acquire good solutions. From this perspective, and for parameter optimization problems in continuous search spaces, the real-coded genetic algorithm is superior to a binary-coded algorithm [154].

Following the simple algorithmic structure shown in Figure 4.6, the design of the genetic algorithm involves the specification of the following items: a parameter encoding scheme; a means of describing the initial population; a scaling function to convert the objective function into a non-negative fitness value for compatibility with the GA selection scheme; a selection scheme to decide which individuals are allowed to reproduce; reproduction operators that produce offspring from parent individuals; and finally; a termination scheme. The following sections develop the RCGA in terms of these features. The notion of population entropy is also introduced as means of examining the dynamics of the population. For a complete listing of the FORTRAN code used to implement the algorithm, a technical report is available [19].

Encoding

Genetic algorithms are a population-based search technique. For the control design problem of this thesis, each individual $\boldsymbol{C}_j$, of the population $\boldsymbol{P}_k$, is encoded with all the control design parameters. Each individual therefore represents a possible solution to the control problem.

Using real-coding, the jth individual is defined by a chromosome vector $\boldsymbol{C}_j$, where each element x is a floating point value within a predefined range $x_j \in [a_j, b_j]$,

$$\boldsymbol{C}_j = (x_{1,j}, x_{2,j}, \ldots, x_{n,j}) \,. \tag{4.11}$$

The population for the kth generation, $\boldsymbol{P}_k$, consists of an array of chromosomes,

$$\boldsymbol{P}_k = [\boldsymbol{C}_1, \ldots, \boldsymbol{C}_n] \tag{4.12}$$

Initialization

The initialization process for genetic algorithms requires the construction of a population $\boldsymbol{P}_0$, of individuals. Mirroring the primordial population of natural evolution, the initial population is typically constructed of randomly generated solution vectors. Of the alternatives to random initialization, many advocate seeding the initial population with existing solution vectors. Grefenstette [89] showed that seeding the initial population with members thought to have high performance can be beneficial. This result can be used in micro-GAs [47] where small populations are used and the process is continually restarted. Davis [53] suggests extending the random search for each member and selecting the best for the initial population. The idea is that even if the same number of function evaluations are performed, an improved final solution can be achieved by extending the initial random search.

Since the primary experiments of this thesis relate to the abstraction of a control design without prior knowledge, a randomly initialized population was generally used. The potential of seeding the initial population was investigated for the flight control problem and further discussion is provided in Chapter 6. The algorithm is constructed such that each parameter can be described by a search domain independent of all other parameters, allowing a direct mapping of the problem space to that used by the algorithm. In the case of the flight control design problem, all parameters are randomly sourced from the range [0,1]. Since procedures were required to map the algorithm parameter set to various parameters defining the controller, scaling was considered part of the conversion.

Population Entropy

A useful measure of the diversity of the population is established by population entropy. It applies the concept of entropy as it is used in information theory [172], where it is referred to as Shannon's entropy [195]. Bessaou and Siarry [28] used the entropy measure in deciding whether to accept a new chromosome in the initialization process of sub-populations. The definition of population entropy provided in their text is reproduced here. It has been used in the results presented in Chapter 6 to examine the search behaviour of the genetic algorithm.

For a population of size N_P, the entropy of the jth gene is

$$H_j(N_P) = \sum_{i=1}^{N_P} \sum_{k=i+1}^{N_P} -P_{ik}\log(P_{ik}) \tag{4.13}$$

where P_{ik} represents the probability that the value of the j th gene of the i th chromosome is different to the value of the j th gene of the the k th chromosome. P_{ik} is evaluated using the following expression:

$$P_{ik} = 1 - \frac{|x_j(i) - x_j(k)|}{a_j - b_j} \tag{4.14}$$

where $[a_j, b_j]$ describes the search domain for the jth gene. The average entropy $H(N_P)$ of the population is then equal to the average of the entropies of the different genes.

$$H(N_P) = \frac{1}{n}\sum_{j=1}^{n} H_j(N_P) \tag{4.15}$$

$H(N_P)$ scales with the size of the population. In Chapter 6 the search behaviour of different sized populations is considered, so the entropy measure is normalized by the entropy of the initial population.

Individual Evaluation

The relative worth of a solution is judged using a scalar variable called the objective function, f_{obj}. The objective function is a property of the problem space and is, of course, application dependent. In the control design problems f_{obj} is evaluated from simulated flight responses, using performance measures such as settling time, steady state error, and the integral of absolute error. A weighted sum of the multiple objectives provides the required scalar value. Constraint violation can be readily incorporated in the form of penalties.

Scalarization of the objective is mandatory when applying evolutionary algorithms.

However, because candidate solutions are processed in parallel, the algorithm is particularly suited to multi-objective optimization. The most common approach when dealing with a multi-objective problem uses an aggregation (or weighted sum) of the individual objectives. In many cases this may require a profound understanding of the solution domain. For situations where the performance objectives are noncommensurate, it may therefore be desirable to provide a nondominated set of solutions, known as Pareto-optimal solutions [77, 245]. Pareto-optimal refers to the set of solutions for which the corresponding objective vectors cannot be improved in any dimension without degradation in another. A commonly encountered example in design would be the dual optimization of cost and performance. For the ultimate selection of a solution, however, it remains necessary to scale the relative importance of the various objectives.

Evolutionary algorithms are notoriously opportunistic, making the construction of the objective function for complex systems a far from trivial task. While this property readily exposes flaws in the definition of the objective function, it also enables the generation of solutions for situations when the parameter encoding or a simulation component is erroneous. For the flight control problem both these situations were repeatedly encountered. Further discussion on the objective function used for the flight controller is contained in Chapters 5 and 6.

Fitness Scaling

The two extremes of population behaviour, in terms of the relative performance of individuals, occurs at the beginning and end of the simulated evolutionary process. Early in the evolutionary stages there may be a small number of very fit individuals which could tend to dominate the next generation and lead to premature convergence, whereby the search direction is focused early rather than later. With later generations, there may be little difference between the average and best performing individual, resulting in average members being given the same reproductive chances as the best members. To regulate the competition between members of a population, it is necessary to scale the objective functions of a population. This process is referred to as fitness scaling, and describes the transformation of an objective function f_{obj} to a fitness value f. The fitness value is then used by the selection procedure to discriminate between the reproductive chances of individuals.

In Holland's original GA [101], the fitness value was implicitly assumed to be non-negative. The reliance of non-negative fitness values by the selection scheme means the algorithm is formatted as a maximization procedure. Fitness scaling therefore has the additional purpose of transforming the objective function to non-negative fitness values where the greater the fitness the better the individual. The detailed formulation of the objective function for the flight control design problem is covered in Chapter 5.

For RCGA, the individual objective function values are mapped to scaled fitness values using sigma truncation followed by linear scaling. The scaled fitness values ensure the selection pressure remains relatively constant through the generations. Here selection pressure refers to the degree to which highly fit individuals are allowed many offspring. The following steps convert the raw fitness values f to a scaled fitness measure f' of generation t:

1. The raw fitness data f is simply the objective function data

$$f_j(t) = \begin{cases} f_{\text{obj},j}(t) - \min(f_{\text{obj}})(t) & \text{if } \min(f_{\text{obj}})(t) < 0 \\ f_{\text{obj},j}(t) & \text{otherwise} \end{cases} \tag{4.16}$$

   where the subscript j refers to the j'th individual of the population.

2. Sigma scaling is applied to transform the fitness values relative to the fitness distribution of the population.

$$f_j(t) = \begin{cases} f_j(t) - \bar{f}(t) + C_M\sigma(t) & \text{if } f(\boldsymbol{C}_j(t)) > (\bar{f}(t) - C_M\sigma_f(t)) \\ 0 & \text{otherwise} \end{cases} \tag{4.17}$$

   where $\sigma_f(t)$ is the standard deviation of the current population, $\bar{f}(t)$ is the mean (non-negative) objective value of the current population, and C_M is a constant. The value $\bar{f}(t) - C_M\sigma_f(t)$ represents the minimum acceptable objective measure for any reproducing individual, and with $C_M = 2$, implies 5% of the population on average are allocated zero fitness. This avoids the potential of poorly performing individuals, that may be created through crossover or mutation, to bias the selection pressure. This would happen if fitness scaling were referenced to the worst performing individuals [55].

3. Linear scaling is applied with the transfer parameters a and b evaluated so that the average scaled fitness is equal to the average pre-scaled value, and the maximum scaled fitness is a preset multiple C_M, of the average fitness, see Figure 4.7.

$$f'_j = af_j + b \tag{4.18}$$

   where

$$\begin{aligned} a &= \bar{f}\frac{C_M - 1}{f_{\text{max}} - \bar{f}} \\ b &= \bar{f}(1-a) + af_{\text{min}} \end{aligned} \tag{4.19}$$

   If f'_{min} is evaluated to be less than zero then the parameters a and b are adjusted to provide $f_{\text{min}} = 0$ and $\bar{f}' = \bar{f}$.

$$\begin{aligned} a &= \frac{\bar{f}}{\bar{f} - f_{\text{min}}} \\ b &= \frac{-a}{f_{\text{min}}} \end{aligned} \tag{4.20}$$

According to the recommendation from Goldberg [83], C_M is typically set in the range [1.2,2.0] for populations in the range $N_P \in [50, 100]$. The implication is that for small populations the selection pressure needs to be large enough such that the fitter individuals benefit, but not so large that population diversity is lost early in the evolutionary process. For the control design results presented in Chapter 6, populations ranging in size from 6 to 100 were used, with $C_M = 2$.

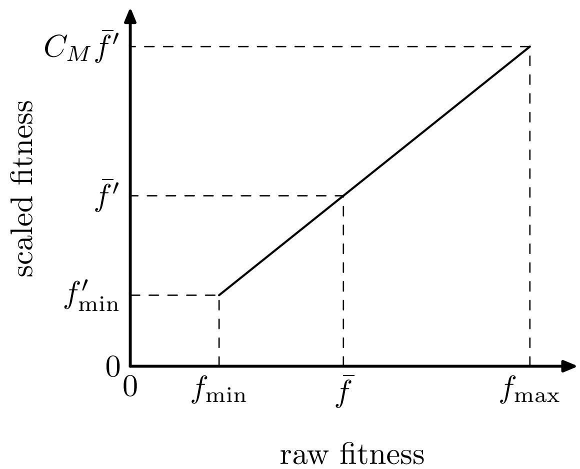


Figure 4.7: Application of linear scaling to map raw fitness values to scaled fitness values.

Selection

According to the theory of natural evolution, natural selection selects for reproductive success. Selection is one of the main operators of an evolutionary algorithm, providing the driving force behind the search process by emphasizing better solutions in the population. The algorithmic form of natural selection is to allocate the offspring generating chances of an individual according to its fitness. Using this approach, individuals with a greater than average fitness have their reproductive chances enhanced, while still allowing the below average members a chance to reproduce. Selection is typically described with a probabilistic nature, requiring the fitness values to be non-negative and the search procedure to be configured as a maximization task. The algorithm is also provided with a degree of robustness against a noisy objective function [94].

Like all facets of evolutionary algorithms, the implementation of selection has been approached in numerous ways [84, 22, 94]. Proportional selection [86] allocates the reproductive chances of an individual in proportion to its fitness relative to the population. Tournament selection [30] conducts a series of tournaments between individuals randomly chosen from the population, with the winner inserted in the next generation. Ranking selection [87] is similar in operation to proportional selection, however the probability of selection is based solely on the ranking of individuals (according to fitness) within the population. The principles of simulated annealing have also been employed in evolutionary algorithms, through Boltzmann selection mechanisms [138].

Proportional selection as introduced by Holland [101], first creates a probability distribution proportional to fitness, and then draws samples from this distribution. Roulette wheel selection is one such scheme, where the selection proceeds by consecutive spins of a roulette wheel, with each slot sized according to the individuals probability of selection, p_i,

$$p_i = \frac{f_i'}{\sum\limits_{i}^{N_P} f_i'} \ . \tag{4.21}$$

There is the possibility that, with the roulette wheel procedure, the fittest individual in the population may be assigned no offspring in a particular generation.

The mechanism used in RCGA follows the proportional selection schemes established by De Jong [55], and improved by Brindle [35] and Booker [31]. De Jong proposed the use of expected value selection, whereby the selection probability is expressed on the basis of the expected offspring generating chances of an individual,

$$e_i = N_p \frac{f_i'}{\sum\limits_{i}^{N_P} f_i'}, \tag{4.22}$$

where N_p is the population size. A procedure referred to as *stochastic remainder selection without replacement* [83], has been implemented in RCGA. It is a two step process, with the construction of a mating array of individuals, followed by the stochastic sampling of the array to establish mating pairs.

The method of constructing the mating population is to first assign positions equal to the integer value of e_i. The preselection process is then completed by filling the mating array using probabilistic selection, with the fractional part of the expected values e_i describing the probability of selection to the mating population. At this stage, once a copy of an individual is added to the mating population further contributions are not permitted. With the fitness scaling scheme previously described, the best individual in the population is guaranteed to have C_M copies in the mating population. Following preselection, mating pairs are then formed by randomly selecting parents from the mating population, and then removed to assure each preselected parent generates offspring.

An additional feature of the selection policy used in RCGA is the application of elitism. It provides an assurance that the maximum objective function value within a population is not reduced across generations. If none of the offspring of the new generation constitute an improvement in the best individual, the best-so-far individual is copied into the new population by replacing the worst individual.

Crossover

The regeneration of a population of parental individuals is subject to the variation operators of crossover (or recombination) and mutation. For genetic algorithms the emphasis is typically on the search capabilities of crossover and selection. Crossover once distinguished genetic algorithms from other evolutionary algorithms, though this is no longer the case, with crossover also being used in evolutionary strategies [193, 24]. For parent individuals selected for reproduction, crossover achieves a recombination of chromosome data, providing a structured but randomized mechanism for the offspring to inherit characteristics of both parents. Atmar [15] suggests the biological function of crossing over serves an informational maintenance purpose, which is similar to the role it plays in genetic algorithms. By exchanging information between diverse chromosomes, crossover in genetic algorithms enables new parts of the solution space to be tried. There is no guarantee however, that good chromosomes will generate even better ones through crossover.

The likelihood of two mating parents undergoing crossover is preset by the crossover rate (or probability), $p_\mathrm{c} \in [0, 1]$. If crossover is not performed, parent values are copied directly to the offspring chromosome, with the possibility of subsequent mutations.

Compared to the crossover operation for binary strings, recombination with real-valued vectors can be implemented in many forms [99]. Due to the history of evolutionary strategies with real parameter optimization, many of these forms are derived from efforts in that field [32]. The simplest form of crossover is based on exchanging information using a randomly chosen reference point $i \in \{1, 2, \ldots, n-1\}$ along the chromosome. If the two chromosomes $\boldsymbol{C}_1 = (x_{1,1}, \ldots, x_{n,1})$ and $\boldsymbol{C}_2 = (x_{1,2}, \ldots, x_{n,2})$ are selected for recombination, simple crossover generates the following offspring,

$$\begin{aligned} \boldsymbol{C}_1' &= (x_{1,1}, \ldots, x_{i,1}, x_{i+1,2}, \ldots, x_{n,2}) \\ \boldsymbol{C}_2' &= (x_{1,2}, \ldots, x_{i,2}, x_{i+1,1}, \ldots, x_{n,1}) \end{aligned} \tag{4.23}$$

The simple crossover is readily extended to multiple-point schemes or a uniform crossover where each element of the new chromosomes will typically be sourced with equal probability from the parents. An alternative to the simple exchange of information between parents is available with intermediate operators, which attempt to blend the components across the parents. The general procedure for generating a weighted average of the parent chromosomes is referred to as arithmetic crossover [154]. Given two chromosome vectors $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, arithmetic crossover results in the following linear combination of chromosome information.

$$\begin{aligned} \boldsymbol{C}'_1 &= \lambda \boldsymbol{C}_2 + (1-\lambda)\,\boldsymbol{C}_1 \\ \boldsymbol{C}'_2 &= \lambda \boldsymbol{C}_1 + (1-\lambda)\,\boldsymbol{C}_2 \end{aligned} \tag{4.24}$$

where $\lambda \in [0, 1]$ is simply chosen at random for each mating pair, and applied uniformly across the chromosome. Variations on this scheme include having λ calculated independently for each pair of chromosome elements, and a non-uniform operator where λ is variable and dependent on the age of the population.

The preferred crossover form is dependent on the characteristics of the search space, as defined by the objective function. One of the limitations of the uniform arithmetic crossover is that the bounded operation favours exploitation of the chromosome features rather than exploration. This places greater emphasis on permutation operators such as mutation to explore the fitness domain. If the search objective is to find the global optimum to high precision, then a degree of experimentation with the crossover definition is worthwhile. However, if the search objective is to find a solution which satisfies some performance bounds, then the search is robust for a range of crossover forms. Both the single point and the arithmetic crossover proved capable of designing the flight control functions for this thesis. Herrera *et al* [99] provides an empirical study of the performance of several crossover operators. The best operators were those that considered the exploration intervals for obtaining offspring genes. Appendix A contains further discussion on the performance of the crossover operator.

Mutation

The original formulation of genetic algorithms emphasized recombination, with mutation being a dedicated background operator via a low level of activation [101]. This reflects the occurrence of mutation in nature, where it is rare with often disastrous consequences. The primary functions of mutation in the genetic algorithm can be separated into the maintenance of population diversity, and the initiation of new search paths through the introduction of new information. As a perturbation function, it can also be used to fine-tune the search, by including a self-adaption feature which reduces the magnitude of the perturbations as the search proceeds.

In binary coded schemes, mutation simply involves the inversion of bits. However, like crossover, there are many mechanisms for the perturbation of floating point values [99, 193]. To be equivalent to the binary case, the probability of mutation is typically much higher in real-coded GAs.

Some evolutionary schemes use mutation as the primary or only regeneration operator [26]. The mechanism has therefore been investigated as a means to improve the velocity and reliability of the genetic search. Recent studies have shown the benefits of using high rates of mutation, which decrease over the course of the evolution [21, 73, 205]. With gross evolutionary optimization occurring quite rapidly, it has also been shown that, by reducing the variance of a Gaussian mutator function as the optimum is approached, the search can be dramatically accelerated [15, 26].

The non-uniform operator introduced by Michalewicz [155] provides a step-size control mechanism for the mutation of real-valued vectors, by reducing the likelihood of large mutations as the search proceeds. In effect, the operator performs similarly to those used in evolutionary strategies, where the width of a Gaussian mutation function is adapted through the search. The basis of the non-uniform operator is the perturbation of a chromosome element through an addition or subtraction to the original, with the probability of large mutations decreasing across the generations. During the course of this thesis, a bias to the centre of the search domain was observed through the unreliable design of the inner loop flight control function, see Chapter 6. A simple redefinition of the operator allows the non-uniform mutation to exhibit the desirable properties of a random walk for early generations and, as the search progresses, fine-tuning of an individual's chromosome. An empirical study of the non-uniform mutation operator and a proposed modification is presented in Appendix A. The modified operator is defined as an *adaptive range* mutation and has been used to generate the flight control designs presented in this thesis.

Each gene x_i of the newly formed offspring chromosomes, undergoes mutation with a preset probability, p_m. The mutation is effected within the variable range, $x_i \in [a_i, b_i]$, producing the mutated value x_i'. There are two steps to the mutation process. The first establishes the mutation range $[\sigma_L, \sigma_U] \leftarrow x \pm \Delta(t, \delta_{\text{max}})$ based on the generation number t, a fixed maximal half-range δ_{max}, and the perturbation function Δ,

$$\Delta(t, \delta_{\text{max}}) = \delta_{\text{max}} \cdot \left(1 - r^{\gamma(t)}\right) \tag{4.25}$$

with r a uniform random number from the range $[0, 1]$, and $\gamma(t)$ providing the fine-tuning capability according to the function

$$\gamma(t) = \left(1 - \frac{t}{T}\right)^{\beta} \tag{4.26}$$

Here T is the maximum number of generations and β the strategy parameter which sets the degree of non-uniformity across the generations. Figure 4.8 plots the normalized perturbation function $\Delta(t,y)/y$ as a function of the random variable r for $\beta = 2$. It shows the possible mutation magnitude decreasing across the generations. To ensure the mutation remains bounded by the variable search range, the mutation range is limited by the variable bounds,

$$\begin{aligned} \sigma_L &= \mathbf{max}\{a_i, x_i - \Delta\} \\ \sigma_U &= \mathbf{min}\{b_i, x_i + \Delta\} \end{aligned} \tag{4.27}$$

The second stage is the actual mutation of the gene, which returns a random value with the range $[\sigma_L, \sigma_U]$, with the assurance of symmetry about the parent value x_i.

$$x_i' = \begin{cases} x_i - (1-2p)\ \ (x_i - \sigma_L) & \text{if } p \leq 0.5 \\ x_i + (2p-1)\ \ (\sigma_U - x_i) & \text{otherwise} \end{cases} \tag{4.28}$$

where p is a random number uniformly distributed within the range $[0,1]$. Figure 4.9 shows the relative mutation frequency for mutations about initial values near the centre and edge of the search domain. Reflecting the behaviour shown in Figure 4.8, the operator uniformly accesses the space for early generations, and with increasing generations, becomes more localized in its search. The mutation profile is similar to a Gaussian distribution which is often used to described the mutation in evolutionary strategies.

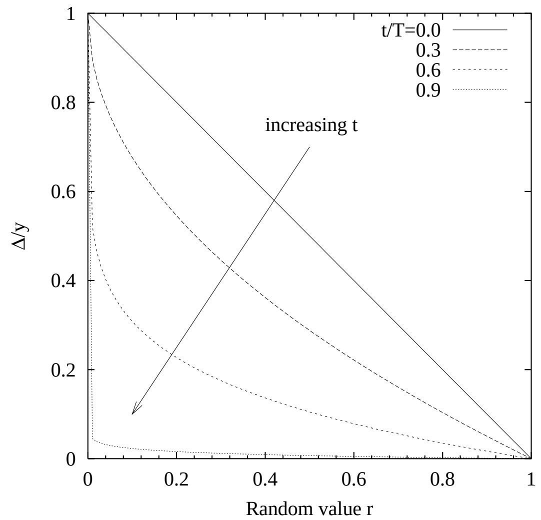


Figure 4.8: Behaviour of the normalized perturbation function $\Delta(t,y)/y = \left(1 - r^{\gamma(t)}\right)$, in terms of the random number r, with $\beta = 2$.

The *adaptive range* definition of Michalewicz's non-uniform mutation operator, improves the reliability of the genetic search. It removes the bias of the Michalewicz non-uniform operator for the centre of the search domain by fixing the maximal possible mutation magnitude rather than having it dependent on the variable position within the search

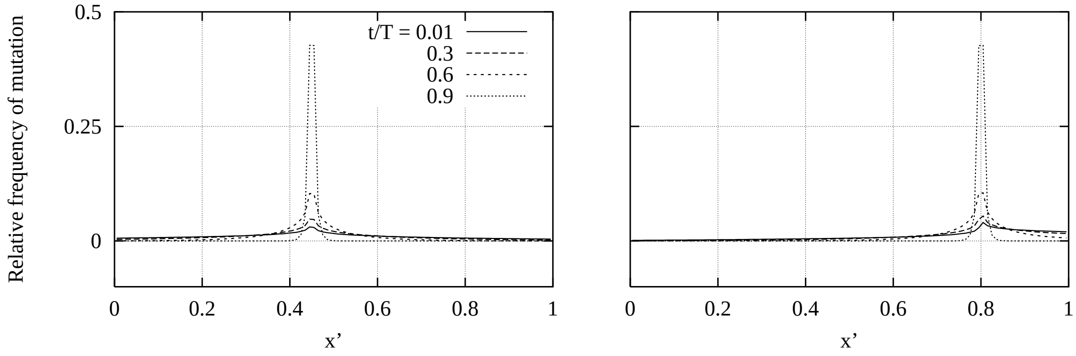


Figure 4.9: Mutation behaviour ($\beta = 2$) for an initial value near the centre of the search domain (left) and a value near the edge (right). The relative frequency represents the probability that a value in the search domain is reached through mutation.

domain. The search potential for arbitrarily complex functions remains constrained by the rapid reduction (dependent on β) in the probability of large mutations. As discussed in Appendix A, this feature does not prevent the algorithm from finding *good* control solutions. Rather, for some function minimization problems, it may be the difference between a very good solution and the global minimum. However, the algorithm performance using the adaptive range mutation can be augmented by the crossover operator, increasing the generation number, the parameterization of the mutation operator, or a change to the perturbation function to allow the possibility of large mutations further into the evolutionary process. The operator as defined in the above equations has proved sufficient for the flight control design problem of this thesis.

Termination

From the viewpoint of optimization by a genetic algorithm, criteria for terminating the search generally fall into two categories, expressed in terms of the search characteristics. The first, which is explicitly coded in the RCGA, measures the search progress (using the objective function) in a predefined number of generations. If no progress is made, or if progress is less than some tolerance, then the search is terminated. The second method is based on a comparison of the chromosome structure amongst the population. Termination is based on the number of converged chromosome elements being a preset percentage of the total number of elements. Depending on the objective function properties, the population convergence test represented in the RCGA implies that chromosome elements are converged. For the flight control problem, relatively few generations are used, and a termination criteria based on convergence is not needed. The applicability of the above methods is further limited by the control objective function changing across the generations.

From an engineering perspective, the termination can be expressed in terms of satisfying criteria for the control performance quality. The control parameters can therefore be optimized against an acceptable level of system performance. Though control solutions may be well established early in the genetic search, the fine-tuning capability of the RCGA can significantly improve the solution quality. For the hypersonic flight control problem it was considered desirable to fully exploit the tuning potential within a predefined number of generations.

4.2.3 RCGA Parameterization

Despite evolutionary algorithms possessing considerable robustness to the parameter settings for a given set of evolutionary mechanisms, there are benefits in terms of solution quality, search robustness, and computation time, in the careful selection of parameter settings. One of the implications of the *no free lunch* theorem is that, for optimal search performance, the algorithm must be tuned to the objective function. The tuning process typically involves finding *good* values through experimentation [83, 88]. Parameter control mechanisms have also been used to provide some means of adaptive the settings during the design process [64].

For the application of the RCGA to the control design problem, the emphasis was search efficiency. Due to the computation cost of evaluating the objective function, it was necessary to minimize the population size and the number of generations. Over a range of experiments it was found that suitable solutions could be obtained with relatively few function evaluations, 150 000 - 250 000 for problems having on the order of 30 - 250 parameters. The search performance was seen to be relatively insensitive to the setting of operator rates, p_c and p_m. When using standard minimization tests such as those in Appendix A, the aim is to find the global solution to a high precision, so much larger populations are often evolved over thousands of generations. Though the design of the flight controller is also described as an optimization problem, it is not an absolute requirement that the design generated be a global optimum. The success of the design is largely dependent on the use of sufficient generations to establish a good solution base and for fine-tuning solution performance. Table 4.1 summarizes the operational parameters of the RCGA used for the control design problem. Some discussion on the application of alternative mechanisms is included in Appendix A.

4.3 Nelder-Mead Simplex Method

The hybridization of evolutionary algorithms and hill-climbing methods provide a tool which exploits global search capabilities of evolutionary algorithms with a means of systematic fine-tuning. Methods of coupling the two search mechanisms vary from the sim-

Table 4.1: Parameterization of the real-coded genetic algorithm. The numbers in brackets indicate the typical values used for the control design problem, with dimension ranging from 27 to 250 design variables.

Operation	Mechanism	Parameterization
Initialization:	Random and seeded populations.	N_P (30-50)
Search length:		N_G (500-1000)
Fitness Scaling:	Sigma truncation with linear scaling.	C_M (2)
Selection:	Stochastic remainder without replacement.	
Recombination:	Whole arithmetic crossover.	p_c (0.6)
Mutation:	Adaptive range mutation.	p_m (0.1-0.3), β (2-5)

ple augmentation of the solution returned by the evolutionary search, to the intermittent application of a localized search throughout the global search [181]. To provide a means of fine-tuning an existing control solution (which may have been evolved through application of the genetic algorithm), the Nelder-Mead Simplex Method was used. Simplex optimization methods are class of gradient-search algorithms that do not require the evaluation of derivatives to determine the search direction. They are therefore suitable for situations where the analytical description is complex or unavailable. The simplex approach to optimizing physical processes or mathematical functions was introduced by Spendley *et al.* [201]. The methods derive their name from the geometric figure which is moved along the surface defining the objective function, in search of the minimum. In the Nelder-Mead method [161, 165], the simplex is able to reflect, extend, contract, or shrink, to conform to the surface topology of the objective function. Modifications by Routh *et al.* [185] and Parker *et al.* [168] led to improvements in the speed and accuracy of the minimization search. As a minimization procedure, the adaption of the simplex is such that it moves away from high values of the objective function, rather than moving directly towards the minimum.

The application of simplex methods generally requires a design space possessing a well defined global minimum. As a general N-dimensional minimization code it benefits from the simplicity of the code organization and the robustness of operation. The algorithm is entirely self-contained, not requiring one-dimensional minimization methods such as is needed for Powell's method [173]. The Nelder-Mead procedure used is based on the Fortran code presented by O'Neill [166]. While the genetic algorithm works to maximize the performance objective, Nelder-Mead acts to minimize. In light of the suggestion by Olsson and Nelson [165] that the procedure is less desirable for large problems with many constraints, it is to be expected that the fine-tuning of the control parameters be relatively inefficient. However, for the final stages of tuning the control parameters and for a moderate number of parameters, the computational effort may be justified by a significant improvement in the performance of the controller.

CHAPTER 5

Configuration of the Flight Control Design Experiments

In its most general form, the control design approach used here is described as a *black box* optimization problem. Such an approach can be easily dismissed with arguments of inelegance and lacking in theoretical grounding, but there are potential advantages. The main asset is that, outside of computational time, there are no restrictions on what is placed inside the black box. From the optimizer's perspective, it sends out a set of parameters and receives a scalar performance measure. For the control design problem this means that the design can be based on the full-nonlinear flight response characteristics of the vehicle, which can include realistic representations of uncertainty (through fluctuations in the nominal performance), disturbances, and noise. On the down side, the assurance of performance and stability robustness does not come cheaply. Covering the full range of state variations requires a large number of sample conditions. This requirement is especially important for a rule based control arrangement since each sample condition may activate only a small number of rules. In contrast, with a constant gain controller, all control parameters continuously contribute to the control command.

The principal computational results of this thesis focus on the capability of an evolutionary based optimizer algorithm to design, without *a priori* knowledge, a robust fuzzy control law for a hypersonic concept vehicle. This work experiments with the potential of fuzzy control to represent a complex, nonlinear, and robust control function, the incorporation of robustness features in the control performance measure, and the capability of the genetic algorithm as a search procedure. The design procedure is similar to the offline learning of a neural net. The structure of the fuzzy rule base defines the mapping procedure and the design procedure learns the output profile through numerical optimization. It also has similarities with the modern developments in stochastic robustness, where Monte Carlo evaluation provides an assessment of controller performance in terms of the probability that a collection of performance metrics are satisfied, and the design maximizes the probability of success.

In the preceding chapters the flight control structure and the design tools were introduced. The flight simulator described in Chapter 3 provides the means of computing the dynamic behaviour of an air-breathing hypersonic aircraft for the purpose of evaluating the controlled flight performance. To provide a link with the preceding chapters and the results in Chapter 6, this chapter discusses the practical features of generating the robust longitudinal flight control laws. These include the overall control design arrangement, the controller parameterization, and the specification of the objective function.

5.1 Overall Approach to Controller Design

The core of the control design approach is a genetic search for control parameters using numerical flight simulations to assess the performance of possible solutions. Within the longitudinal autopilot configuration introduced in Chapter 2, there are two control functions which require specification, the guidance function and the inner-loop attitude control function. These are specified by first addressing the inner-loop controller such that the vehicle is stable and can robustly track attitude commands. For this design stage, the performance analysis is based on stability and attitude maintenance over a short timescale (2 seconds), using a step response in angle of attack. The second stage of defining the gains for the guidance function makes use of the stable inner-loop design and performs simulations over 30 seconds.

The focus of this work was the design of a fuzzy rule base controller for the inner-loop. Through the many degrees of freedom available in detailing the transformation of input values to a control command, an extremely complex control function can be configured. There are obvious benefits for the configuration of an optimal control law, however, it also means the search for a solution is of high order, over a potentially complex design space. Because the evaluation of the objective function is computationally demanding, it was desirable to have a design procedure which would allow rapid determination of the control function. To this end, a real-coded genetic algorithm was constructed and used to search for an optimal control configuration. The real-coded form has been shown to be both more efficient and more reliable in numerical optimization problems of high dimension [154].

There are many variations on this theme and, to simplify the description of each specific design arrangement, a set of indices are used to define the setup for the genetic algorithm ($\boldsymbol{GA}_i$), the controller (C_i), and the objective function (F_i). The following sections discuss the initialization of the genetic algorithm, and the design options available with the inner-loop and outer-loop control functions.

5.1.1 Genetic Algorithm Setup and Operation

Considerable effort is usually undertaken to investigate both the arrangement of genetic operators and their parameterization. For the real-coded genetic algorithm (RCGA), the performance was examined using a benchmark control problem [17] and a collection of function minimization problems, see Appendix A. The original format of the operator [155] greatly inhibited the ability of the genetic algorithm design process to satisfy the target criteria of the control problem. It was through this experimentation that the need for a modification to a well known mutation operator was recognized. Appendix A details the new mutation operator and also addresses some general performance characteristics of the RCGA. Primarily, for the RCGA, the reliable generation of good solutions was relatively insensitive to the setting of activation rates of the variation operators. The results of Chapter 6 also show that relatively small populations (30 - 50) and relatively few generations (~ 500) were sufficient for the design of controllers ranging in size from 27 to 228 parameters. This is due, in part, to the definition of an objective function which changes as the search proceeds, gradually providing greater demands on the controller in terms of the vehicle response. With the function minimization problems which feature in Appendix A, there is no avoiding the complexities of the topology of the function and the use of large populations together with large generation numbers is often unavoidable.

The basic construction of the real-coded genetic algorithm (RCGA) was established in Chapter 4. Vector $\boldsymbol{GA}$ is used to classify the RCGA application:

$$\boldsymbol{GA} = [N_P, N_G, p_c, p_m, \beta] \tag{5.1}$$

Since the operators used for the flight control problem were predominately arithmetic crossover and adaptive range mutation, their selection has not been included in $\boldsymbol{GA}$. Table 5.1 collects the various GA settings used for the results presented in Chapter 6.

Table 5.1: GA parameterization.

$\boldsymbol{GA}_i$	N_p	N_g	p_c	p_m	β
1	6	100	0.6	0.3	5
2	6	30	0.6	0.5	4
3	30	500	0.6	0.2	2
4	30	500	0.6	0.2	4
5	30	500	0.6	0.1,0.2	2,5
6	50	500	0.6	0.2	4
7	10 - 100	500	0.6	0.2	2

Through the course of experimenting with the control design problem, the genetic algorithm was applied in a number of distinct forms. These included seeding the population with existing solutions, dividing the initial search effort into sub-populations which are then recombined, and using the algorithm as a fine-tuning tool. Despite the obvious attraction of these variations, there was in general, no clear benefit over applying the algorithm in its conventional form: evolution of a single population which is randomly initialized. All the applications presented in Chapter 6 employ the conventional algorithm structure of randomly initialized population which is exposed to the full extent of the design problem for all generations. The present application also performs the design as a single processor operation. Since the genetic algorithm is an inherently parallel process, there are significant performance gains to be had by distributing the evaluations of individuals to separate processors. In this manner, the computation could be reduced from 24 hours to an hour say, and the genetic algorithm would then be more attractive as a design tool.

5.1.2 Longitudinal Attitude Controller

Vehicle stability with attitude maintenance is provided by a longitudinal fuzzy attitude controller, positioned in a feedback loop. The rule base which forms the control function consists of three inputs, $(\alpha_{\text{err}}, \dot{\theta}, \theta_{\text{e,err}})$, and one output, $u = \dot{\theta}_{\text{e,cmd}}$. Symmetric triangular memberships are used to uniformly partition the input space and a scalar output is used for the control command. Design trials with Gaussian membership functions were generally unsuccessful compared to those using triangular partitions, though a thorough examination of the design possibilities has yet to be completed. The number of rules is implied by the degree of partitioning of the input space, $N_r = \prod_{j=1}^{N_j} p_j$, where p_j is the number of partitions for the j^{th} input variable. Since each partition of an input variable corresponds to a possible condition statement of the form "*if* x_i *is* A_{ij}", this arrangement provides a rule base with all possible combinations of condition statements. With a high level of partitioning, there will be rules (once the control solution is reasonably evolved) which are never fired since they would represent an unrecoverable vehicle condition.

The basic rule base structure is fixed for all design examples. Within this constraint two design cases are presented in the results. The first uses predefined input scaling values and a design task of determining the rule output array $\boldsymbol{y}$, of dimension N_r. In the second case, input scaling, $\boldsymbol{k}$, is included in the design task, meaning the search is performed for $N_r + N_I$ parameters. Table 5.2 defines the controller definitions and the design parameters, for the results generated in Chapter 6.

A constant gain linear feedback controller was also considered, providing a benchmark for the fuzzy controller. The design task for the linear controller is simply the evaluation of the feedback vector, $\boldsymbol{K} = [K_\alpha, K_q, K_{\theta_\text{e}}]$.

Table 5.2: Fuzzy controller definitions.

Rules	Partitions			Design parameters
	α_{err}	$\dot{\theta}$	$\dot{\theta}_{e,\text{err}}$	
27	3	3	3	$\boldsymbol{y}$
125a	5	5	5	$\boldsymbol{y}$
125b	5	5	5	$\boldsymbol{y}, \boldsymbol{k}$
225a	9	5	5	$\boldsymbol{y}$
225b	9	5	5	$\boldsymbol{y}, \boldsymbol{k}$

5.1.3 Longitudinal Guidance System

The design of the longitudinal guidance laws is reliant on having a stable inner-loop response. Since the focus of the control design was vehicle stabilization and attitude maintenance, a relatively simple feedback law was used to guide the vehicle along the nominal trajectory,

$$\alpha_{\text{cmd}} = \boldsymbol{K}_G \cdot [h_{\text{err}},\ \dot{h}_{\text{err}}]^T \ . \tag{5.2}$$

The guidance design problem is therefore the evaluation of the gain vector $\boldsymbol{K}_G$.

5.2 Construction of the Objective Function

In optimal control theory the desired performance is expressed by an objective function, which may incorporate a range of performance metrics. For linear-quadratic type problems the objective is an integral squared function, representing weighted state and control energies, which are minimized. In this thesis the objective function is extracted from a collection of simulated flight responses, using performance measures such as the steady state error and the integral of absolute error.

Evolutionary design procedures are often associated with searching within a design space which is a complex, nonlinear, and multimodal function of the design parameters. However, the search performance is not independent of an arbitrarily complex search domain. One of the guidelines for forming the objective is therefore to avoid so-called *needle in a haystack* problems. This is especially relevant for the design of the inner-loop controller as the vehicle is highly unstable and rapid failure occurs unless the appropriate control action is commanded. With an initially random set of solution vectors, there needs to be some beneficial discrimination to provide the selection pressure. If the cost of evaluating the objective were not so computationally expensive then this would not be such a problem. However, with many expensive flight simulations contributing to the objective function, it is necessary to encourage the growth of good control solutions by providing a path to the final performance goals. To achieve this, the multiple objectives used to evaluate the controller performance, are scheduled through weights expressed as a function of generation number. The avoidance of vehicle failure is given the highest priority at the start of the search.

While the performance measures reflect the desirable features of the controlled flight response, the objective must also encourage the development of a robust control solution. This is achieved with the inclusion of parametric uncertainty, disturbance, and signal noise in the flight simulation, and by assessing the controller performance over a large set of initial conditions covering the entire flight envelope. In the case of the inner-loop design the flight simulations are only performed for $t_f = 2$ seconds, with an emphasis on attitude maintenance and stability. Since the altitude response occurs over a much longer time scale, simulations of length $t_f = 30$ seconds were conducted for the outer-loop design.

Genetic algorithms are extremely opportunistic, and are therefore proficient at exposing failings in the relationship between the objective and the intended measure. This can be a frustrating way of exposing inadequacy in the reasoning behind the performance measures, but can also lead to a greater appreciation of the behaviour of the system. The remainder of this chapter details the formation of the initial condition set and the performance measures used to evaluate the controlled flight response.

5.2.1 Simulation Initialization

In addition to satisfying robustness concerns, the size of the initial condition set is also a reflection of the parameterization of the control function. When using the fuzzy controller, a single flight response may only activate a fraction of the complete rule base. To guarantee completeness in the fuzzy controller it is necessary to provided full coverage of expected variations in the controller input values. Performance and stability robustness can then be addressed by combining the input variable combinations with varied flight conditions and with a vehicle simulation corrupted by parametric uncertainty and signal noise.

There are two basic classes of simulation used in the design of the vehicle autopilot, with both being set by a single flight objective. For the inner-loop, the target is to trim the vehicle to an angle of attack (α_{cmd}), while the outer-loop target is simply to maintain the dynamic pressure (q_∞) of the desired flight trajectory. The complete initial condition set is formed by the vehicle flight condition, vehicle attitude, guidance command, and actuator setting:

$$\boldsymbol{x}'_{\boldsymbol{0}} = [h, V, m_f, (\alpha_{\text{cmd}}, q_\infty), \alpha, \theta, \dot{\theta}, \theta_e]^T \tag{5.3}$$

Six nominal operating points along the flight trajectory, are used to generate the vehicle flight condition $[h, V, m_f]^T$. Table 5.3 presents the selection of flight conditions along with the vehicle centre of mass (cm_x) and inertia (I_y). With the fuel tank positioned near the vehicle structural centre of mass, there is little variation in the centre of mass or inertia along the trajectory. The purpose of including different flight conditions is therefore to include the variation in the vehicle aerodynamics and propulsion performance along the flight trajectory. Each flight condition selected was allowed to be randomly perturbed within the following ranges,

$$[h, V, m_f]^T = \begin{bmatrix} h_i \pm 200\,\text{m}, \\ V_i \pm 200\,\text{m/s}, \\ m_{fi} \pm 100\,\text{kg} \end{bmatrix}. \tag{5.4}$$

Initialization bounds were also provided for the vehicle attitude, the guidance command, and the initial control setting. These are summarized in Table 5.4.

Table 5.3: Nominal flight conditions along a trajectory defined by $q_\infty = 188\,\text{kPa}$.

T_i	Velocity (m/s)	Altitude (m)	Fuel (kg)	$\text{cm}_x(m)$	I_y (kg/m^2)
1	2500	22420.4	2485	5.199	11187
2	3000	24745.0	2000	5.176	10941
3	3500	26682.3	1685	5.158	10804
4	4000	28450.0	1300	5.132	10655
5	4500	30000.0	985	5.107	10547
6	4900	31043.4	635	5.072	10437

Table 5.4: Initial condition bounds which also represent constraints on the perturbation of the vehicle from nominal operating conditions.

Parameter	Constraint
α	$\pm 5°$
α_{cmd}	$\pm 3°$
α_{err}	$\pm 3°$
θ	$\pm 5°$
q	0.1 rad/s
θ_e	$\theta_{\text{e,trim}} \pm 1°$

Four means of initialization were arranged, $IC_1, \ldots, IC_4$. The first allows user specification of the initial condition set and was primarily used for analysis of the controller after it had been configured. For the second method, all elements of the initial condition set are randomly generated within preset bounds. This method was used to experiment with varying the simulations throughout the design process, and for examining the robustness of the final control solution. The third method provided systematic coverage of possible α and α_{ref} combinations with 42 sets of initial conditions. Of the results presented in Chapter 6, this approach was predominately used. Finally, IC_4 was arranged for the design of the guidance control function. This was the only case where the flight objective was specified using the dynamic pressure of the nominal trajectory. Flight condition values for cases IC_2 through IC_4, were drawn from six predefined combinations of h, V and m_f defined in Table 5.3. For the guidance design (IC_4), it was necessary to limit the perturbation applied to the nominal flight conditions and the flight angle ($\gamma = \theta - \alpha$), to prevent excessive departure from the nominal flight trajectory.

5.2.2 Inner-Loop Performance Measures

The following performance measures are configured for a maximization task. Each is defined such that the maximum contribution to the overall objective is 100. The overall objective function is simply a weighted sum of m performance measures,

$$F_{\text{obj}} = \sum_{i=1}^{m} w_j \, J_i \,. \tag{5.5}$$

For all the results presented in Chapter 6 the test simulation length was $t_f = 2$ seconds.

Simulation completion:

The primary selection pressure early in the evolution of the controller is the establishment of a controller function which can at least prevent vehicle failure. To do this, simulation time is represented as a performance measure,

$$J_{t_f} = w_{t_f} \frac{t\,(\alpha < \alpha_{\text{lim}})}{t_f}, \tag{5.6}$$

where $w_{t_f} = 100$, and α_{lim} is an angle of attack limit marking vehicle failure.

Settled system status:

Following simulation completion, the next target is the reduction in the final state error. Two performance measures are used to evaluate the quality of the response after a time t_s, which was typically set to $t_f - 1$ seconds for the inner-loop design. These incorporate the desire for the angle of attack and pitch rate to be below a given tolerance. The contribution of each to the overall objective function is scheduled according to the generation number g relative to the length of the search N_G, allowing simulation completion to dominate during the early stages of design. For the angle of attack history:

$$J_\alpha = w_\alpha \left(1 - \frac{\mathbf{max}\left[\alpha_{\text{err}}\left(t > t_s\right)\right]}{\alpha_{\text{tol}}}\right), \tag{5.7}$$

where $\alpha_{\text{err}} = \alpha_{\text{cmd}} - \alpha$ and

$$\begin{aligned} w_\alpha &= \mathbf{min}\left(1, \left(\frac{2\,g}{N_G}\right)^2\right) \cdot 100 \\ \alpha_{tol} &= \begin{cases} 0.0523 \text{ radians} & \text{if } g < \frac{N_G}{2} \\ \mathbf{max}\left(0.00873, -0.0436\frac{2g-N_G}{N_G} + 0.0523\right) & \text{otherwise} \end{cases} \end{aligned} \tag{5.8}$$

Similarly for the pitch rate response:

$$J_q = w_q \left(1 - \frac{\mathbf{max}\left[q\left(t > t_s\right)\right]}{q_{\text{tol}}}\right) \tag{5.9}$$

where $q_{\text{tol}} = 0.2$ rad/s and $w_q = w_\alpha$. The reason for having the pitch rate tolerance set at a constant and relatively large value is that, with system noise, large pitch rates can be induced in the attitude response and dominate the overall objective. The weighting of both J_α and J_q prevents the generation of large negative performance measures during the early stages of design, when the vehicle fails.

Integrated absolute error:

The overall vehicle response is measured by the integration of the absolute error (IAE) versus time. Since there is no benefit in using IAE to discriminate against individuals early in the design evolution, it is gradually made more significant during the design. Applied to the angle of attack response,

$$\text{IAE}_\alpha = \int_0^{t_f} |\alpha_{err}|\mathrm{d}t. \tag{5.10}$$

The performance function $J_{\int\alpha}$ follows the same structure of the functions used for the settled system response,

$$J_{\int\alpha} = w_{\int\alpha}\left(1 - \frac{\text{IAE}_\alpha}{\text{IAE}_{\alpha,\text{tol}}}\right), \tag{5.11}$$

where

$$\mathrm{IAE}_{\alpha,\mathrm{tol}} = \frac{\alpha_{err}(t_0)t_s}{2} + \alpha_{\mathrm{tol}}(t_f - t_s)$$
$$w_{\int\alpha} = w_\alpha \tag{5.12}$$

The inclusion of α_{tol} in the reference response measure $\mathrm{IAE}_{\alpha,\mathrm{tol}}$ keeps the flight response demands inline with those for the settled system response. It represents an initial response rate of 0.1 rad/s following which an attitude maintenance tolerance of set by α_{tol}.

5.2.3 Outer-Loop Performance Measures

The performance measures for the guidance design follow the same format as those for the inner-loop, except the focus is the altitude response history. Though vehicle closed-loop stability is considered a prerequisite for the outer-loop design, system failure is possible through the growth of large altitude errors. For large enough departures from the nominal trajectory, the vehicle dynamics grow sufficiently different to those represented in the inner-loop design that the inner-loop fails. If the objective function for the guidance performance were simply stated as an integral error, the search procedure rapidly ascertains that by forcing the inner-loop to fail, it can provide superior performance measures. This is an example of the opportunistic nature of the genetic algorithm as a search procedure.

Two performance measures are used to assess the altitude response, with test simulations typically over the interval $[0, t_f = 30]$ seconds. Since the design of the guidance law is simply to determine two feedback gains, it is not necessary to schedule the performance measures.

Altitude response:

With the potential for large departures from the nominal trajectory (if the flight angle is not close to that required to follow the trajectory), it is difficult to establish a desired response to cover the initial stages. The altitude response measures therefore consider the response history after a time of 15 and 20 seconds, with the assumption that the vehicle should be tracking the nominal trajectory. The following performance function combines the integral of absolute error and the settled response measure,

$$J_h = 100\left[\left(1 - \frac{\mathbf{max}\left[h_{\mathrm{err}}\left(t > 20\right)\right]}{h_{\mathrm{tol}}}\right) + \left(1 - \frac{\mathrm{IAE}_h}{\mathrm{IAE}_{h_{\mathrm{tol}}}}\right)\right] \tag{5.13}$$

The integral error term is expressed in the same form as that used for the angle of attack response. A tracking tolerance of 50 m was typically applied and the integral evaluated over the time range [15, t_f] seconds.

5.2.4 Specific Objective Functions

The performance measure defined in the previous sections were applied in a number of ways. For example, with the design of the linear controller the scheduling of performance measures was considered unnecessary. Table 5.5 summarizes the specific objective functions used to design the linear controller, the fuzzy inner-loop controller, and the linear guidance function.

Table 5.5: Objective function specification. Tolerance values indicated refer to radians for angles and radians/s for angular rates.

OF_i	Application	Components
1	CGLF	J_{t_f}, $J_\alpha(\alpha_{\text{tol}} = 0.00873, w_\alpha = 1)$, $J_q(q_{\text{tol}} = 0.02, w_q = 1)$, $J_{\int \alpha}(w_{\int \alpha} = 1)$
2	CGLF	J_{t_f}, $J_\alpha(\alpha_{\text{tol}} = 0.00873, w_\alpha = 1)$, $J_{\int \alpha}(w_{\int \alpha} = 1)$
3	FC	J_{t_f}, $J_\alpha(\alpha_{\text{tol}}(g), w_\alpha(g))$, $J_q(q_{\text{tol}}(g), w_q = (g))$, $J_{\int \alpha}(\alpha_{\text{tol}}, w_{\int \alpha}(g))$
4	K_G	J_h

CHAPTER 6

Results - Controlled Hypersonic Flight

Aside from the acceleration capabilities of the scramjet vehicle, the most basic longitudinal performance characteristics refer to stability. Inherent stability for longitudinal flight relates to the dynamic stability of the perturbed longitudinal motion and the static stability. Static stability is typically expressed through stability derivatives describing the variation in forces and moments with respect to the vehicle state and control variables. For example, the sign of the stability derivative M_w (or M_α), which refers to the pitching moment variation with the angle of attack, determines the longitudinal static stability. With $M_w < 0$, a change in the angle of attack generates a restoring moment. For a fixed geometry vehicle, M_w is principally dependent on the location of the aerodynamic centre relative to the vehicle's centre of gravity. Without the need for manoeuvrability in the scramjet-powered launcher, it would seem desirable to have some degree of passive stability. However, the long inlets needed for scramjet operation and a large drag penalty associated with aerodynamic surfaces, makes inherent longitudinal stability an unreasonable objective. The inner-loop control system is therefore used to provide stability and guidance command tracking.

Dynamic stability is typically expressed by considering the sensitivity of the short period and phugoid poles to variations in the flight condition and vehicle attitude, following a linearization of the vehicle dynamic equations. Dynamic stability is not possible without static stability.

One of the features of the configuration studied in this thesis is the rate of system failure without any stabilizing control action. Here system failure is considered to be the vehicle reaching an angle of attack equal to the inlet angle. Beyond this point, the shadowed engine flow path would have no flow and it would be very difficult to return the vehicle to a smaller angle of attack. Figure 6.1 shows the establishment of trimmed flight conditions for the nominal vehicle, following which the controller is switched off (at t = 2 s). Growth in the control-free response is due to the non-equilibrium flight conditions altering the trim condition, through changes in altitude and velocity. If a small disturbance such as atmospheric turbulence is included in the simulation, then the time to failure is reduced to less than 0.5 s. The potential for rapid system failure, combined with

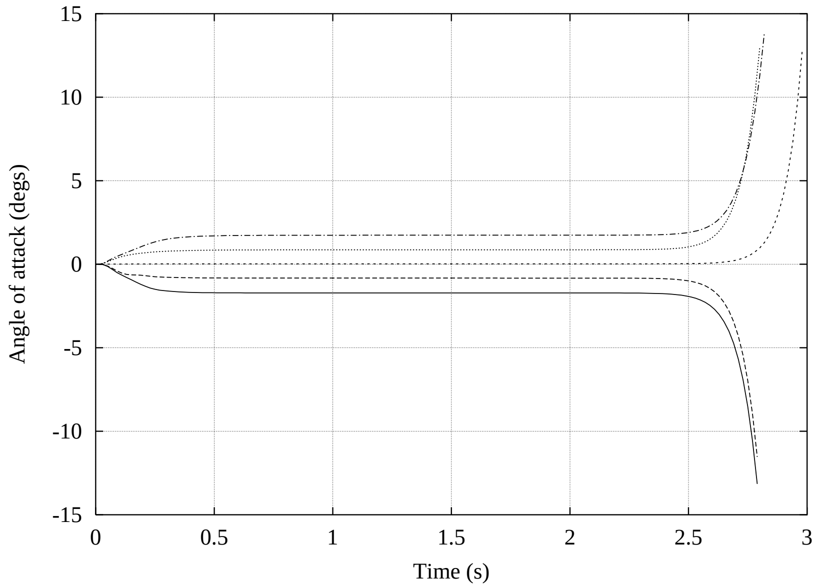


Figure 6.1: Control free stability following the establishment of various trimmed conditions. The controller is switched off at $t = 2\,\mathrm{s}$.

performance variation and disturbances, places great demands on the control system.

Previous chapters established the control structure, the flight simulation tools, and the fuzzy control definition and design procedure. The purpose of this chapter then, is to examine the capability of the evolutionary design procedure in configuring the control laws needed for stable altitude tracking of the scramjet vehicle across its hypersonic trajectory. Since the principal design problem was the specification of the longitudinal inner-loop controller, the results deal primarily with this aspect of the vehicle autopilot. The organization of the results seeks to address issues relating to the controller parameterization, the design procedure, and vehicle operation. These are as follows:

Control law parameterization: This is essentially a comparison of the design and performance of a constant gain linear controller and a fuzzy controller. While the greater design freedom available with a fuzzy controller enhances the general performance of the inner-loop controller, it also presents a considerably more demanding design problem.

Design procedure: This examination covers the issue of designing with an uncertain system model and the many variations available with the genetic algorithm and the design objective.

Vehicle operation: The control design results reveal operating characteristics such as: the necessary feedback variables; control sampling requirements; sensitivity to performance uncertainty, disturbances, and signal noise; and the broad range vehicle performance.

6.1 A Results Guide

The controlled flight examination begins with a comparison of a linear controller to a fuzzy equivalent which has undergone tuning. Due to the relative slowness of the Nelder-Mead procedure for high order problems, the tuning process is limited to a fuzzy controller with the minimum rule set. The comparison represents the primary control parameterization issue, namely the basic structure for describing the control function.

With the application of a fuzzy controller there are many additional parameterization issues due to the additional degrees of freedom available in defining the control function. Of these, input scaling and rule base size are addressed. The effect of an uncertain vehicle model on the control solution and the design procedure is also considered. Following the discussion of inner-loop controller, a guidance design is presented for the purpose of providing a full trajectory simulation. The results conclude with the considerations of issues relating to the performance of the genetic algorithm.

For the most part, three basic plots are used to represent the operation of the controller and the mechanics of the design procedure. To present the performance of the inner-loop controller, a series of step commands in α are issued. In most cases, with each step command the vehicle is also shifted along the trajectory by resetting the flight velocity, altitude, and fuel load, according to the nominal conditions provided in Table 5.3. This allows the broad range performance of the controller to be presented. In cases where a single flight condition was used, the trajectory index T_i refers to one of six flight conditions along the nominal trajectory. Further analysis of the controllers considers the flight response to a large set of randomly generated initial conditions (IC_2) and applies the performance measures of the integral of absolute error and the steady state error. For these tests, 500 simulations of length 4 seconds were performed, with a step command at 2 seconds. The number of completed simulation, N_C, represents the total number of tests minus those that failed due to the violation of the vehicle angle of attack limit. All the flight simulations are for the vehicle design specified in Chapter 3, and use the general flight dynamics equations formulated for a spherical, rotating Earth.

Two types of plots are used to represent the behaviour of the design procedure, using the measures of objective function and population entropy. The objective function describes the performance of the controller design and its history is averaged over the number of initial conditions used for each performance evaluation during the design. Further consideration of the objective function evolution looks at the relative contribution of the performance measures through the design process. Population entropy is a measure of the diversity of the population of potential control solutions, and is defined in Chapter 4.

6.2 Constant Gain Linear Feedback (CGLF)

The control function for a linear feedback controller is written as a linear combination of scaled state variables, using a gain matrix and a state vector. Evaluating the feedback gains typically follow the linearization of the system behaviour about a nominal operating point and applying optimal control theories such as the linear quadratic regulator (LQR) approach [211]. Due to the limitations of the linearized model in representing the nonlinear dynamic behaviour of the vehicle, performance and stability robustness can be generated by gain scheduling a set of controllers, thereby interpolating between locally optimal gains. It is generally accepted that the application of a linear state feedback controller to the hypersonic air-breathing vehicle would require scheduling against numerous operating variables, (h, V, α, θ_e) for example. For the flight envelope considered in this thesis, the potential complexity of this arrangement is significantly reduced. Broad range variations due to the changing flight condition are tempered by flying along a constant dynamic pressure trajectory. Further, the vehicle attitude constraint of a few degrees means the nominal vehicle behaviour at a given flight condition is likely to be reasonably linear.

Robust control theory extends the capability of linear feedback by providing a means for representing performance uncertainty in the design process. Following the thoughts in the previous paragraph, it is therefore considered reasonable that, by applying a design procedure based on the full nonlinear vehicle behaviour across the flight envelope, a functional constant gain linear feedback (CGLF) controller can be established. The design of the CGLF controller provides a benchmark for the additional complexity (degrees of freedom) associated with designing a fuzzy controller (FC). Applying the general (GA) design procedure, set out in Chapters 4 and 5, allows parametric uncertainty and signal noise to be included in the evaluation of the controller performance. With a design dimension of three, and a genetic search procedure, the design procedure is the archetypal brute force approach. The design based on the nominal vehicle model used the genetic algorithm and objective function specifications (GA_1, OF_1), see Chapter 5.

Due to the relative simplicity of the design task, the linear control example was used to examine possible input arrangements and the control sampling time required. The input combinations tested included, $[\alpha_{err}, \dot{\theta}]$, $[\alpha, \alpha_{\text{err}}, \dot{\theta}]$, $[\alpha_{\text{err}}, \dot{\theta}, \theta_e]$, and $[\alpha_{\text{err}}, \dot{\theta}, \theta_{e,\text{err}}]$, of which only $[\alpha_{\text{err}}, \dot{\theta}, \theta_{e,\text{err}}]$ proved capable of forming a useful control law. Not all of the input combinations were trialled for the fuzzy controller, though, from the ones tested, $[\alpha_{\text{err}}, \dot{\theta}, \theta_{e,\text{err}}]$ was generally superior. It was also noted that improved performance was possible with more states included in the feedback loop, however since one objective of the linear design was to establish relative gain sizes for the fuzzy controller, the states used were limited to the attitude and control variables.

With regards to the control sampling time, closed loop stability could be achieved for sampling times (Δt_c) up to 0.04 s. This is, of course, a reflection of the actuation capabilities, the vehicle operating constraints, and the inherent dynamics of the vehicle. All the results in this chapter use a control sampling time $\Delta t_c = 0.02\,s$.

Feedback gains were sourced from the range, $K_i \in [-50, 50]$. For the nominal vehicle model, the following feedback vector was generated,

$$\boldsymbol{K} = [44.887, \quad -5.731, \quad -17.287].$$

Figure 6.2 shows the control response of the nominal vehicle to a series of step changes in the reference angle of attack (α_{ref}). The main obstacle in generating the feedback gains comes from the approximation of the trim elevator condition, which is used to determine $\theta_{e,\text{err}}$. Since the trim condition follows a predefined function of angle attack, averaged over the entire trajectory, the controller represents a best fit to the trim uncertainty. With only three inputs, the maintenance of a steady state response where the pitching rate is zero requires either $\alpha_{\text{err}} = 0$ and $\theta_{e,\text{err}} = 0$, or that $K_\alpha \alpha_{\text{err}} = -K_{\theta_e}\theta_{e,\text{err}}$. Consequently, errors in the trim estimate are transferred to a steady state attitude error. The effect of trim uncertainty also means the controller is sensitive to the flight condition variation along the trajectory, as shown in Figure 6.3. The oscillations that appear in Figure 6.2 after 6 s, are due to the non-tracking of the nominal flight trajectory which places the vehicle several kilometres off-course. Since unchecked altitude travel is not desirable (in terms of the vehicle loads and engine performance), the initial condition variation used to assess the controller performance during the design does not include these large variations. Further departure from the nominal trajectory ultimately leads to the failure of the vehicle. Though the performance robustness of the controller is limited by the trim errors, the gains appeared sufficient to provide stability robustness.

With the inclusion of system uncertainty and signal noise in the design, all the control gains are increased in magnitude relative to the nominal case,

$$\boldsymbol{K}_{\text{unc}} = [48.816, \quad -8.639, \quad -19.44].$$

The main difference in the design setup compared to that for the nominal system, is the removal of the pitch rate penalty from the objective function (OF_2). Noise in the vehicle performance can generate relatively large pitching rates, which can overwhelm the objective measure if the small tolerance from the nominal design is applied. Figure 6.4 compares the flight response using the two gain settings $\boldsymbol{K}$ and $\boldsymbol{K}_{\text{unc}}$ in the feedback loop. The design path for the uncertain system model generates a more gradual response, with smaller angle of attack oscillations than that provided by $\boldsymbol{K}$. If $\boldsymbol{K}_{\text{unc}}$ was to be applied to the nominal vehicle, the oscillations which appear between six and eight seconds in Figure 6.2 are no longer seen.

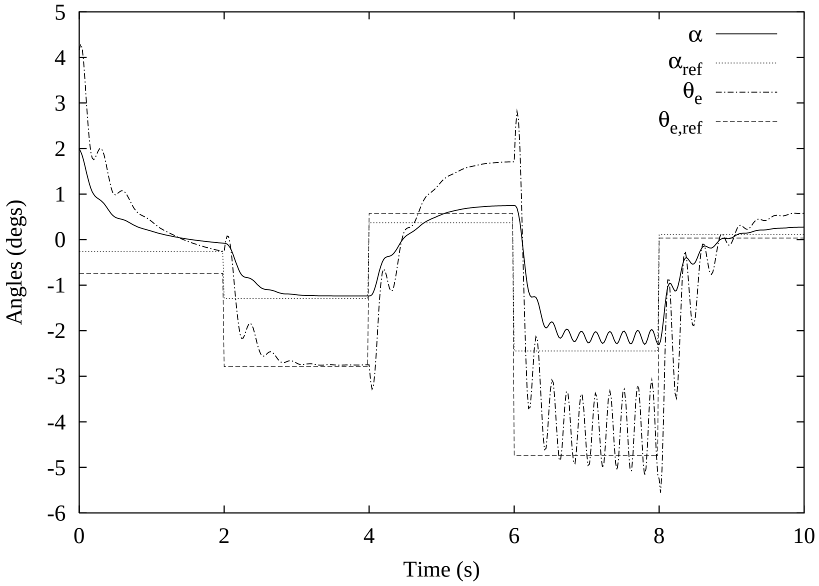


Figure 6.2: Nominal vehicle response to a series of step commands in angle of attack, using the CGLF controller. The flight condition corresponds to T_5, where $V_0 = 4500\,\text{m/s}$.

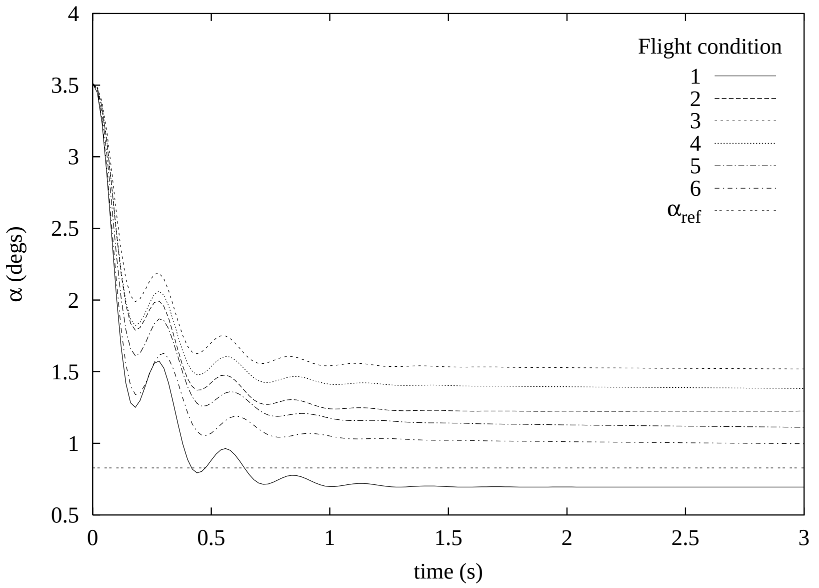


Figure 6.3: Sensitivity of the linear controller ($\boldsymbol{K}$) to the flight condition. The flight conditions T_i are defined in Table 5.3, and represent a sampling of operating points along the nominal trajectory.

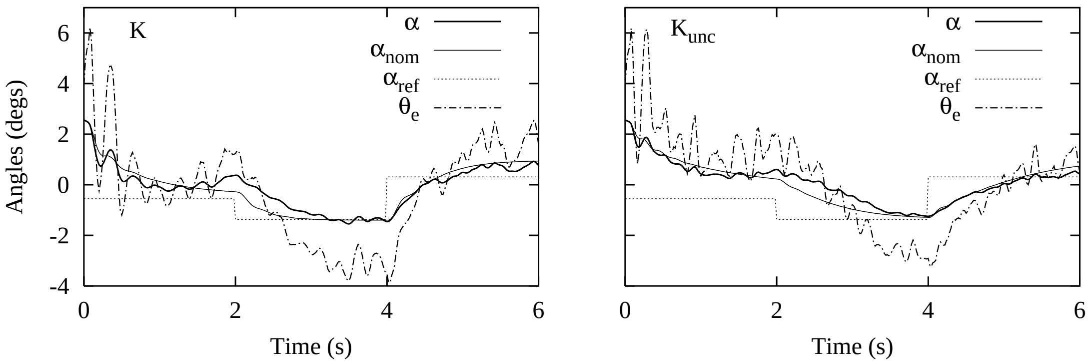


Figure 6.4: CGLF ($\boldsymbol{K}$ and $\boldsymbol{K}_{\text{unc}}$) response to performance uncertainty and sensor noise. For each gain set, a thin line is used to show the angle of attack response of the nominal vehicle model.

6.3 Fine-Tuning the Linear Fuzzy Controller

One of the mechanisms for generating a nonlinear fuzzy control law is to fine-tune an established linear control law. For the three input variables used in the linear controller, an exact fuzzy control copy of the linear controller within the angle of attack constraint, can be realized with a rule base containing 27 rules, see Table 5.2. Uniform variable coverage is provided by 50 % overlap amongst neighbouring partitions. This represents the minimal discretization of the input space. To preserve the relative contributions of the linear controller inputs to the control function, the input and output scaling factors used to normalized the fuzzy variable domains, must be scaled in proportion to the feedback gains.

$$[k_\alpha, k_q, k_{\theta_e}] = a\left[\frac{1}{K_\alpha}, \frac{1}{K_q}, \frac{1}{K_{\theta_e}}\right] \tag{6.1}$$

where a in this case is used to spread the rules over the full angle of attack range. It is also necessary to set the output scaling in accordance with the linear gains,

$$k_u = \boldsymbol{k} \cdot \boldsymbol{K}^T \tag{6.2}$$

where $\boldsymbol{k}$ is the scaling vector for the fuzzy controller. Using the nominal gains for the CGLF controller, the input and output scalings for the fuzzy controller are as follows,

$$\begin{aligned} \boldsymbol{k} &= [0.05236, \;\; 0.41, \;\; 0.1364] \\ k_u &= 7.051 \end{aligned} \tag{6.3}$$

These values mean the control surface formed by the fuzzy rule base is bound by an angle of attack error of 0.05236 radians, a pitch rate of 0.41 rad/s, and an elevator trim error of 0.1364 radians. Though the output scaling states that an individual rule can generate an elevator actuation command of 7.051 rad/s, the maximum actuation rate returned by the controller is limited to 2.0 rad/s.

Once the input and output scalings are defined, the linear controller can be transferred to an arbitrarily large fuzzy rule base. The possibilities for fine-tuning the rule base extend to all degrees-of-freedom available, but only tuning of the rule consequents is considered here. A Nelder-Mead procedure (described in Section 4.3) was used to fine-tune the vector of rule outputs $\boldsymbol{y}$. The relative slowness of the procedure in terms of the number of objective function evaluations, meant that for large rule bases, the tuning process could require as much computation as that needed to evolve the rules without prior knowledge. For example, 3000 objective function evaluations were used to tune 27 rules. After 7000 function evaluations for a 125 rule case the objective was equivalent to the 27 rule case, but the 24 hours of computation time used is roughly the same as that need to evolve a superior rule base from scratch. Consequently, only the results for the fuzzy controller with 27 rules are presented.

An additional limitation on the Nelder-Mead application is the need to provide consistent objective function evaluations so that the local gradient of search space is itself consistent. The inclusion of performance variation through uncertain system parameters would require the time history of inputs to the uncertainty filters to be fixed during the fine-tuning process, but this has not been considered. The objective function is the same form as that used for designing the linear controller.

The performance advantage available with the fuzzy controller is due to the capacity of the FC to allow local manipulation of the control surface. Figure 6.5 shows the control surface before and after fine-tuning of the linear controller. Each surface maps the normalized inputs (α^*, q^*) to the normalized control command (u^*), for a given elevator trim error (θ_e^*). The capping of the output to ± 0.3 represents the control command constraint of 2 rad/s. With so few partitions for each input variable, there is limited scope for manipulation of the control surface. The tuned rule base provides high actuation rates over a larger range, thereby mitigating the error in the trim condition. While the response rate for large attitude errors remains the same as the linear case, the additional control authority for smaller errors aids the reduction of the steady state error.

The benefits of manipulating the control surface are shown in Figures 6.6, as a reduced sensitivity to the variation in vehicle performance with flight condition. In Figure 6.7 the tuned fuzzy controller is compared to the CGLF controller using a series of step commands. The flight condition was chosen since it highlights both the benefits and dangers with fine-tuning the fuzzy controller. While the large actuation rates available with the FC provides a faster approach to the settled response, the downside is the potential for significant overshoot. Without an overshoot penalty in the objective function, this feature is allowed because it is not a large component of the integral error function and it improves the steady state cost function. A further limitation of the design is that, without

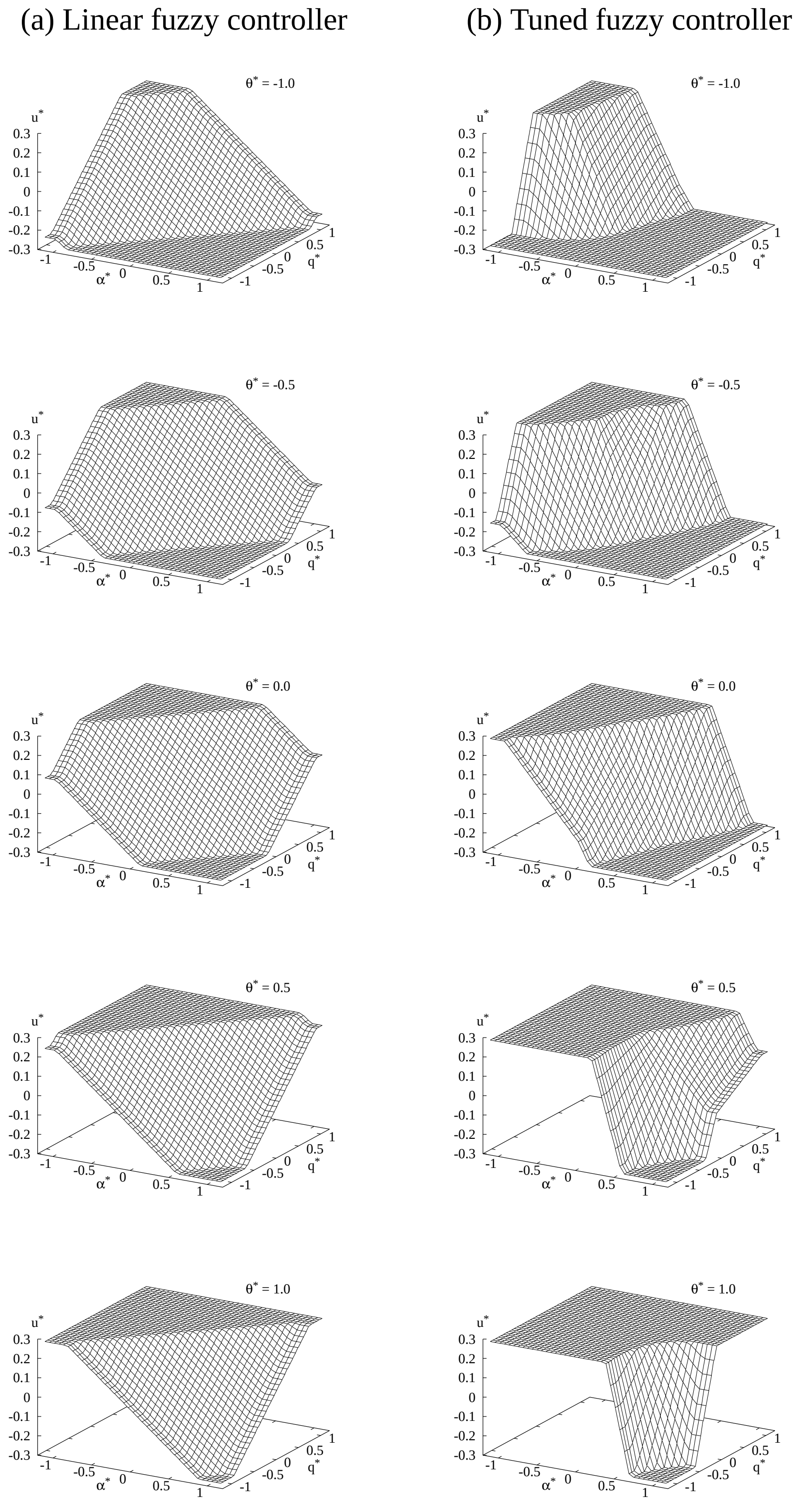


Figure 6.5: Tuned control surfaces.

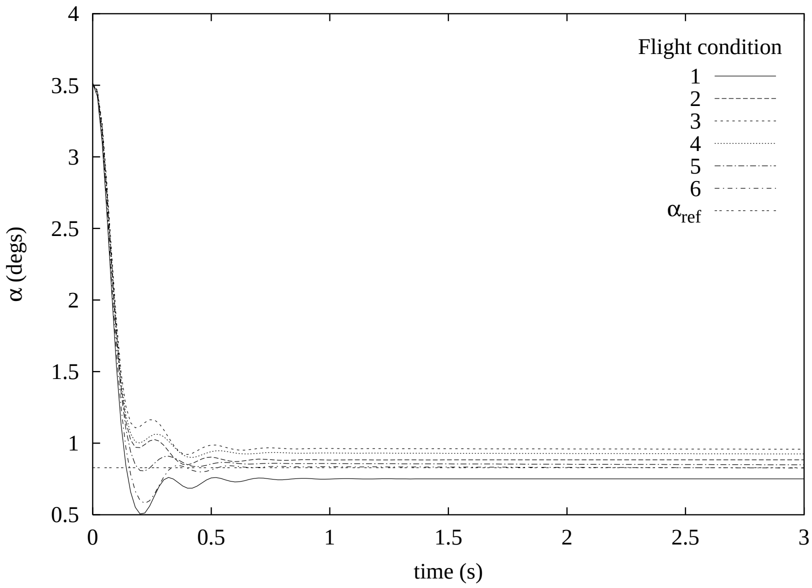


Figure 6.6: Sensitivity of the tuned 27 rule fuzzy controller, to the flight condition.

any performance variation through uncertainty, the controller can be tuned more precisely to the collection of test conditions provided. If the test conditions do not provide sufficiently large angles of attack, then control inputs exceeding the scaling bounds can lead to system failure. The first step response of Figure 6.7 is an example of a response which may lead to an unrecoverable position.

The fuzzy rules used for attitude controller are bound by the input scalings used to normalize the variable domains. Regardless of how large the input signals to the controller are, the maximum contribution of a single rule is set by its consequent value. One of the dangers in fine-tuning the rules is that the boundary condition may contribute to instability if the inputs exceed the scaling of the input variables. Now, part of the avoidance of this situation comes from the selection of the input scalings. With this in mind the angle of attack range used in the optimizer's test condition set must extend beyond the boundary.

A summary of the performance of the linear and fuzzy controllers is provided in Table 6.1. The performance measures include the number (N_C) of completed simulations out of the test set of 500, and the average integral error and settled response error for the simulations which did not lead to vehicle failure. It is clear that the tuning process based on a nominal system model leads to a loss in robustness of the fuzzy controller. This is also a reflection of the bounded nature of the FC and the design would likely be improved by increasing the input scaling parameters, adding an overshoot penalty, or by extending the design set of initial conditions. However, from the set of simulations that were completed the FC provided much improved response characteristics over both linear controllers.

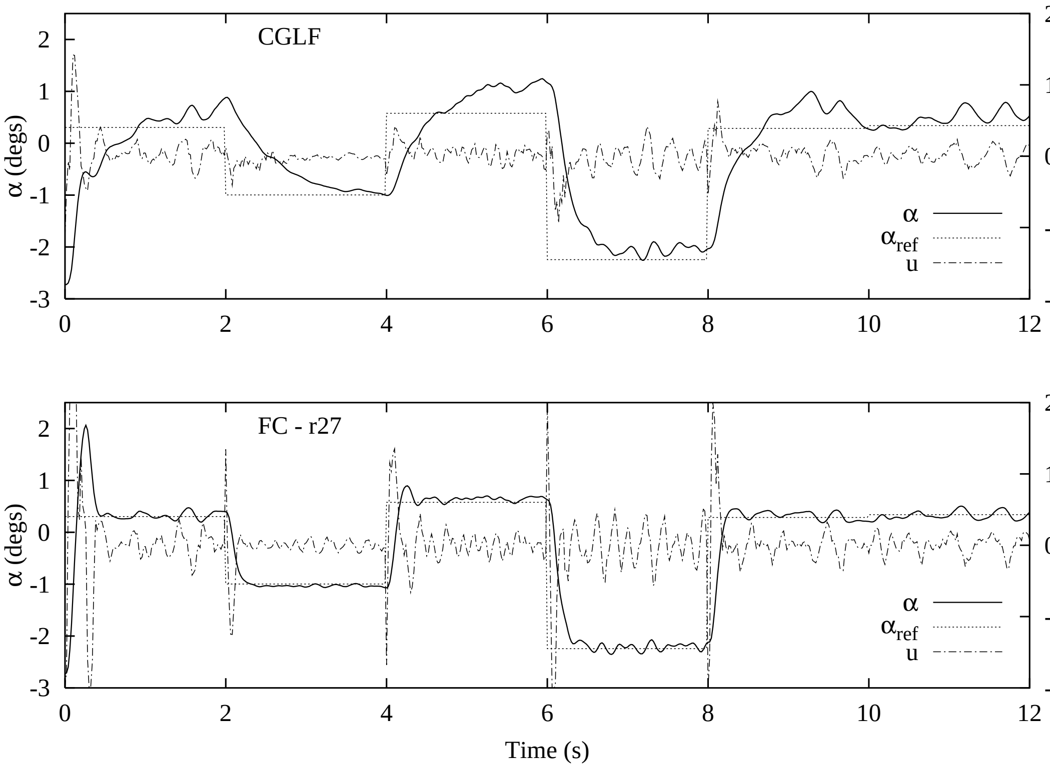


Figure 6.7: Improved trimming of the vehicle with the tuned 27 rule fuzzy controller (FC_{27}), compared to the linear controller (CGLF). The vehicle model includes parametric uncertainty and signal noise and the flight condition is T_5.

Table 6.1: Performance assessment for the CGLF controller and the tuned fuzzy controller over 500 test simulations. N_C refers to the number of completed simulations.

Controller	N_C (/500)	$\int \lvert\alpha_{\text{err}}\rvert \mathrm{d}t$ (rad.s)	$\alpha_{\text{err}}(t_s)$ (degs)
$\boldsymbol{K}$	499	0.03861	0.701
$\boldsymbol{K}_{\text{unc}}$	497	0.04539	0.719
FC_{27}	450	0.01640	0.325

6.4 Evolutionary Design of the Inner-Loop FC

The previous section demonstrated the worth of the fuzzy rule base, provided care is taken in the specification of the input gains and the range of input conditions used to evaluate the performance. In this section the evolutionary design procedure is applied directly to the fuzzy controller, generating the control parameters without any prior knowledge. The results address a number of issues relating to the design procedure:

- Design with uncertainty addresses the effect of an uncertain vehicle model on the control solution and the evolutionary process.
- The mechanics of evolution examines the process by which useful control laws are evolved from a random population of potential solutions.
- To evaluate the impact of the level of discretization of the rule base, the design of fuzzy controllers ranging in size from 27 to 225 rules is examined.
- An extended design case is considered whereby the input scaling parameters are designed simultaneously with the rule consequents.

Design with Uncertainty:

To address the impact of system uncertainty on the evolution of the controller, the design of a fuzzy controller with 125 rules (FC_{125a}) is presented. The rules are established by discretizing each fuzzy input variable with five uniformly distributed partitions. The design follows the specification (GA_3, OF_3), established in Chapter 5. Figure 6.8 compares the performance on the uncertain vehicle model, of the controllers designed without (FC_{125}) and with uncertainty ($FC_{125,\text{unc}}$) present in the system model. In the same manner as the design of $\boldsymbol{K}_{\text{unc}}$ for the CGLF controller, the presence of uncertainty in the design objective function results in a more conservative (slower) angle of attack response with a small penalty in the steady-state error. It also appears that the characteristic frequencies of the performance uncertainty models and the signal noise are manifested through an elevator command history of higher frequency.

The overall performance of the two controllers are summarized in Table 6.2. In general, the controller designed with uncertainty offers greater robustness. The rapid response associated with FC_{125} leads to failure of the vehicle when exposed to large angle of attack errors. If the initial condition used in Figure 6.7 was applied, then failure occurs within 0.4 s. Though less frequent in the $FC_{125,\text{unc}}$ design, failure also occurs for extreme initial conditions. While the direct cause is likely a combination of signal noise and large angle of attack errors, it is also a result of the design process being exposed to a fixed arrangement of initial conditions. Certain extreme combinations of attitude and control values can, with performance uncertainties, lead to failure.

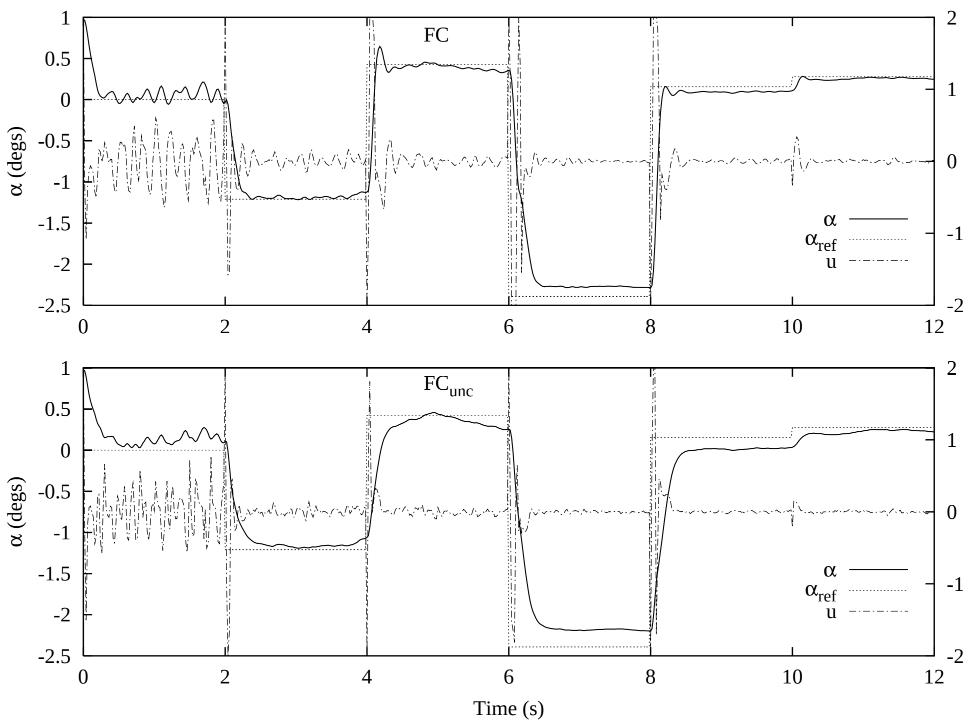


Figure 6.8: Response of 125 ruled fuzzy controller with an uncertain system model. The top plot shows the angle of attack and elevator command response for the controller designed for the nominal vehicle model. In the bottom plot the controller was designed with an uncertain vehicle model.

Due to the many more degrees of freedom in designing the fuzzy controller, it is much more susceptible than the linear controller, to conditions unseen during the design. The reasoning is that the additional degrees of freedom can provide a more precise match to a fixed set of initial conditions, with precision leading to reduced generalization and, consequently, robustness problems. Support for this argument can be seen in the greater stability available in the evolved design of a rule base with 27 rules, which is also shown in Table 6.2. While the response characteristics of the 27 rule set are mostly inferior to the large rule bases, the extra generality required of each rule augments the controller robustness. For the larger rule bases, once solutions are established in the population which prevent vehicle failure across the majority of test, there are likely to be a number of unused fuzzy rules. When exercised against more varied test simulations, these untuned rules can contribute to undesirable control commands. Since simulation failure was generally the result of actuator overshoot, it is felt that a series of step responses would provide a more appropriate assessment on performance during the design.

Figure 6.9 shows the evolution of the objective function during the design of FC_{125} and $FC_{125,\text{unc}}$. Oscillations in the $FC_{125,\text{unc}}$ trace are due to the inconsistent evaluation of the best solution from one generation to the next. The most important feature, in terms of the performance of the genetic algorithm, is that the growth profile for $FC_{125,\text{unc}}$ follows the same form as FC_{125}. The transition at generation 250 is due to the objective com-

Table 6.2: Performance comparisons of controllers designed on the nominal vehicle model and the uncertain vehicle model. A total of 500 tests simulations were run over 4 seconds, providing 1000 step responses. The performance measures are averaged over the simulations (N_C) which did not fail.

Controller	RCGA	Vehicle	N_C (/500)	$\int \lvert\alpha_{\text{err}}\rvert \mathrm{d}t$ (rad.s)	$\alpha_{\text{err}}(t_s)$ (degs)
FC_{125}	GA_3	Nominal	485	0.01124	0.047
	”	Uncertain	453	0.01550	0.248
$FC_{125,\text{unc}}$	”	Nominal	500	0.01635	0.089
	”	Uncertain	497	0.02371	0.282
$FC_{27,\text{unc}}$	”	Uncertain	500	0.02357	0.363

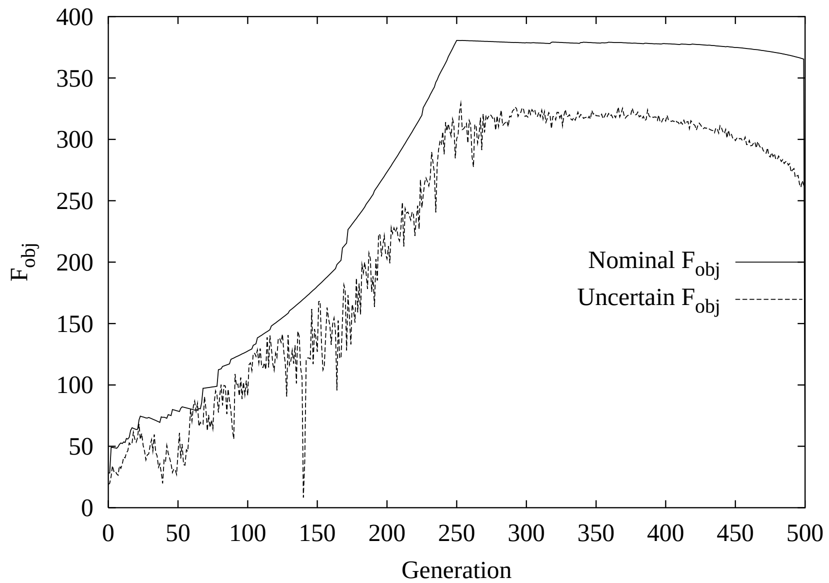


Figure 6.9: Evolution of the objective function with and without system uncertainty. F_{obj} refers to the average response for the best solution at each generation.

ponent weightings reaching their maximal value of one. Over the later generations, the gradual decrease is due to the continued decrease in the angle of attack tolerance, placing greater demands on reducing the steady state error. For both designs, the population is well established with *good* control solutions by generation 250. An indication of the quality of the control solutions present in the population is revealed in the reduction of the objective function noise for the $FC_{125,\text{unc}}$ profile.

The fuzzy control solutions generated by the genetic algorithm demonstrate the enhancement of controller robustness with the inclusion of uncertainty in the vehicle model. Having made the argument for the benefits to stability robustness, all remaining results relate to the controller design using the uncertain system model.

Mechanics of Evolving Control Solutions:

One of the remarkable attributes of evolutionary design is to see the rapid development of good control solutions from an initially random set of possible solutions. The 125 ruled fuzzy controller (FC_{125}) is again used here, to illustrate the evolutionary process. The selection (GA_4, OF_3, FC_{125}) defines the design configuration.

Figures 6.10 and 6.11 show the evolution of the controlled flight response using a set of step responses at the trajectory point T_3, corresponding to flight at 3500 m/s. The first series (Figure 6.10) covers the best solution available from generations 0 to 100, while the second (Figure 6.11) covers generations 120 through to 500. As expected with a random initial population, even the best solutions available from the initial population lead to rapid vehicle failure. However, through the bonus available in preventing vehicle failure, there is a general lengthening of the responses, which ultimately leads to the development of closed-loop stability. From this point on, the search is performing the function of fine-tuning the control surface to better satisfy the desired response characteristics. It is worth noting that, for this small set of initial conditions, there is practically no difference in the response history from generations 300 to 500. In general, by the half way point (g=250) of the evolutionary search, the control solution has been well established.

The reason for using a non-uniform objective function to direct the search, was to promote the rapid evolution of desirable response characteristics. Using OF_3, the four performance measures where scheduled according to the relative stage of the search (g/N_G). In Figure 6.12 the contributions of the various performance measures to the overall objective are displayed for the best solution at every 10^{th} generation. Across the first 100 generations, the J_{t_f} measure dominates the objective function, resulting in the rapid rise of the robustness of the control solution. By the time the majority of test conditions satisfy the simulation time measure, the remaining performance measures lead to a rapid rise in the quality of the solution. As was previously noted, the drop in the values of J_α and

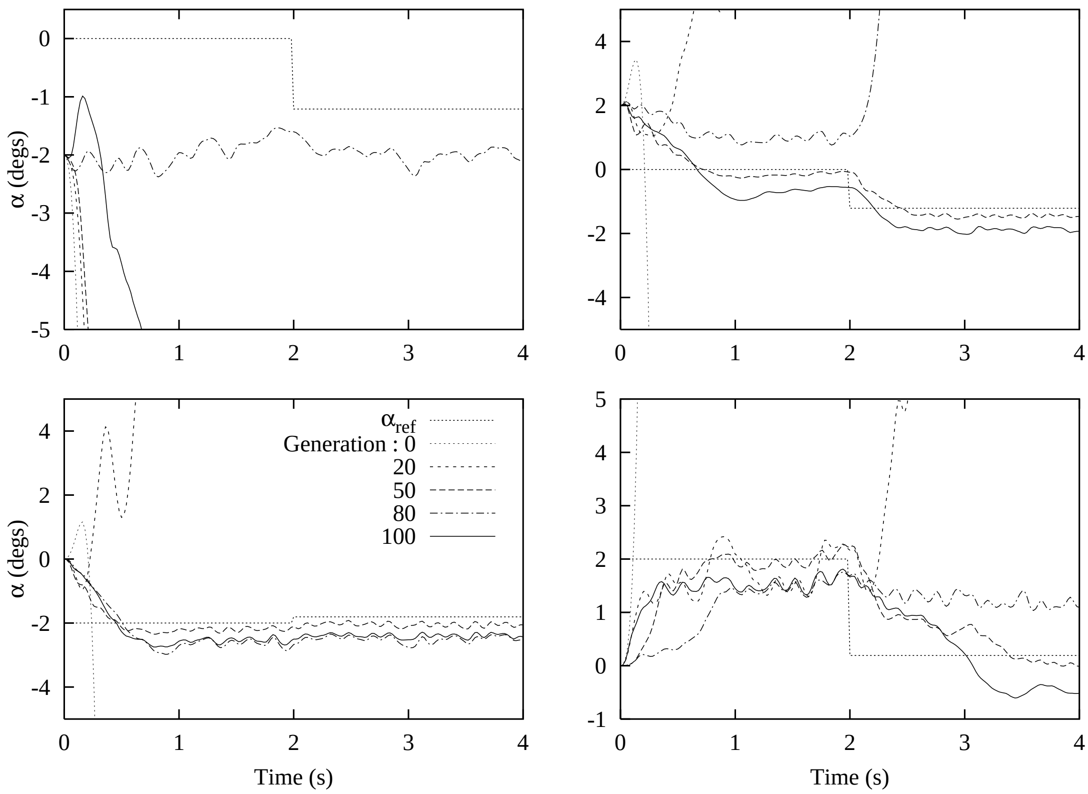


Figure 6.10: Evolution of the controlled flight response over the first 100 generations. The total generations used in the search is 500.

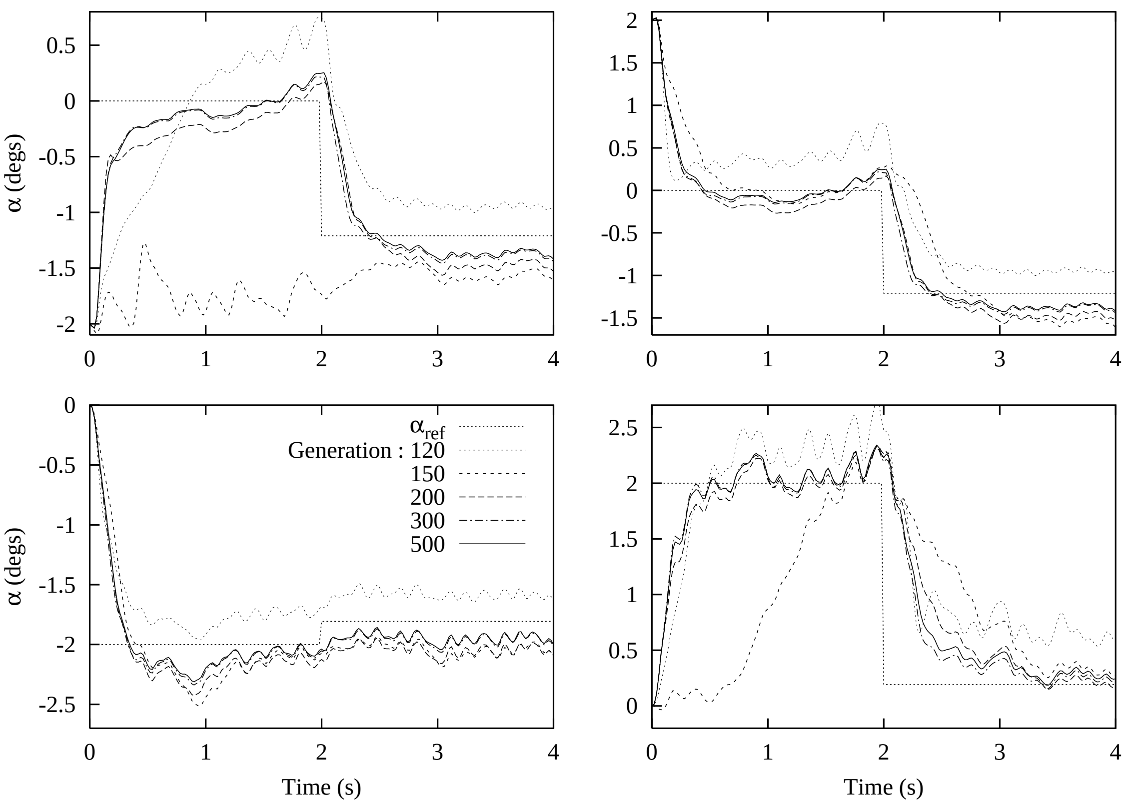


Figure 6.11: Evolution of the controlled flight response from generation 120 through to 500. The total generations used in the search is 500.

$J_{\int\alpha}$ in the later half of the search reflect the establishment of near optimal solutions by generation 250 and a reduction in the tolerance for angle of attack error (α_{tol}).

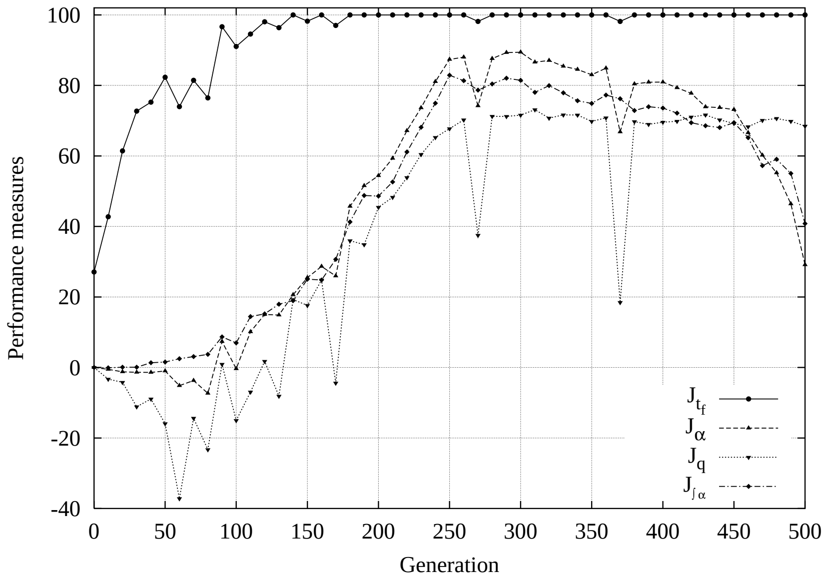


Figure 6.12: A history of performance measures during controller evolution. The best control solution is sampled every 10 generations, and evaluated over the design set of initial conditions (IC_3). Each performance measure is averaged across the initial condition set.

To check that the design set of initial conditions provides sufficient coverage for the development of a robust controller, the best control solution available after every 10^{th} generation is tested against a large set of random initial conditions. The comparison is shown in Figure 6.13. In general, the performance comparison supports the use of IC_3 initial condition set for the design. Most notable is that, in terms of the settled response, by generation 300 the available control solution is effectively equivalent to the final solution. Contributing to this search feature is the characterization of the mutation operator with $\beta = 4$ (see Appendix A), which provides a very localized variation after the half-way point of the search. After the controller solution has managed to prevent vehicle failure across the full set of test conditions, there is a rapid reduction in the final angle of attack error, which is then maintained through to $g = N_G$. While the rise in completed simulations for the IC_2 set closely follows that for the design set, it is only for the control sample at generations 200 and 210, that the full compliment of IC_2 are fully executed. Though hesitant in reading too much into this single design example, a possible cause is that further refinement of the control solutions against a fixed set of test conditions results in a loss in generality.

A final point of interest is the topology of the control surface for the evolved control solution. Figure 6.14, for the 125 rule FC can be compared with Figure 6.5 for the 27 rule FC derived from the linear controller. Features from the tuned 27 rule FC and the

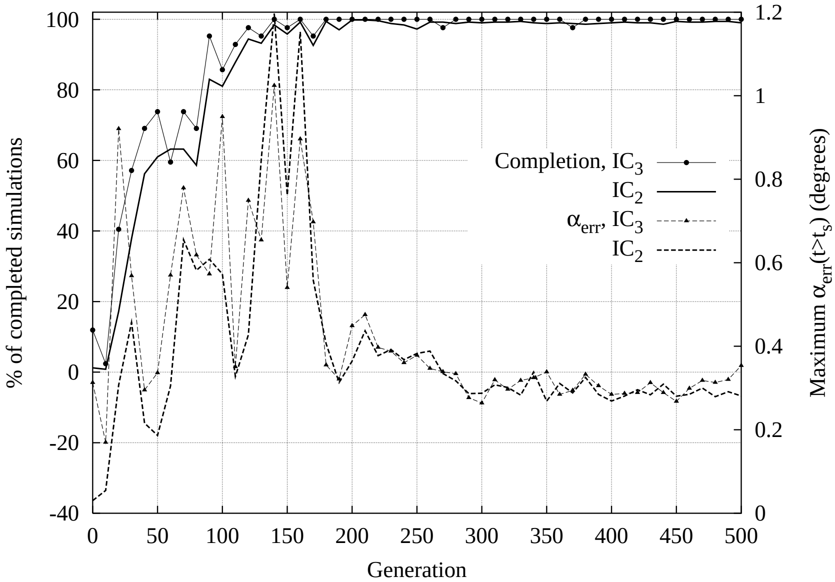


Figure 6.13: Comparison between the evolution of the design set of initial conditions to the performance on a larger random set. The best control solution is sampled every 10^{th} generation, and evaluated over the design set of initial conditions (IC_3) and 500 random initial conditions (IC_2). Each of the IC_2 tests involved a 4 s simulation with a step command after 2 s, with the α_{err} measure taken for the second step.

original linear controller appear in the evolved solution. With the advantage of additional degrees of freedom, the evolved solution provides greater manipulation of the gradient surfaces. The more complicated surface may, in part, be due to spurious rule consequents which exist since some rules in rule base of high dimension are likely to not impact on the controlled response.

Rule Base Size:

For a given application the optimal number of fuzzy control rules is influenced largely by the desired performance and the means of constructing the controller function. This section considers the evolution of rule bases ranging in size from 27 rules to 225 rules and, in so doing, assesses the flight response performance of the control designs and the capability of the genetic algorithm to configure both small and large design spaces.

Table 6.3 summarizes the results of the design experiments, based on the controller performance over a set of 500 randomly generated initial conditions. The expectation is that, with greater partitioning of the input variables, the controller can better provide for the varied demands across the range of input space. There is also the possibility to better deal with uncertainty in the trim estimate which forces a smaller rule base to compromise on the performance. With reference to the simulated response of selected designs shown in Figure 6.15, there is a general improvement in the angle of attack response, due mainly to a reduction in the error of the settled responses. Though the solution quality is relatively

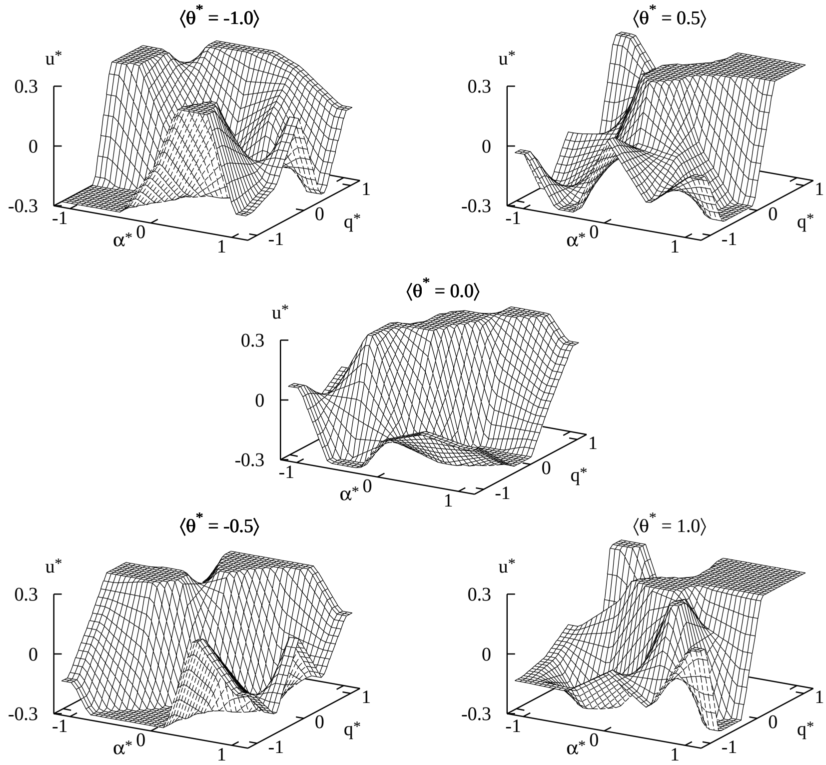


Figure 6.14: Control surface topology for the evolved 125 rule fuzzy controller.

poor for the 27 rule design, it also the only configuration which did not generate any vehicle failures across the set of test simulations. The need for greater generalization of the rules in the case of FC_{27}, improves the robustness of the controller relative to the limitations of the initial condition set and the performance measures.

Table 6.3: Quantitative comparison of controller performance for difference sized rule bases.

Rules	RCGA	N_C (/500)	$\int \lvert\alpha_{\text{err}}\rvert \mathrm{d}t$ (rad.s)	$\alpha_{\text{err}}(t_s)$ (degs)
27	$GA_3(\beta = 2)$	500	0.02357	0.363
125	$GA_3(\beta = 2)$	497	0.02371	0.282
125	$GA_4(\beta = 4)$	498	0.02047	0.271
225	$GA_4(N_p = 30)$	498	0.02093	0.242
225	$GA_6(N_p = 50)$	499	0.01799	0.193

The different settings used for the genetic algorithm (indicated in Table 6.3), reflect the needs of the evolutionary search to achieve the full potential available with the specific controller definition over a search length of 500 generations. The evolution of the objective function and population entropy, as shown in Figure 6.16 reveal a number of features of the search procedure. Not surprisingly the smallest rule base (FC_{27}) has the fastest initial growth in the design performance. As the rule base size increases the search time

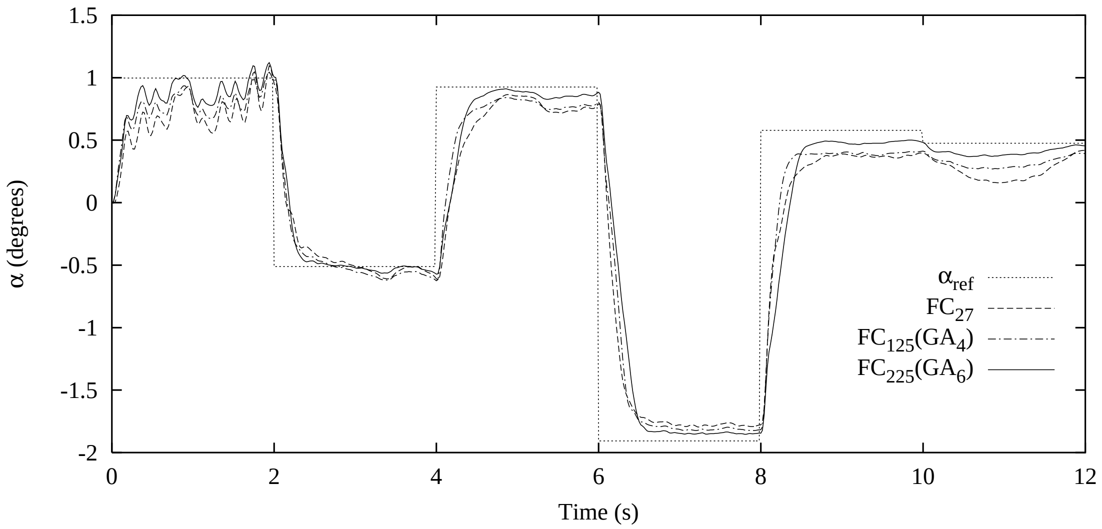


Figure 6.15: Comparison of controlled attitude response for rule bases ranging in size from 27 rules to 225 rules.

required to match the quality of the (FC_{27}) also increases. This turns out to be the critical element in terms of fully realizing the potential of a given controller definition. Comparing the two traces for FC_{225}, the larger population example (GA_6) provides a more rapid improvement in the solution quality over the early generations which ultimately generates a superior control solution. With average mutation magnitude decreasing across the generations, the search benefit of mutation relies on the early development of good control solutions. Population size is not the only factor influencing the growth of the objective function however. For example, if the FC_{125} configuration were designed using a mutation strategy parameter of $\beta = 2$, the more disruptive mutation delays the growth in the objective and the final design only just manages to match the 27 rule design.

The noise present in the objective function traces is due to the representation of uncertain features in the vehicle model. The magnitude is coupled to the quality of the control solution and the population convergence. For the $FC_{125}(GA_3)$ and $FC_{225}(GA_6)$ designs, the delayed growth in the solution quality means that the noise persists further into the search, until the point where the performance of the control solutions are less susceptible to modelling noise.

Input Scaling Design:

For the control designs presented so far the input scaling values have been derived from the gains generated by the design of the linear controller. The main reason for this was the removal of three design variables which impact on the global features of the fuzzy control function, and could potentially disrupt the search. It is reasonable to expect however, that tuning the input scalings for the particular rule base configuration should improve the

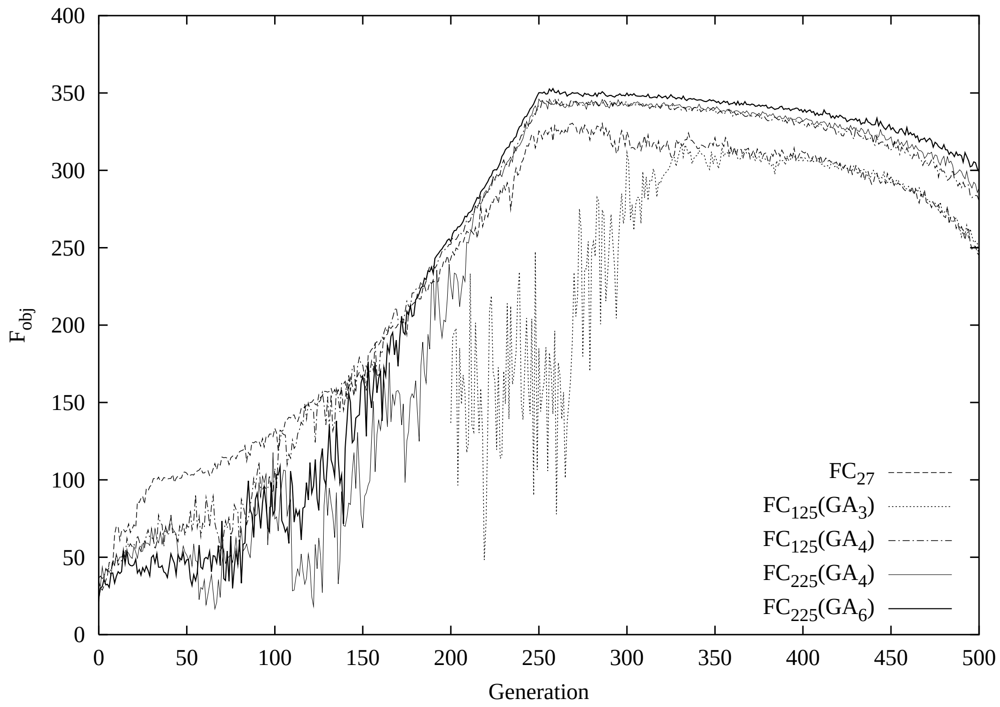


Figure 6.16: Evolution of the objective function and population entropy for controllers of size between 27 rules and 225 rules. For clarity, the $FC_{125}(GA_3)$ trace begins at the 200^{th} generation.

utilization of the discretization of the input space. This type of experimentation is often extended to the complete design of all features of the rule base, including fuzzy variable definitions, individual rule structure, and the overall combination of rules [103, 136, 224]. While some of these possibilities were tested, the assessment requires additional work and is not presented here. Table 6.4 summarizes the results for the design of the input scalings $\boldsymbol{k}$ and the output array $\boldsymbol{y}$ for controllers with 125 rules and 225 rules.

Table 6.4: Overall performance of controller with rules and input scaling as part of the design. The input scaling values bound the rule base and are represented as: k_α radians; k_q rad/s; and k_{θ_e} radians.

Rules	RCGA	k_α	k_q	k_{θ_e}	N_C (/500)	$\int \lvert\alpha_{\text{err}}\rvert \mathrm{d}t$ (rad.s)	$\alpha_{\text{err}}(t_s)$ (degs)
125	GA_4	0.08788	1.0312	0.2601	500	0.02047	0.318
225	GA_4	0.07237	0.08211	0.1914	500	0.01826	0.261

Figure 6.17 shows the evolution of the input scaling values for the 125 rule design. The large variations significantly alter the behaviour of the rule base, however, within 100 generations they have effectively reach their final levels. This process can take longer with a larger rule base and be disruptive to the evolution of control solutions. The performance of the final control solutions indicate that they are more robust (*ie.* $N_C = 500$) then their counterparts from Table 6.3. The robustness is derived from allowing the global features to be adjusted to the demands of the test simulations rather than being forced

to compromise with fixed bounds. There is also a flight response benefit as shown in Figure 6.18. Though there is little difference in the behaviour of the 125 rule example, the tuned input scaling values provide a more rapid attitude response for the 225 rule controller.

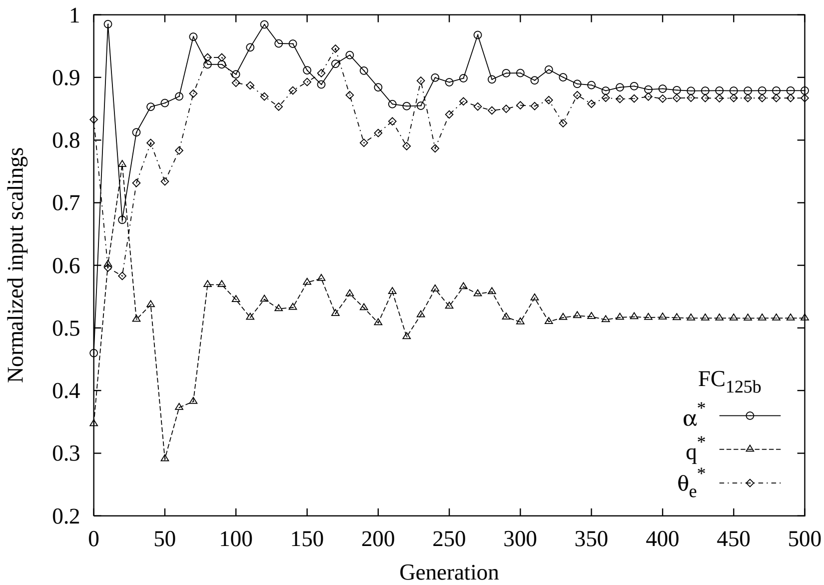


Figure 6.17: Evolution of the input variable scaling parameters during the design of the 125 rule controller. The values have been normalized by the search range of each parameter.

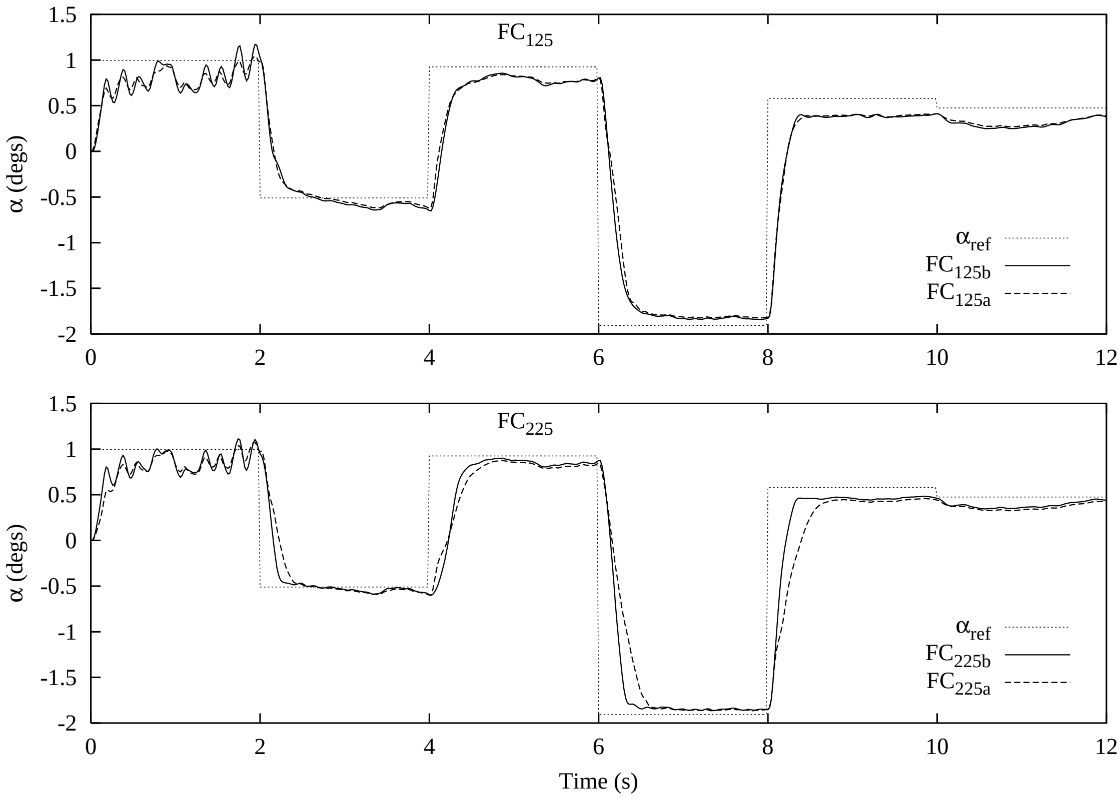


Figure 6.18: Simulated performance of controllers where both the rules and the input scaling are designed. The notation established in Table 5.2 have been used to distinguish between the extended design example (125b,225b) and the previous solutions generated by design of the rule output array only (125a, 225a).

6.5 Longitudinal Autopilot

To provide a demonstration of the scramjet powered vehicle flying over the full hypersonic trajectory, a guidance loop was designed, thus completing the longitudinal autopilot specification. The outer-loop guidance control function is simply defined by the linear feedback vector $\boldsymbol{K}_G$, where the inputs are altitude error and the climb rate error, and the output is an angle of attack command. The nominal altitude is determined by flight along a constant dynamic pressure trajectory, while the nominal climb rate is evaluated by combining the vehicle acceleration with the nominal trajectory. Since the guidance vector is designed using the simulation of the uncertain vehicle model, a small population genetic algorithm search was used to design the $\boldsymbol{K}_G$ gains. The design configuration, following the evolutionary approach used for the inner loop, is (GA_2, OF_4). The objective function is established from 12 simulations ($t_f = 30$ s) selected from the full flight envelope of the vehicle.

Using the 225 rule controller presented in Table 6.4 (FC_{225b}), guidance gains were designed for an update period of $\Delta t_g = 0.5$ s and 1.0 s,

$$\boldsymbol{K}_G(0.5) = [-8.307\times 10^{-4},\ -2.396\times 10^{-3}] \tag{6.4}$$

$$\boldsymbol{K}_G(1.0) = [-2.004\times 10^{-4},\ -1.054\times 10^{-3}]. \tag{6.5}$$

The top part of Figure 6.19 shows the simulated altitude response using the two guidance timesteps. Both show a long-period mode associated with longitudinal motion, with the $\Delta t_g = 0.5$ s simulation displaying superior tracking of the nominal trajectory. The lower two parts of Figure 6.19 show the angle of attack command (with response) and the elevator history for the $\Delta t_g = 0.5$ s simulation. Most notable is the cycling of angle of attack between the two command limits, generating the flight angle (γ) oscillation about what should be a near constant flight angle. Of importance to the stability of the inner-loop controller is the lack of any significant overshoot in the angle of attack responses. The noise in the elevator response is a direct result of the uncertainty in the vehicle performance model plus the signal noise.

A full trajectory simulation is displayed in Figure 6.20 for the $\Delta t_g = 0.5$ s arrangement. The 240 seconds of flight approximately represents the full flight time allowed by the fuel supply. Further consideration of the vehicle configuration and the guidance law development is required to fully appreciate the requirements for improved altitude tracking performance, but is beyond the scope of the present work. The inner-loop controller is also deserving of further study, particularly in terms of providing improved attitude response for the typical command series generated by the autopilot.

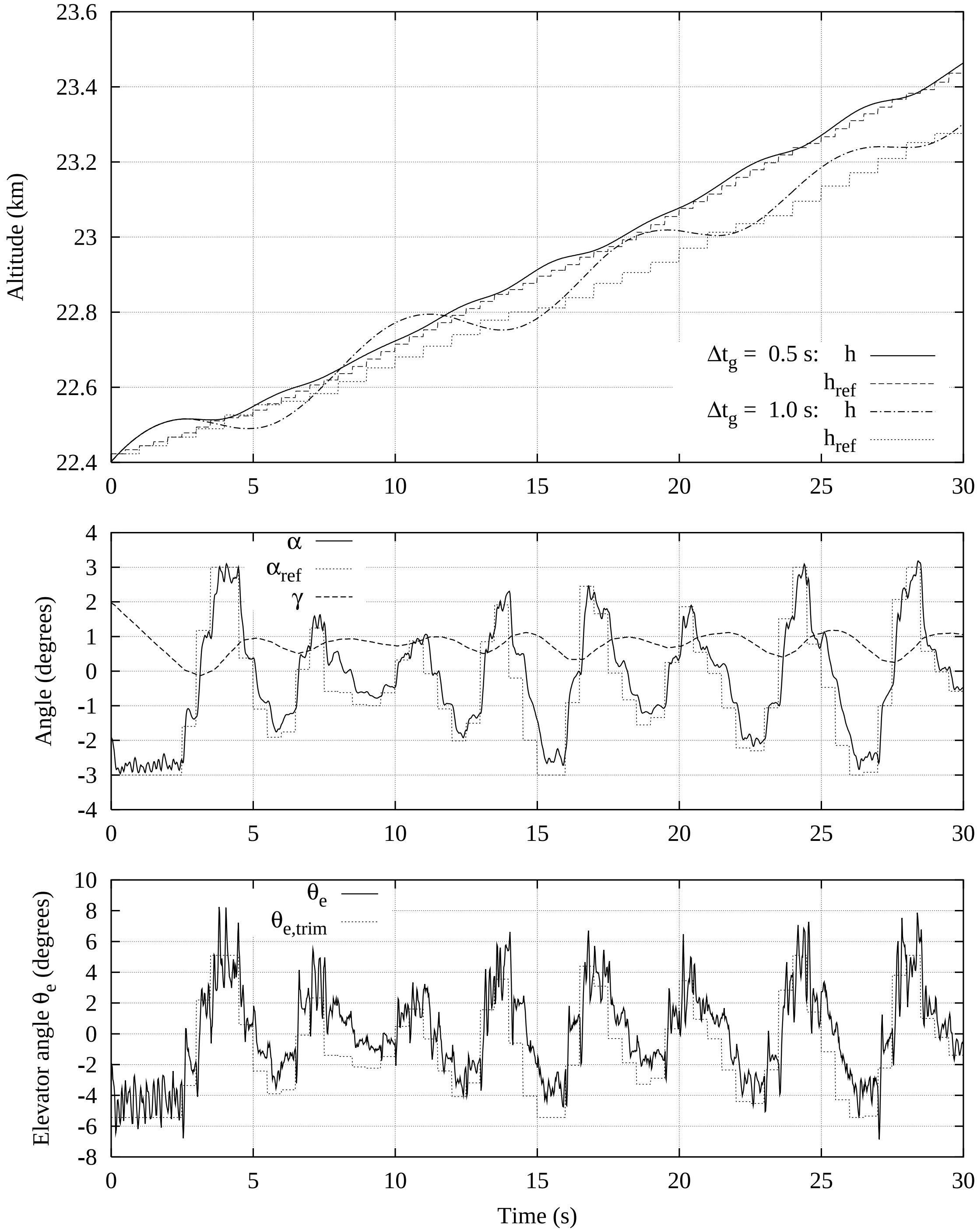


Figure 6.19: Autopilot response for two guidance update timesteps: $\Delta t_g = 0.5,\ 1.0$ seconds. The angle of attack and elevator response refer to $\Delta t_g = 0.5$ seconds.

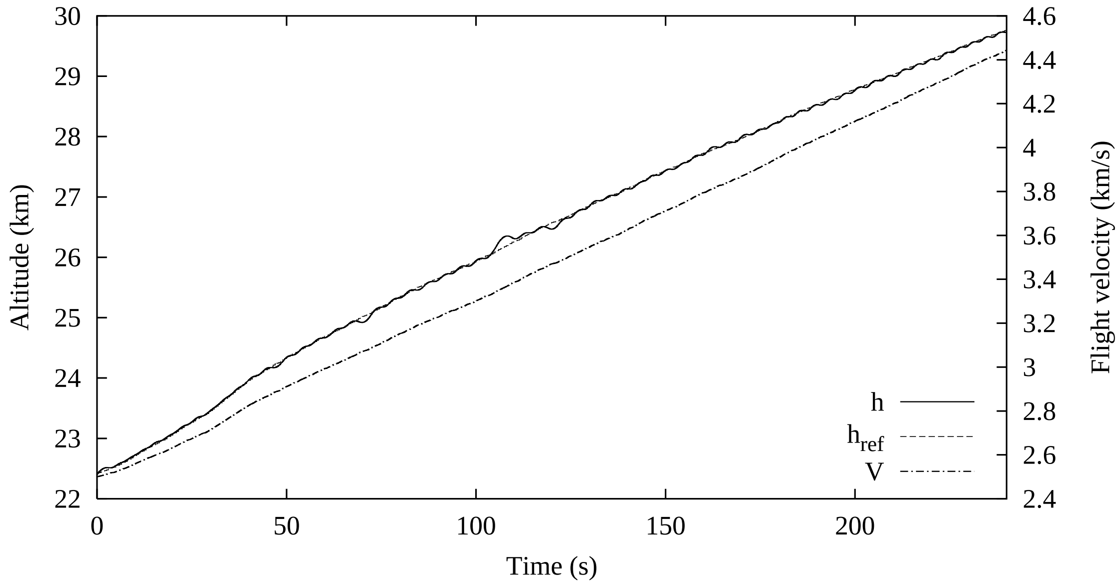


Figure 6.20: Full trajectory simulation of the hypersonic air-breathing vehicle, $\Delta t_g = 0.5$ seconds.

6.6 Genetic Algorithm Considerations

The application of the genetic algorithms to a new design problem invariably leads to experimenting with the structure and parameterization of the algorithm. As part of the overall control design approach of this thesis, the GA experiments investigated the effect of the genetic operations of mutation and crossover, and the dimension of the search as specified by the population size and the generation number. The following sections discuss the crossover and mutation operators, and population size.

With regard to the crossover operation, it was noted in Appendix A that the potential benefits of an operator which provided exploration as well as exploitive qualities, depends on the nature of the objective function. It appears that with the non-uniform objective function used for the inner-loop control design, rapid evolution of the control solution is possible, and the design benefits from the exploitive nature of arithmetic crossover. The arithmetic crossover used to generate the results of this chapter provided superior performance to a single point crossover and the BLX crossover used in Appendix A. The tendency of the arithmetic crossover to converge to the centre of each parameter domain is mitigated by the use of high mutation rates. However, it is possible that the initial rapid decrease in population entropy (see Figure 6.21) is due to the crossover action. Further investigation is required to fully understand the population behaviour in the initial stages of the search.

Given the search behaviour is closely linked to the crossover and mutation operations, there also appeared to be little benefit in extending the search over more generations. In particular, the non-uniform action of the mutation operator is coupled to the generation number rather than the fitness of the individuals, so that the mutation behaviour, and consequently the population behaviour, scales with the length of the search (N_G). The net result is that the objective function profile also scales with search length and the increased computation does not necessarily lead to improved solutions.

6.6.1 Mutation Operator

During the description of the real-coded genetic algorithm in Chapter 4 it was noted that the non-uniform mutation operator proposed by Michalewicz, displayed a bias to the centre of the search range. Using a collection of standard test functions, Appendix A details an empirical study of the mutation operator and a beneficial modification. The modified operator, referred to as an adaptive range mutation (ARM), displayed greater search reliability and was less sensitive to the parameterization of the genetic algorithm. In this section, the performance of the mutation operators are assessed using the control design problem for a 125 ruled inner-loop controller, with the design specification (GA_5, OF_3, FC_{125a}). It was because of initial difficulties with the control design that the performance of the mutation operator was first examined.

Figure 6.21 shows the evolution of the objective function (F_{obj}) and the population, for the two mutation operators and various parameter settings. NUM refers to the Michalewicz non-uniform operator and ARM is the adaptive range mutation formed by redefining NUM. The collection of settings for the action of NUM (p_m and β) address the general need for low activation rates and a rapid reduction in the available mutation with generation number g. When using the NUM operator the quality of the solution provided by the genetic algorithm is sensitive to the parameterization of the genetic operators. In general, the lower value of β generates more noise in the objective function. The fact that the noise level for $p_m = 0.2$ and $\beta = 2$ is maintained through all the generations indicates that the population is slow to converge, with the converged solutions being far from optimal. The trend is also observed in the population entropy lines. By reducing the mutation probability (p_m) and increasing the rate of fine-tuning (β), the performance of the genetic algorithm with NUM improves but remains inferior to the solution provided when using ARM.

Figure 6.22 shows the operation of the various control solutions on a series of step commands in angle of attack, α. In addition to the set from Figure 6.21, a design using ARM with $p_m = 0.2$ and $\beta = 2$ is shown to highlight the relative insensitivity of the search performance with the modified mutation operator. Though the assessment of search reliability requires a number of independent control design simulations, the trends observed here follow those for the more thorough investigation in Appendix A: specifically, that fewer generations were required to recover from bias, and that the solution was

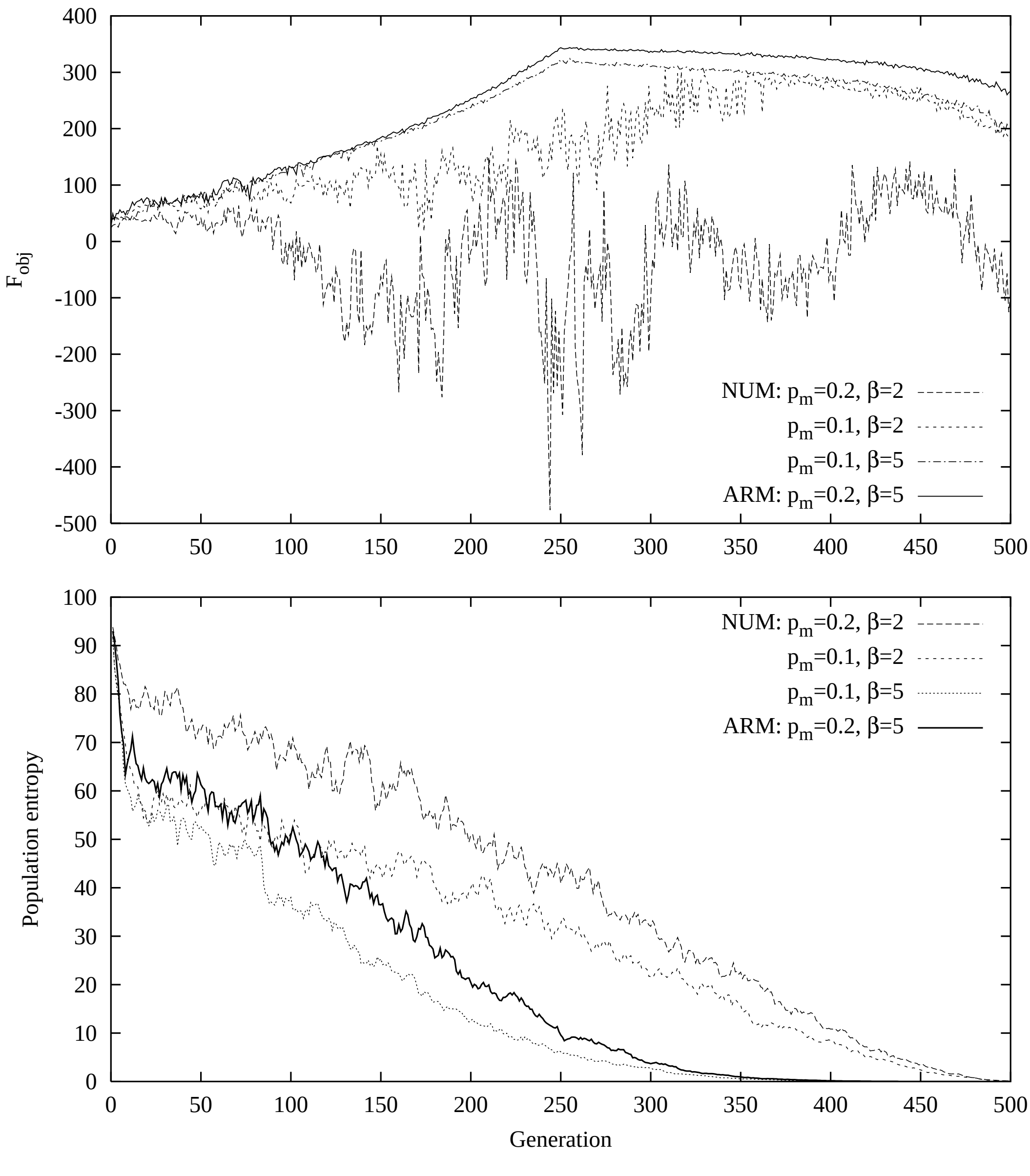


Figure 6.21: Evolution of the objective function and the population entropy using Michalewicz's non-uniform mutation (NUM) and the modified version proposed for this thesis (ARM).

less sensitive to its placement in the search domain and the parameterization of the operator. The search bias of the original non-uniform operator is demonstrated in the scattering of the rule consequents for the control solution in Figure 6.23. Despite the mutation preference for values in the centre of the search domain, the inherent search robustness of the genetic algorithm is still able to generate a reasonable control solution.

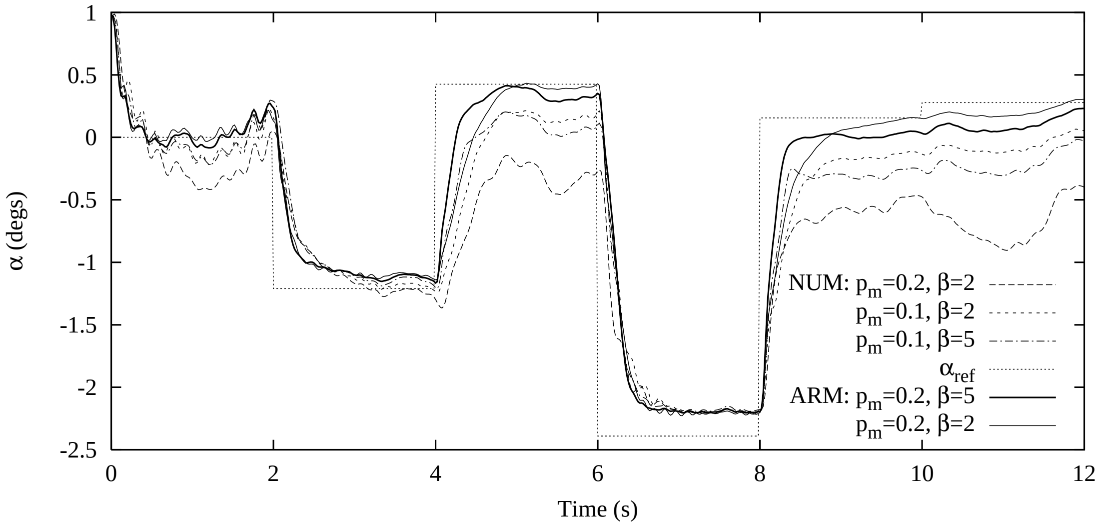


Figure 6.22: Impact of the mutation operator on the angle of attack performance of the evolved controller. Every 2 seconds α_{cmd} is reset and the flight condition is shifted to another point T_i on the trajectory, starting with T_1.

6.6.2 Population Size

The choice of population size is fundamental to the operation of the genetic algorithm, affecting both the convergence rate of the search and the quality of the final solution. Too small a population and premature convergence will likely lead to a poor solution, while increasing the population must be considered against the time required to generate the final solution. In the population experiments here, the design time on a single processor of an SGI Origin 3000 ranged from 9.2 hours for $N_P = 10$ to 81.3 hours for $N_P = 100$, see Table 6.5. The most efficient arrangement is likely to utilize a moderate population size to capture a *good* solution, and then apply a fine-tuning procedure to enhance the controller performance.

Figure 6.24 shows the growth of the best objective function and the population dynamics throughout the evolution of a controller of 125 rule controller. The design arrangement is specified as (GA_7, OF_3, FC_{125a}). The most consistent growth in the quality of the controller is provided by the two larger populations. With smaller populations, there is less variation amongst individuals in the population, and they are therefore more susceptible to large variations in the quality of the solution across generations. Population entropy is

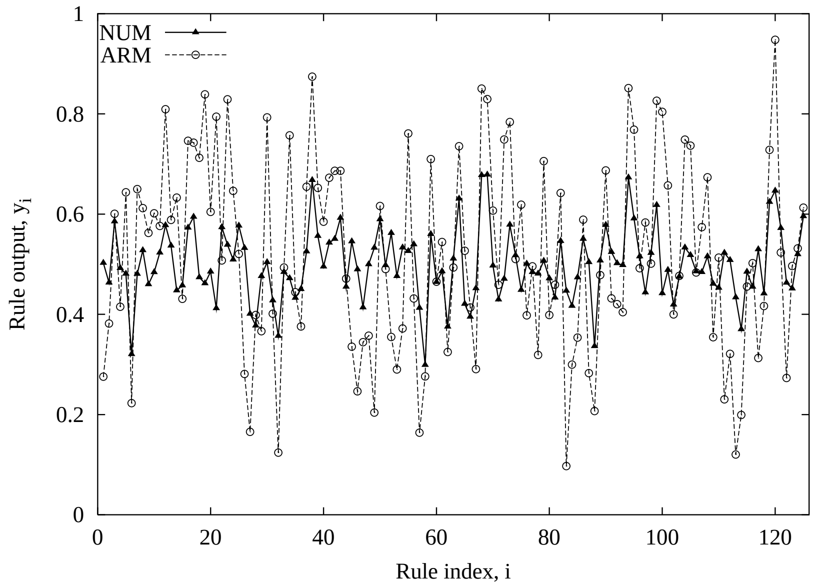


Figure 6.23: Rule consequents designed by the two mutation operators, NUM ($p_m = 0.1, \beta = 5$) and ARM.

Table 6.5: Quantitative performance comparison of the controller design generated using populations 10, 30, 50, and 100. Design times refer to the CPU time used on a single processor of an SGI Origin 3000.

N_P	RCGA	Design time (hours)	N_C (/500)	$\int \lvert\alpha_{\text{err}}\rvert \mathrm{d}t$ (rad.s)	$\alpha_{\text{err}}(t_s)$ (degs)
10	GA_7	9.2	495	0.04211	0.636
30	”	25.4	497	0.02371	0.282
50	”	42.9	497	0.02097	0.300
100	”	81.3	499	0.02129	0.318

used as a measure of the chromosomal variation amongst individuals in the population. As the population size is increased, it is able to maintain greater diversity while still providing superior solutions. The history for a population of 10 individuals stands apart from the others, with rapid convergence of the population and large variations in the entropy. Table 6.5 summarizes the performance of the solutions generated and Figure 6.25 shows the performance for a particular series of step commands. Here again, the population size of 10 stands out as providing a solution of significantly lower quality. The greater exploration of the search space available with larger populations allows the early generation of good solutions while maintaining population diversity, and an extension of the time for which the search is effectively performing a fine-tuning role. However, there is no benefit, in terms of the response characteristics, for the doubling of the design time going from a population of 50 to 100. It also appears from the results in Table 6.5 that the robustness issue addressed in previous sections, is due to the limitations of the objective function rather than insufficient search time.

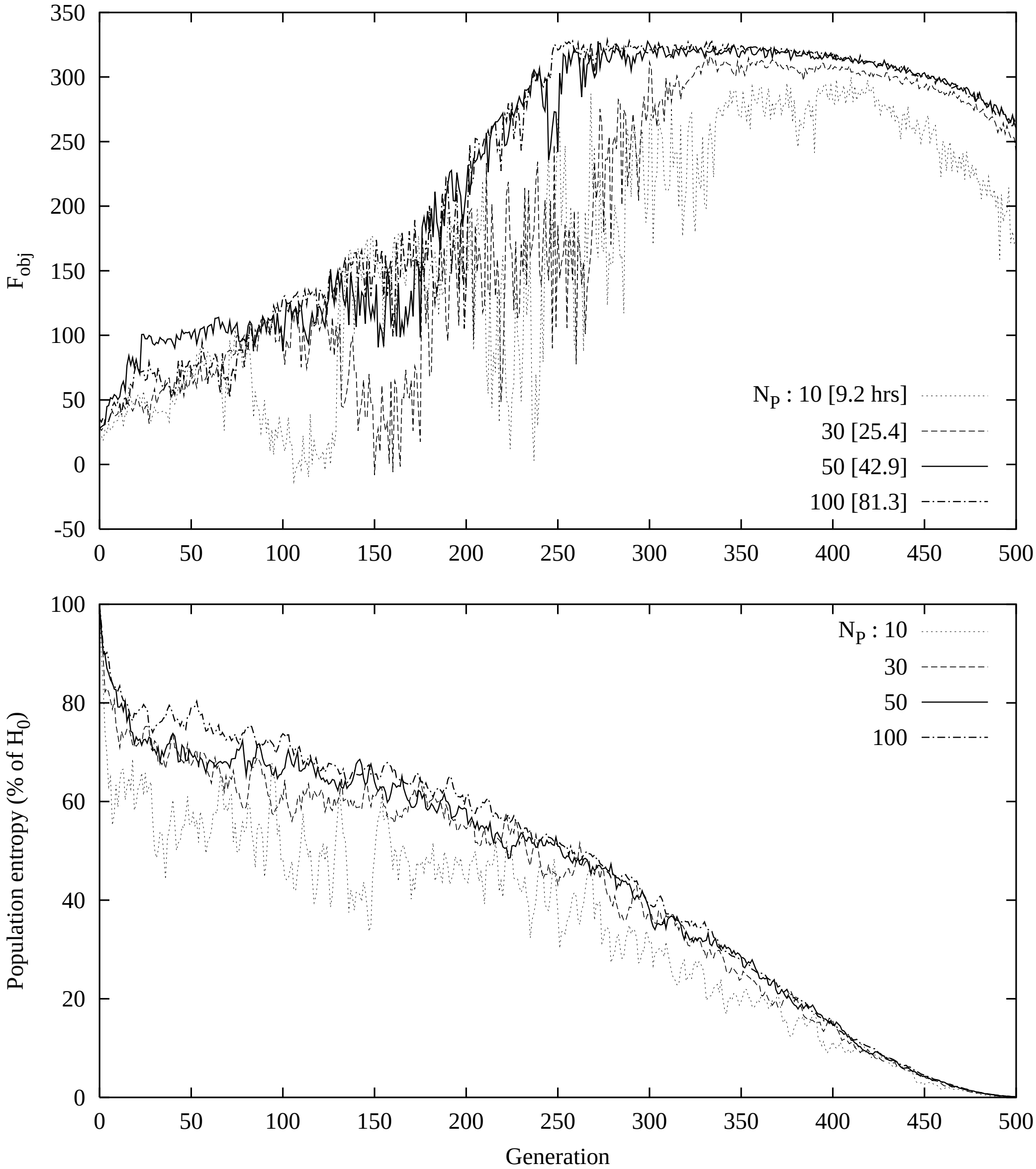


Figure 6.24: Evolution of the objective function and the population entropy, for populations ranging in size from 10 to 100. The population entropy history is represented as a percentage of the entropy measure for the initial population.

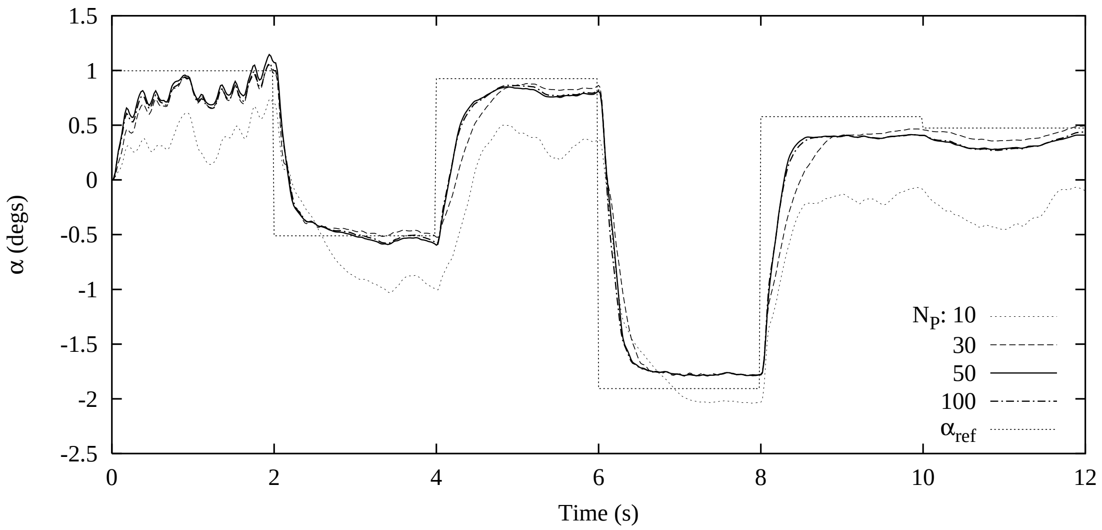


Figure 6.25: Performance of 125 rule controller solutions provided by population sizes in the range 10 to 100.

CHAPTER 7

Conclusions

The aim of this thesis was to investigate the application of an evolutionary design approach for the configuration of a robust flight control system for a hypersonic air-breathing vehicle. It is a problem recognized (characterized) by the strong interaction of engine operation with the flight condition and attitude, its nonlinear performance, uncertainty in the performance of system components, and its highly constrained operating envelop. Consequently, most other investigations of flight control approaches for hypersonic vehicle applications have centered on applications of robust control theory, generally involving a linear description of the vehicle with uncertainty models accounting for performance variation and unmodelled features, included in the design process. For this work a full nonlinear flight simulation module was constructed for the purpose of provided control performance assessment during the design procedure

In Chapter 2 the basic arrangement of a the longitudinal autopilot for the hypersonic vehicle was introduced. Two control loops were defined: a longitudinal inner-loop providing stability augmentation and attitude tracking; and an outer guidance loop for the maintenance of the nominal flight trajectory. The guidance function generates attitude commands for the inner-loop to follow. To satisfy vehicle operating requirements, a nominal trajectory with constant dynamic pressure was used. Chapter 2 also introduced the evolutionary design approach to be used for the determination of the control functions.

Since the control design procedure was dependent on the simulated response of the vehicle, considerable time was spent developing a detailed numerical flight simulation module. This is in contrast to previous studies where the vehicle model is either defined by linear analytical expressions, or extracted from a database of performance parameters. The general arrangement for the hypersonic vehicle is taken from a small payload launch vehicle application with an axisymmetric scramjet powered second stage. Despite the ultimate desire to accurately portray the vehicle properties and behaviour, it was necessary to simplify the vehicle model, to ease the computational burden while maintaining the essential operating features. The principal simplification was the use of two-dimensional flow paths, thereby representing the axisymmetric scramjet vehicle as a box section. Chapter 3 detailed the representation of the vehicle physical properties, the aerodynamic and propulsive simulation, an environment model, and the description of the general six degree-of-freedom flight dynamics equations.

Though the aerodynamics and propulsion performance were described by a combination of one-dimensional and two-dimensional flow models, the basic longitudinal behaviour of the vehicle is captured. Primarily, this includes the performance dependency on angle of attack and the flight condition. The aerodynamics and propulsion analysis describe an instantaneous representation of the flow structures throughout the vehicle. External aerodynamics are treated separately to the gas dynamics within the engine flow paths, which themselves are divided into inlet, combustion, and nozzle processing regions. For the engine analysis, 75 % of the computational effort is directed toward the expansion fan interaction model used to described the pressure profiles generated along the nozzle surfaces. Significant reduction in the design time would be available through the parameterization of the nozzle forces and moments in terms of the upstream Mach number, with the upstream pressure applied as a multiplier. Since the vehicle performance is inherently uncertain, parametric uncertainty was introduced to describe stochastic perturbations in the engine performance, control effectiveness, and the vehicle centre of mass. These processes were implemented as low-pass filters (based on a Nyquist frequency of 50 Hz, with cut-off frequencies representing the characteristic behaviour), and a white noise input whose variance is adjusted to realize the appropriate perturbation magnitude. Atmospheric turbulence and input signal noise were also simulated as random processes. The inclusion of uncertainty and disturbances in the flight simulation provides a means for the control design to be robust against unmodelled behaviour.

The use of two-dimensional flow paths makes the vehicle particularly difficult to control. Moments generated by the inlet wedge (in contrast to the conical forebody for an axisymmetric configuration) place great demands on the control actuators, requiring large control surfaces and actuation rates. Due to the extension of the cowl over the full length of the nozzle, there was no stabilizing benefit from the differential throttling of the engines. Stabilizing capability of the nozzle would be greatly improved with a shortened cowl section, lessening the actuation required by aerodynamic surfaces. Such an arrangement is used for the American Hyper-X project, which also benefits from having the scramjet engines on one side of the vehicle only. In terms of the overall vehicle performance, being a hydrocarbon fuel scramjet, the specific impulse is quite low. Even though viscous losses were neglected in the analysis, the relatively poor acceleration capability can be apportioned to the non-optimal vehicle geometry and the two-dimensional flow paths.

Chapter 4 introduced the overall approach to control design, as applied to the hypersonic vehicle. The central control problem was considered to be the specification of longitudinal inner-loop controller and, for this role, a fuzzy controller was used. The reasoning was that fuzzy control offers desirable robustness characteristic and provides a relatively simple means of describing a complex, nonlinear control function. With the fuzzy controller defined by a set of fuzzy variables, a rule base array, and an inference mechanism, the design approach is one of knowledge acquisition. A real-coded genetic algorithm (GA) was constructed to evolve the necessary knowledge, using the simulated vehicle response as a performance indicator. Evolutionary algorithms such as the GA, have shown to be efficient search tools for complex, nonlinear, and noisy design spaces. Prompted by initial difficulties in designing the control functions, an investigation of genetic algorithm operation led to modification of a well known mutation operator to avoid the bias it generated in the solutions. An empirical investigation of this modification is the focus of Appendix A, where a set of standard minimization test functions were used to analyze the performance of the genetic algorithm. Reliable performance of the real-coded genetic algorithm, with the modified operator, was shown to be relatively insensitive to the parameterization of the algorithm and the use of different crossover operators.

The evolutionary design procedure uses full nonlinear vehicle simulations in the design loop. With the search starting from a random set of initial solutions the approach is well deserving of the *brute force* title. Using vehicle simulations in the design loop is a scheme with great potential, but one where considerable care is required. On the positive side system features exposed to the controller configuration during the design, are not constrained by the reduction of the system to a set of analytical expressions. Performance uncertainty can readily be considered through the inclusion of models which can describe expected variations on the basis of their physical origins. There is also considerable flexibility in the configuring of the cost function, in terms of the possible characteristics which may be used to encourage rapid response, minimum overshoot, steady state error, and stability for example. However, with the genetic algorithm being extremely opportunistic, the coupling of competing performance measures requires close attention to avoid the design procedure exploiting any loop-holes in the definition of the cost function. To provide robustness assurances, it is also necessary to provided a large sample of test conditions. This is particularly important as the discretization of the control function through a set of fuzzy rules, means that for each flight response, only a portion of the total rule is likely to be activated.

Chapter 5 was used to introduce the specific details of the current experiment in evolutionary design, specifically those relating to development of performance and stability robustness in the control design and the promotion of rapid evolution of good solutions. A large set of initial conditions, covering the full range of allowed attitude, control and flight condition variations, are used to generate desirable performance and stability qualities in the control solutions. The genetic algorithm is a population based search tool, so with each performance evaluation requiring many flight simulations, a non-uniform objective function was introduced to provide useful selection pressure. For the case where the search begins from an initially random set of potential solutions, avoiding *needle in a haystack* type problems is critical for the design using small populations and relatively few generations.

For the design of the fuzzy controller a preset discretization of the input variables was used. This provided a structure to the rule base, based on providing all possible condition statements for the input variable definitions. Two design problems are therefore created, the first configuring the output array and the second including the input scaling values. The inclusion of input scaling in the list of design variables, allows the global features of the rule base to be influenced.

The results presented in Chapter 6 focus on the design of the inner-loop control function, used for stability and attitude maintenance. An initial investigation of a linear controller showed that, while a robust controller could be configured, the performance was compromised by the uncertainty in the vehicle trim condition. Through experiments with the design of the linear controller, the set of necessary control inputs (attitude error, pitch rate, and elevator trim error) were established. Experiments with the control update frequency revealed the sensitivity of the vehicle to disturbances, with a maximum timestep of 0.04 seconds being possible while still providing stable vehicle operation. Using the gains provided by the linear controller to evaluate the input scalings of the fuzzy variables, a tuned fuzzy controller was generated using the nominal vehicle model (no uncertainty) and a Nelder-Mead optimization procedure. Though it was evident that the fuzzy controller could provide improved attitude response, the lack of any variation in the vehicle model and test conditions meant the optimization process reduced the generality of the controller and consequently its robustness suffered. For the most part, the improved response characteristics were realized by the extension of the input range over which large actuation rates could be applied. The fuzzy controller provides this capability through the discretization of the control function which allows local manipulation of the control surface.

The bulk of the results related to the evolution of fuzzy controllers. By using the genetic algorithm as the design tool, performance uncertainty could be included in the design process, which ultimately generated controllers with greater robustness. The genetic algorithm demonstrated a remarkable ability to rapidly evolve good solutions. Due to the non-uniform nature of the mutation operator, the earlier good (stable) solutions appear in the population the better the performance of the design. While the objective function promoted this behaviour, it is also desirable to use a larger population to realize the true potential of larger rule bases. In general, however, the genetic algorithm proved very capable of configuring controllers ranging in size from 27 to 225 rules, over 500 generations and with relatively small populations (30-50). It was also observed that the set of initial conditions and step responses used during the design, closely mirrored the behaviour of the controller against a much larger set simulation experiments. It was also noted that, due to the bounding of the rule base with predefined input scalings, the controller stability was susceptible to extreme combinations of initial conditions which were not directly included in the design. Extending the design set or introducing variations in the design set throughout the evolutionary process, may allow controller robustness guarantees to be met. The experiment with a variable initial condition set was not included in the results but showed promise in terms of improving the overall controller performance.

The representation capability of the fuzzy controller covers both the inherent capability of a fuzzy rule base to encode a desired multi-dimensional function and the means of achieving the representation. Though better control might be realized with larger rule base (through greater partitioning of the input space), there is a tradeoff between the control accuracy and the tuning cost. The larger the rule base the greater the capacity for specialization, which can lead to robustness issues if the test set of simulations does not provide sufficient coverage of possible environmental conditions. Essentially, for the simulation-based optimization used in this work, the representation capability of the fuzzy controller is strongly dependent on the preparation of suitable test conditions. Various means of specifying the test set of flight responses were discussed ranging from varying the test set during the design and using a series of step responses from each trial initial condition. Further study on these approaches is needed to assess their worth.

With input scaling included amongst the design variables it was possible to improve the attitude response characteristics of the vehicle. This amounted to a faster initial response while still providing reasonable settled-response features, and avoiding overshoot which could lead to vehicle failure. It would be desirable in the future to extend the design set of simulations to multiple step responses and to provide better stability guarantees through the objective function.

Given the successful design of an inner-loop controller, a simple guidance law was designed, using linear feedback of the altitude error and the climb rate error. As a demonstration of the broad range operation of the vehicle, it showed the steady climb along the nominal trajectory with persistent oscillations in the flight angle. The inner-loop controller was also shown to perform adequately over the full flight trajectory. Further investigation of the guidance arrangement is desirable, with particular consideration to the vehicle configuration and the design of the guidance function.

Recommendations and future work

The design freedom available with the combination of fuzzy control, genetic algorithms, and numerical flight simulations, generates considerable experimentation which, has led to many open questions and avenues for future research. A few these are now mentioned.

To begin with there is much work possible in improving the accuracy of the vehicle simulation module. This may involve the inclusion of more advanced computational techniques for the performance analysis and a more accurate representation of the vehicle structure. Of particular interest is the modelling of aeroelastic effects (which have been neglected in the present work) as small variations in the surface angle relative to the flow path can significantly impact the flow structures. Since the control actuation in this thesis used a rear all-moving wing, further work is warranted on the validity of alternative control actuators with the aim of reducing the large penalties associated with the current aerodynamic surfaces.

In terms of the fuzzy controller, the full design freedom available has not been considered here. The potential of including fuzzy variable definitions and rule generation in the design is worth consideration. To prevent an explosion of the design complexity it may be preferable to maintain a predefined rule base structure and, for large rule bases, to provide a mechanism for removal of rules which are essentially unused. Another approach to improving the representation capability of the fuzzy controller is the application of coevolutionary algorithms [167]. A coevolutionary scheme works simultaneously on two populations. One provides a set of possible control solutions while the other describes a set of test simulations (initial conditions) used to evaluate the fitness of the control solutions. Evolution of the environmental conditions is based on the success rate of controllers, meaning candidate solutions can evolve against worst case scenarios.

One of the pitfalls of working with the genetic algorithm is that there are so many ways of manipulating the operation of the algorithm that one ends up in an experimentation cycle from which is difficult to escape. Though GAs are well established algorithms, research continues on the efficiency and reliability of the search they provide. The major avenue for enhancement of the design performance would be to exploit the inherent parallelism of the genetic algorithm. In so doing, it is possible to reduced the design time from days to a small number of hours.

At the beginning of this research, automation of the fuzzy control design was commonly associated with genetic algorithms. This led to choosing them for the flight control design problem. They have generally been applied using binary-coding however, real-valued coding, as was used here, has been shown to provide better performance in real-value parameter optimization problems. Evolutionary strategies have since shown generally superior performance on real-valued problems and would be particularly suited to the flight control problem. The reasoning follows the argument by Salomon [190], that GA's are very time consuming if the parameters exhibit epistatis, where epistatis describes the interaction of parameters with respect to the fitness of the individual. Further, it has been indicated that the high global convergence performance of GAs is reliant on the independence of the parameters. For control problems the fitness of a set of parameters is generally highly dependent on the interaction of the parameters. For evolutionary strategies, which have been especially designed for real-valued applications, the performance is invariant with respect to epistatis and would therefore be an ideal candidate for the control design problem presented in this thesis. However, in the case of this thesis, the GA was able to provide a control solution which represents a near optimal formulation.

APPENDIX **A**

An Adaptive Range Mutation Operator for a Real-Coded Genetic Algorithm

This appendix presents an empirical study of a modification to the Michalewicz non-uniform operator for real-coded genetic algorithms. The modification aims to improve the reliability of a genetic algorithm applied to function minimization problems. Both the original non-uniform operator and a more recently proposed adaptive non-uniform operator are shown to direct the search to certain areas of the search space. This search bias reduces the potential benefits of mutation in generating useful solutions, reducing the robustness of the genetic algorithm as a general search tool. An alternative operator definition is presented, and is described as an *adaptive range* mutation. It displays a general improvement in search quality and less sensitivity to the evolutionary mechanisms and parameterization of the algorithm.

A.1 Background

Genetic algorithms have, in the past, been distinguishable from other evolutionary algorithms by their use of crossover as the primary method of producing variation. Crossover generates new offspring and new search vectors by sharing the parents' chromosomal information. Mutation, initially introduced as a background operator through small activation probabilities, provides a randomized perturbation of chromosome elements. This provides a mechanism for reintroducing data that which was previously lost because of selection pressures and allows the exploration of new areas of the search space. Studies have shown that higher rates of mutation can improve the velocity and reliability of a genetic search, see for example [21]. To allow fine-tuning of optimal solutions the mutation can be configured to provide a random walk through the search space for early generations, and refinement in the later stages by gradually reducing the mutation magnitude. In real-coded genetic algorithms [116], where an individual's chromosome is represented by an array of floating point numbers, such a mutation has been referred to as a non-uniform mutation operator.

The motivation for modification of the mutation operator came from the application of the genetic algorithm to the design of a flight controller for a hypersonic vehicle, discussed in the main body of this thesis. Configured as a numerical optimization task, the controller design requires many expensive flight simulations to evaluate the performance of each potential controller. It is therefore desirable to rapidly acquire a good solution in order to limit the number of function evaluations, hence the use of a real-coded algorithm scheme over the binary coded algorithm. Whilst examining the performance of the genetic algorithm, a potential source of the design difficulties was identified as a mutation preference for values in the centre of the search range. This property of the Michalewicz non-uniform operator had also been recognized by Neubauer [162] who provided a theoretical analysis on the mutation variance and the expected value following mutation. Apparently unsighted was the preference of this corrected form to the extremes of each variable's feasible range. The full potential of the mutation operator in terms of search velocity and reliability is lessened by the operator's bias. The relative success of both mutation forms in other optimization problems may be attributed to the robust nature of the basic genetic algorithm structure, problem specific features, and the setting of exogenous parameters to mitigate the mutation bias.

This appendix examines the performance of the real-coded genetic algorithm, discussed in Chapter 4, to standard function minimization problems. The focus is a comparison of the mutation operators proposed by Michalewicz and Neubauer, and a modified non-uniform operator described here as an *adaptive range mutation*.

A.2 Real-coded Genetic Algorithm (RCGA)

Real-coding for genetic algorithms refers to the representation of an individual's chromosome as an array of floating-point values. We have configured our genetic algorithm using real-coding for the benefits it offers in reliability and search velocity on numerical optimization tasks. The evolutionary mechanisms used to construct the algorithm are detailed in Chapter 4. Before focusing on the mutation operators, a brief summary of the real-coded genetic algorithm is provided.

A simple algorithm structure has been used, as shown in Figure A.1, starting with a randomly generated initial population. Each individual in the population represents a search point in the space of potential solutions to the optimization problem. The problem definition provides an evaluation measure, referred to as the objective function. When scaled using linear scaling with sigma truncation, the objective function becomes the fitness measure used to direct the search. Stochastic remainder selection without replacement is used to select parents for mating, with complete replacement of the population for each generation. New individuals are created via the perturbation operations of crossover

```
begin
  t = 0
  initialize population (random)
  evaluate population
  while (t < T) do
    t = t + 1
    select parents for reproduction biasing fittest
    recombine individuals via crossover with mutation
    evaluate new population
  end
end
```

Figure A.1: The basic genetic algorithm structure.

and mutation. Arithmetic crossover [155] at a fixed probability p_c is used with the mixing parameter randomly generated each time recombination occurs. It is applied uniformly to the parent chromosomes. A single point crossover scheme has also be used to generate results for this appendix. Each gene in the children's chromosome also undergoes mutation with probability p_m.

The non-uniform mutation operator was introduced by Michalewicz and Janikow in their modified genetic algorithm [155], which they applied to numerical optimization problems. It has subsequently been reproduced (often with favourable results) in numerous publications, as one of a number of potential mutation operators for real-coded genetic algorithms, see for example [99, 154, 74]. Each gene that undergoes a mutation does so within the variable range, $x_i \in [a_i, b_i]$, producing the mutated value x_i' following an addition or subtraction to the original value x_i.

$$x_i'(t) = \begin{cases} x_i(t) + \Delta(t, b_i - x_i(t)) & \text{with probability } q \\ x_i(t) - \Delta(t, x_i(t) - a_i) & \text{with probability } 1 - q \end{cases} \tag{A.1}$$

where $\Delta(t, y)$ is the perturbation function, dependent on the generation t and the position y of the original value relative to the search boundaries,

$$\Delta(t, y) = y \cdot \left(1 - r^{\gamma(t)}\right) \tag{A.2}$$

with r a random number uniformly distributed in [0,1], and $\gamma(t)$ providing the fine-tuning capability according to

$$\gamma(t) = \left(1 - \frac{t}{T}\right)^{\beta}. \tag{A.3}$$

Here T is the maximum number of generations and β the strategy parameter which sets the degree of non-uniformity across the generations. An alternative representation of the perturbation function described in Equation A.2 is provided in [156],

$$\Delta(t, y) = y\, r\, \gamma(t). \tag{A.4}$$

Though Equation A.4 can improve the fine-tuning capability of the operator, it does so by reducing the maximal possible mutation with time, rather than simply reducing the probability of the maximum mutation being applied, see Figure A.2. Since many problems may benefit from large mutations in later generations, Equation A.2 is considered the more robust of the two, and is used here. The control design problem is one such problem which benefits from using Equation A.2. To match the controller design when using the two forms of Δ, it was necessary to offset T in Equation A.4 against the actual final generation, thereby allowing mutations of greater magnitude.

(a) $\Delta/y = 1 - r^{(1-t/T)^2}$ (b) $\Delta/y = r\,(1 - t/T)^2$

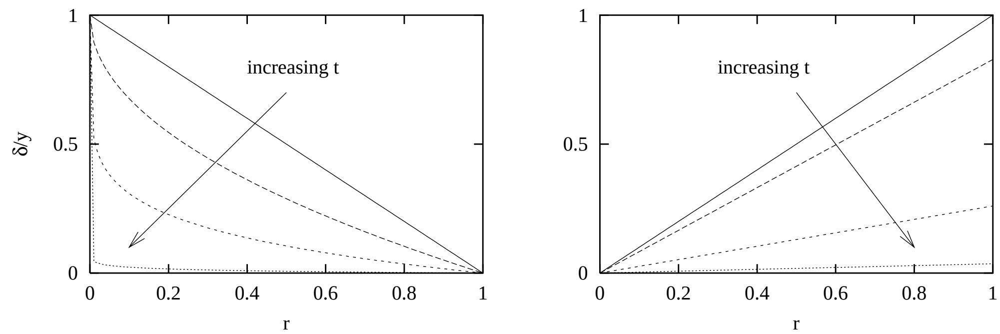


Figure A.2: Perturbation function for the non-uniform mutation operator, with $\beta = 2$. The four curves represent t/T ratios of 0, 0.3, 0.6, and 0.9.

Michalewicz [154] originally proposed that mutation to the left or right of the original value be equally likely, that is $q = 1/2$. When used with the variable scaling factor y in Equations A.1 and A.2, the result is a preference over time for values in the middle of the search range. This was noticed in the control design problem and supported by numerical experiments on the mutation operator. Figure A.3(b) shows the convergence of a random data set, under the action of mutation, to the centre of the search region. Neubauer [162] provided theoretical proof of the non-uniform mutation not being a zero-mean deviation operator. By forming an expression for the expected value of mutation and using the delta function represented by Equation A.2, the mutation was shown to concentrate the search between the parent value x and the centre of the search range. The effect lessens with increasing generations, as the possible mutation magnitude decreases. It is therefore possible to offset the potential for poor performances on some function minimization tasks by evolving the population over a large number of generations.

To change the non-uniform mutation to a zero-mean operator, Neubauer [162] suggested the use of an adaptive probability q, for the additive mutation of the parent value,

$$q = \frac{x_i - a_i}{b_i - a_i}. \tag{A.5}$$

The analysis of the new operator was based on the mutation variance and the expected value following mutation. While the modification provides zero-mean mutation, it does so by disproportionately sending values to the boundaries of the search space. From Equations A.5 and A.1, the closer a parent value is to the edge of the search space the more likely it is to be perturbed closer to that edge. By managing zero-mean mutation in this manner the population mean following mutation is maintained, but not by maintaining the population diversity. For example, a hundred data points each starting with the same value, $x_i = 0.9$ and $x_i \in [0, 1], i = 1, 100$, undergoes a hundred consecutive mutations according to Neubauer's adaptive non-uniform mutation. The result is that ninety points are placed along the upper boundary and ten points along the lower boundary, maintaining the initial population mean of 0.9. The effect of consecutive mutations on the random data set of Figure A.3(a) is shown in part (c) of the same figure. To allow for reliable solution finding a large number of generations are needed to counter the mutation bias, as well as an independent means to restore or maintain population entropy. The bias is more severe than Michalewicz's original operator, though the effect lessens as generations progress and the mutation is generally confined to a smaller range.

A.3 Adaptive Range Mutation

The mutation operators mentioned so far suffer from not allowing a random walk through the search space. For Michalewicz's operator, it is due to the operation not having zero-mean deviation, or $E(x') \neq x$, while Neubauer's correction is biased in the perturbation direction. A simple redefinition of the operator allows the non-uniform mutation to exhibit a random walk for early generations and, as the search progresses, provide the fine-tuning of the non-uniform operator. Instead of adding or subtracting increments to the parent value, the modified operator establishes a mutation range $x \pm \Delta(t, y)$ based on the generation number t and a fixed preset value y, and randomly selects a point within this range. For these reasons we have named the mechanism *adaptive range* mutation.

There are two steps to the modified mutation operator: establishment of the available mutation range, followed by a mutation yielding a value within that range. The maximal mutation is fixed by the search range $x_i \in [a_i, b_i]$, meaning early mutations are likely to access the entire range. Non-uniformity across generations is achieved by gradually re-

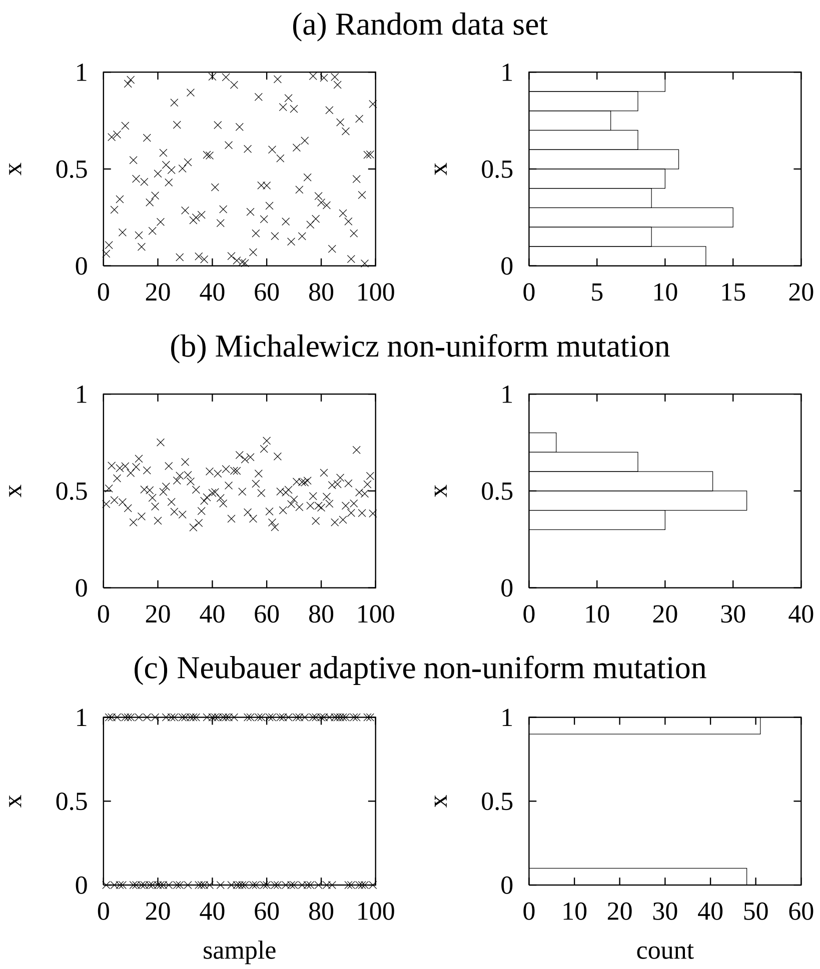


Figure A.3: The effect of 100 consecutive mutations on a random set of data points. The left column shows the individual data and the right shows the distribution as a histogram.

ducing the probability of large mutations, using the perturbation function of Equation A.2 to scale the mutation range relative to the maximal allowed mutation y. The mutation range $[\sigma_L, \sigma_U]$, is thus described by the following,

$$\sigma_L \quad = \mathbf{max}\,\{a_i,\; x_i - \Delta(t, y)\} \tag{A.6}$$

$$\sigma_U \quad = \mathbf{min}\,\{b_i,\; x_i + \Delta(t, y)\} \tag{A.7}$$

where $y = b_i - a_i$ and the maximum and minimum functions ensure bounded mutation. The act of mutation returns a random value within the range $[\sigma_L, \sigma_U]$, with the assurance of symmetry about the parent value x.

$$x_i' = \begin{cases} x_i - (1 - 2p) \;\; (x_i - \sigma_L) & \text{if } p \leq 0.5 \\ x_i + (2p - 1) \;\; (\sigma_U - x_i) & \text{otherwise} \end{cases} \tag{A.8}$$

where p is a random value uniformly distributed within the range $[0, 1]$. Figure A.4 shows the adaptive range mutation operating on initially random and linear data sets. Without any selection pressure the population entropy is maintained as each undergoes a random walk with fine-tuning. The distribution of points across the search range following 100 mutations is shown in the histogram beside each mutated data set.

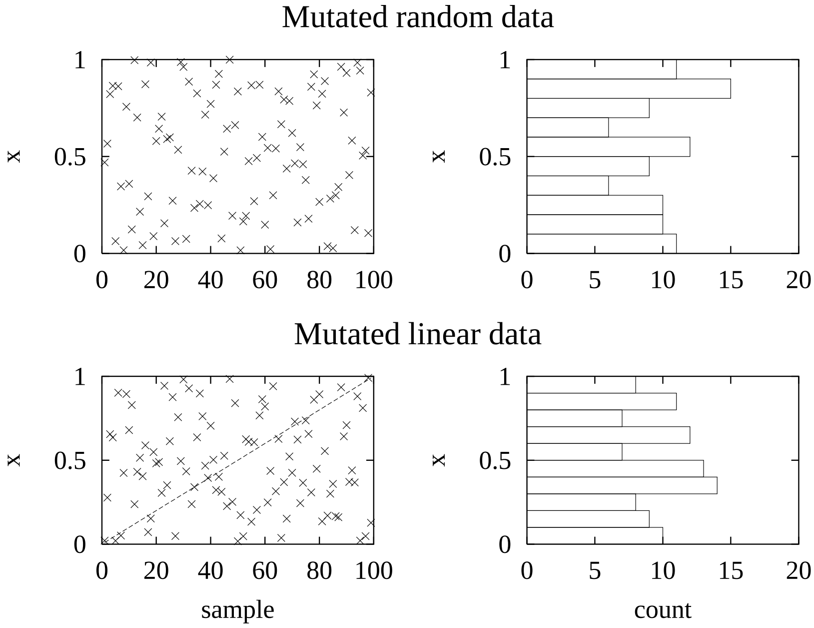


Figure A.4: Adaptive range mutation in isolation, perturbing initially random and linearly distributed data sets. The dashed line indicates the initial linear distribution.

Figures A.5 and A.6 show the mutation profiles for the mutation of a central ($x = 0.45$) and an edge ($x = 8$) initial value. Snapshots are taken for generations at $t/T = 0.01, 0.3, 0.6,$ and 0.9. The plots show the general symmetry of the adaptive range operator across the generations. To correct the bias of the non-uniform operator to the centre of the search space, Neubauer's adaptive scheme reduced the likelihood of mutations moving towards the centre. The Figures also show how with a large β, the resulting rapid reduction in mutation magnitude can mitigate the bias of the operators defined by Michalewicz and Neubauer. With the mutation magnitude described by Equation A.2, setting $\beta = 5$ effectively means that past half way through the evolutionary process, the mutation produces negligible changes to the chromosome values.

A.4 Test functions

To compare the performance of the mutation operators, a set of six benchmark functions are used. These are sourced from a much larger collection which have been applied to

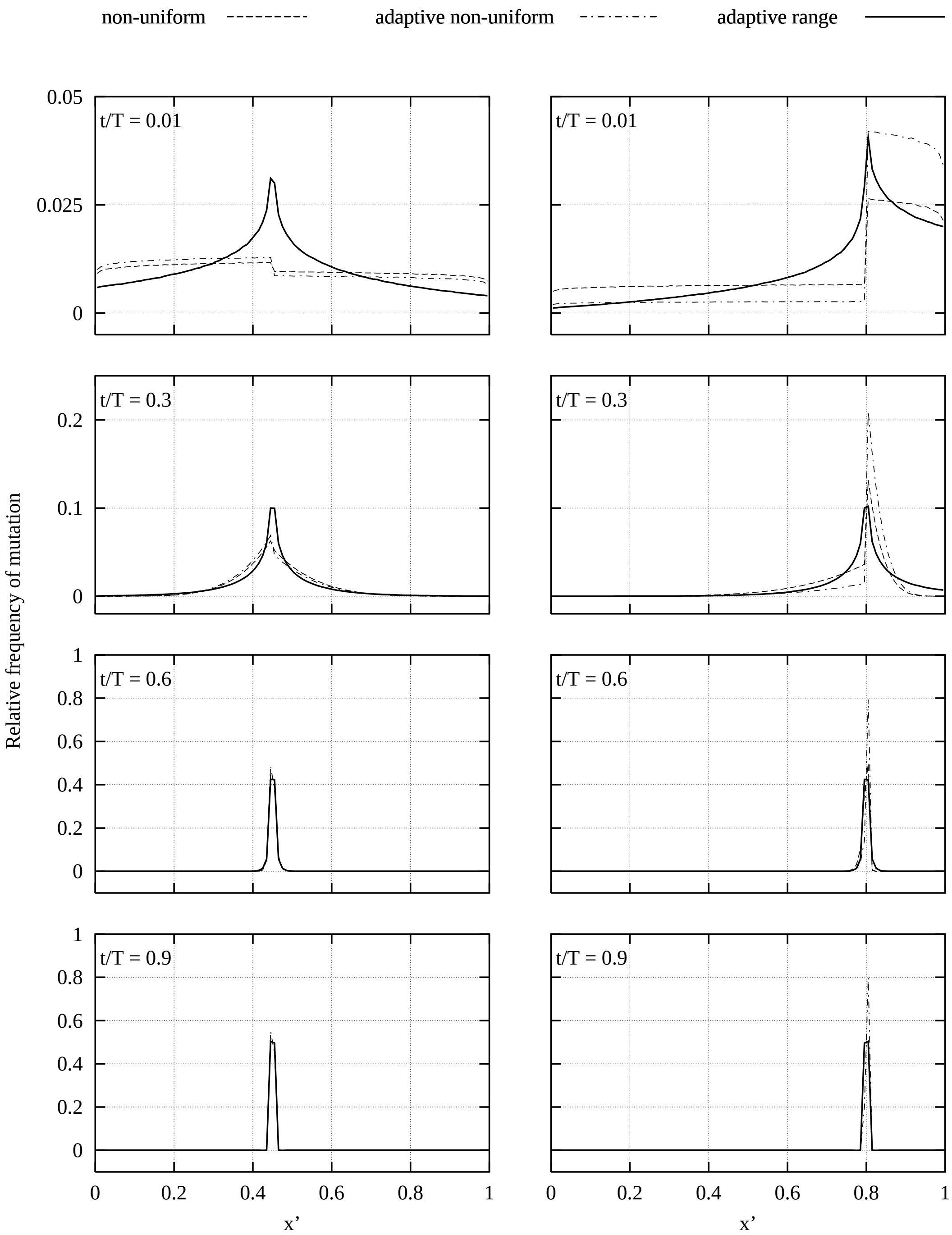


Figure A.5: A history of mutation profiles with $\beta = 5$. The left column shows the mutation of a value near the centre of the search range (x=0.45). The right column describes the mutation of a value near the edge of the search range (x=0.8).

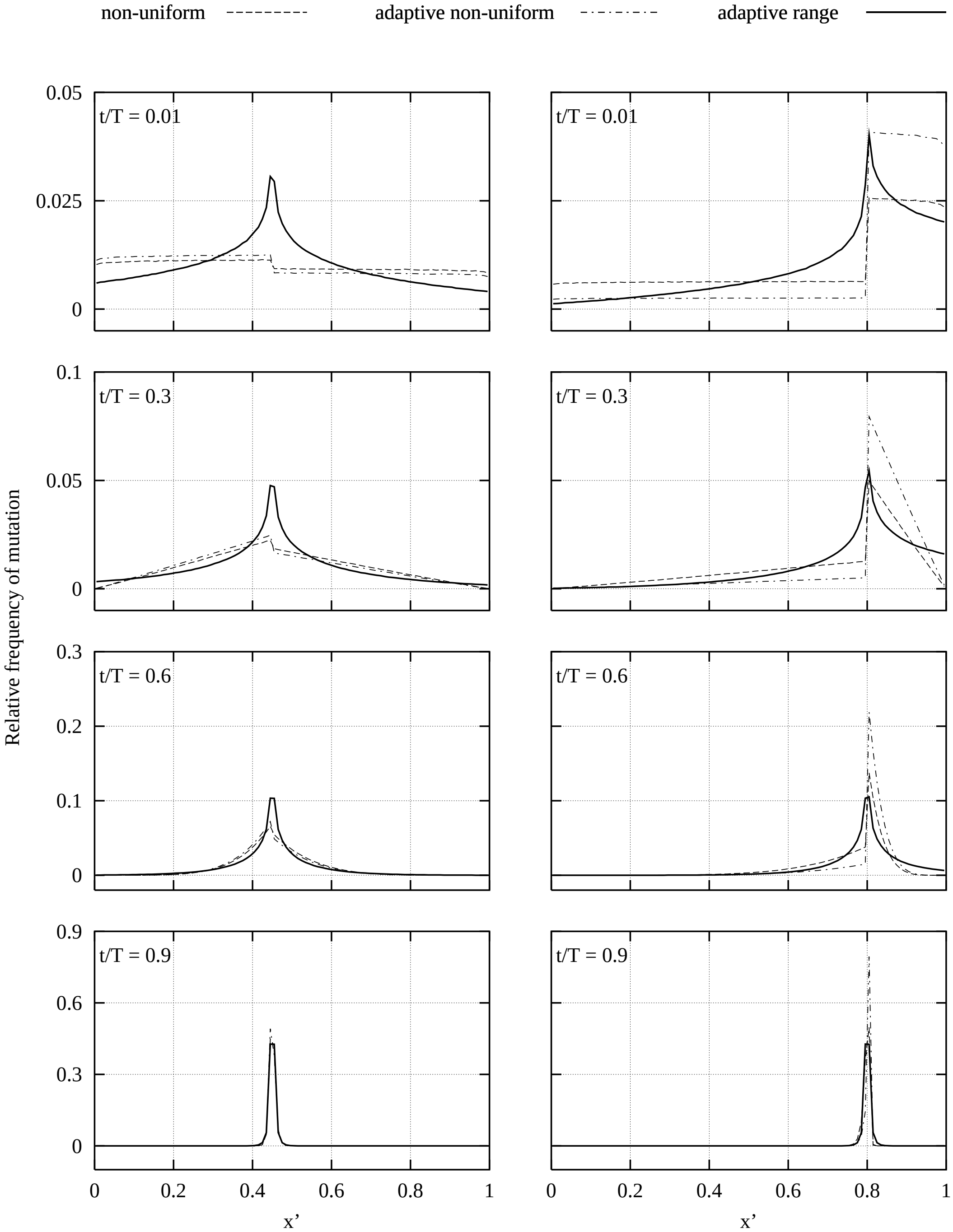


Figure A.6: A history of mutation profiles with $\beta = 2$. The left column shows the mutation of a value near the centre of the search range (x=0.45). The right column describes the mutation of a value near the edge of the search range (x=0.8).

evolutionary algorithms [55, 193, 26, 23], and include unimodal and multimodal functions. Each of the objective functions $f_i(\boldsymbol{x})$, are generalizable to an arbitrary dimension n, with the lowest minimum in the search range denoted by $f_i^*(\boldsymbol{x}^*)$. For each of the following functions a two-dimensional version is plotted in Figure A.7.

Sphere Model, f_1:

The sphere model is a continuous, convex, unimodal function [193]. It has been used in all fields of evolutionary algorithms, providing a test for convergence velocity. The topology of the two-dimensional sphere model is shown in Figure A.7(a). An additional generalization of the model places the minimum objective at $\boldsymbol{x}_{\text{min}}$.

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} (x_i - x_{\text{min},i})^2, \quad \text{for } n = 30 \tag{A.9}$$

where

$$-10 \leq x_i \leq 10 \;\; \forall i \;\; ; \;\; \boldsymbol{x}^* = (x_{\text{min},i}, \ldots, x_{\text{min},n}) \;\; ; \;\; f_1(\boldsymbol{x}^*) = 0.$$

Neubauer [162] used this function to illustrate the performance improvement of his adaptive mutation scheme over the original non-uniform operator. Two solution vectors were used in the experiments: (a) at the centre of the search range $x_{\text{min},i} = 0 \; \forall i$, and (b) near the boundary $x_{\text{min},i} = 8 \; \forall i$.

Step Function, f_2:

The step function is generated by discretizing the sphere model to introduce small plateaus to the topology. With $\lfloor x \rfloor$ denoting the largest integer value less than or equal to x, the step function is formalized as follows:

$$f_2(\boldsymbol{x}) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2, \quad \text{for } n = 30 \tag{A.10}$$

where

$$-100 \leq x_i \leq \;\; \forall i \;\; 100 \;\; ; \;\; \boldsymbol{x}^* = ([-0.5, 0.5))^n \;\; ; \;\; f_2(\boldsymbol{x}^*) = 0.$$

Generalized Rosenbrock Function, f_3:

Rosenbrock developed a method for the finding the greatest or least value of a function of several variables [182]. To evaluate the method he used a continuous, unimodal, bi-quadratic function of two variables, generally referred to as the Rosenbrock function. It formed part of the function set used by De Jong [55] and has since been generalized for n variables. The difficulty that evolutionary algorithms have with the Rosenbrock function is finding the global minimum within the flat valley, see Figure A.7(c).

$$f_3(\boldsymbol{x}) = \sum_{i=1}^{n-1} 100\left(x_{i+1} - x_i^2\right)^2, + (x_i - 1)^2, \quad \text{for } n = 30, \tag{A.11}$$

where

$$-30 \le x_i \le 30 \ \ \forall i \ ; \ \boldsymbol{x}^* = (1, \ldots, 1) \ ; \ f_3(\boldsymbol{x}^*) = 0.$$

Ackley's Function, f_4:

The function presented here is a generalized version [23] of the continuous, multimodal function by Ackley [3]. Ackley's function is obtained by modulating an exponential function with a cosine wave of moderate amplitude. The term $20 + e$ is added to move the global minimum function value to zero.

$$f_4(\boldsymbol{x}) = 20\text{exp}\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \text{exp}\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e, \quad \text{for } n = 30, \tag{A.12}$$

where

$$-32 \le x_i \le 32 \ ; \ \forall i \ ; \ \boldsymbol{x}^* = (0, \ldots, 0) \ ; \ f_4(\boldsymbol{x}^*) = 0.$$

Schwefel's Function, f_5:

This function is from Schwefel's catalogue of functions [193]. It is a multimodal function characterized by the second-best minimum being far away from the global minimum.

$$f(\boldsymbol{x}) = -\sum_{i=1}^{n}\left(x_i \text{sin}\left(\sqrt{|x_i|}\right)\right), \quad \text{for } n = 30 \tag{A.13}$$

where

$$-500 \le x_i \le 500 \ ; \ \forall i \ ; \ \boldsymbol{x}^* = (420.9687, \ldots, 420.9687) \ ; \ f_5(\boldsymbol{x}^*) = -12569.5.$$

Fletcher Powell Function, f_6:

First introduced by Fletcher and Powell in 1963 [72], this function is also multimodal. The objective function lacks any symmetry due to the use of random matrices $\boldsymbol{A}$ and $\boldsymbol{B}$

in the definition of the problem [193, 23]:

$$\begin{aligned} f_6(\boldsymbol{x}) &= \sum_{i=1}^{n}(A_i - B_i(x))^2 \\ A_i &= \sum_{j=1}^{n}(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j) \\ B_i(x) &= \sum_{j=1}^{n}(a_{ij}\sin x_j + b_{ij}\cos x_j) \end{aligned} \tag{A.14}$$

where a_{ij} and b_{ij} are random numbers in the range [-100,100], and α_i are random numbers in the range $[-\pi, \pi]$. The search domain and minimum for this function are

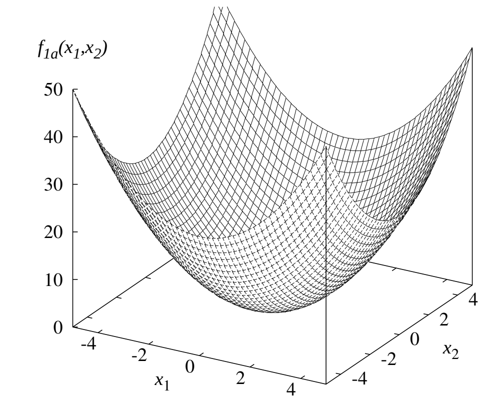
$$n = 10 \;\; ; \;\; -\pi \leq x_i \leq \pi \;\; \forall i \;\; ; \;\; \boldsymbol{x}^* = \boldsymbol{\alpha} \;\; ; \;\; f_6(\boldsymbol{x}^*) = 0$$

For the experiments presented in this appendix the following arrays were generated:

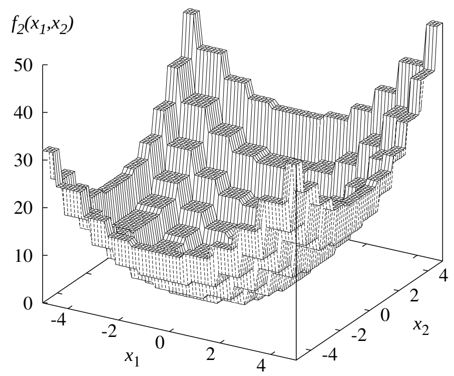
$$\boldsymbol{A} = \begin{bmatrix} -93.40 & 6.84 & 78.21 & 86.33 & 43.16 & 6.22 & -45.73 & 66.68 & 33.68 & 6.83 \\ 6.79 & 58.18 & -23.22 & 46.42 & 4.20 & 99.65 & 63.18 & 59.25 & -93.57 & 84.43 \\ -86.65 & 66.28 & -74.64 & -94.09 & 91.64 & -1.43 & -26.70 & -2.04 & -97.17 & -35.26 \\ 58.73 & -7.84 & -24.44 & 48.66 & -90.76 & 36.80 & -31.89 & 96.79 & 28.27 & 5.97 \\ -60.54 & 42.88 & -44.92 & 35.91 & -47.90 & 64.18 & 14.16 & 39.90 & -90.19 & -60.45 \\ -14.33 & -9.31 & 83.47 & 79.77 & 71.49 & 53.53 & -93.25 & 28.77 & 46.80 & -43.83 \\ 69.70 & 45.59 & -36.21 & -19.19 & -35.80 & -82.60 & 24.83 & 62.03 & 23.65 & -46.67 \\ 95.93 & 72.50 & -8.11 & 86.77 & -7.72 & 38.13 & 92.16 & -67.77 & -69.95 & 27.94 \\ 18.16 & 77.97 & -4.42 & 19.13 & -84.16 & -78.21 & -56.08 & -91.17 & -15.13 & 51.67 \\ 36.26 & -44.86 & -69.11 & -69.85 & 1.60 & 67.22 & -73.33 & -35.40 & 78.07 & 77.54 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} 74.88 & 26.32 & -48.49 & -44.55 & -3.32 & -63.31 & 20.61 & -54.39 & 5.81 & -69.56 \\ 52.24 & 35.22 & -50.37 & -73.16 & -30.27 & -23.08 & 23.49 & 86.77 & 98.68 & 88.58 \\ 78.71 & -81.98 & 24.43 & 70.03 & 47.91 & 55.45 & -93.05 & 43.65 & -81.04 & 79.90 \\ 8.10 & -76.66 & 54.33 & -79.75 & -0.09 & -61.20 & -91.78 & -15.11 & 78.82 & 43.59 \\ -10.17 & 18.19 & -68.67 & 11.21 & -49.98 & 79.34 & -23.46 & -1.83 & -71.43 & -22.36 \\ -95.68 & 15.98 & -35.93 & -11.43 & -16.95 & 27.80 & 94.55 & 78.00 & 35.65 & 38.85 \\ -77.73 & 53.14 & 17.53 & 1.42 & 99.05 & 72.10 & -14.06 & 36.13 & 26.31 & 89.11 \\ 11.17 & -90.36 & 30.96 & -65.62 & 47.76 & -86.56 & -31.73 & 65.86 & -78.56 & -43.88 \\ -6.06 & 31.87 & -77.96 & -13.89 & 48.89 & -53.43 & -68.43 & 53.42 & -68.44 & -49.24 \\ 42.93 & -46.06 & 7.97 & -59.45 & -92.81 & 73.27 & 74.14 & 28.58 & -16.95 & 1.89 \end{bmatrix}$$

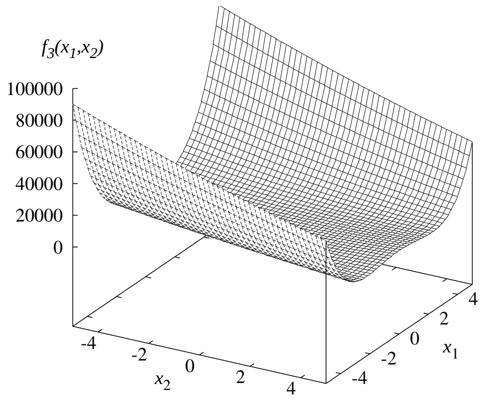
$$\boldsymbol{\alpha} = \begin{bmatrix} 1.35 & -2.64 & -2.70 & -1.36 & -2.89 & 2.87 & 2.88 & 2.37 & 1.96 & -0.63 \end{bmatrix}.$$



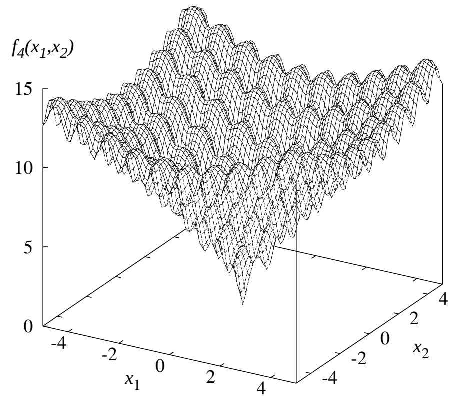
(a) Sphere function, f_1.



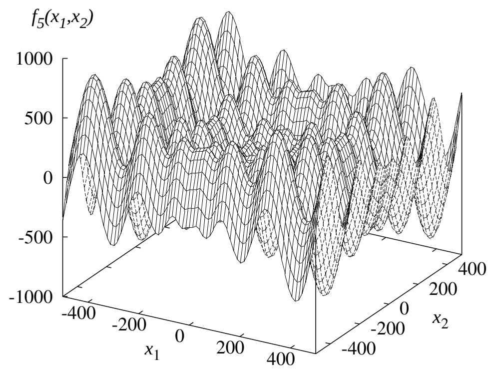
(b) Step function, f_2.



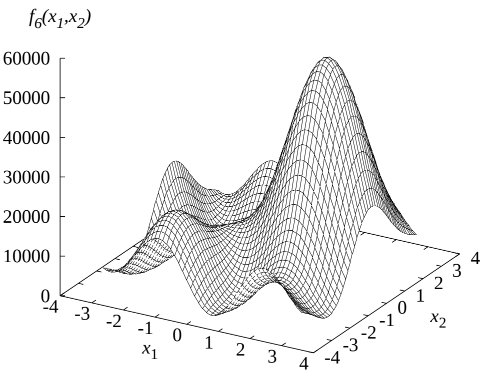
(c) Rosenbrock's function, f_3.



(d) Ackley's Function, f_4.



(e) Schwefel's function, f_5.



(f) Fletcher-Powell Function, f_6.

Figure A.7: Topology of two-dimensional versions of the minimization test functions.

A.5 Experiments

For the functions defined in the previous section, each minimization experiment involved running the genetic algorithm 50 times. The following algorithm parameters were used: population size $N_{\mathrm{P}} = 30$, generations $N_{\mathrm{G}} = 1000$, the crossover probability $p_{\mathrm{c}} = 0.6$, the mutation probability $p_{\mathrm{m}} = 0.2$, and the strategy parameter $\beta = 5$. These values represent a compromise on the overall performance on the six test functions. With the evolution taking place over a relatively small number of generations, the emphasis is on a fast global search. The three mutation operators are tested: Michalewicz's non-uniform mutation, Neubauer's adaptive non-uniform, and the adaptive range operator described in Section A.3. Each test result is represented by an average of the best objective function from each of the 50 experiments, f^*_{avg}, the standard deviation of the best solution σ_{f^*}, and the best overall solution, f^*_{best}.

Tables A.1 and A.2 summarize the performance of mutation operators when used in combination with arithmetic crossover and single point crossover [154]. Arithmetic crossover produces values bounded by the parent values, and as such offers superior fine-tuning. It also provides a mechanism for the reintroduction of parameter values which may have been lost to the extremes of the search domain through the bias of the adaptive non-uniform mutator. However, the generations needed for this reintroduction means that the time for which fine-tuning can improve the solution precision is reduced. The experiments run with the single point crossover isolate the behaviour of the mutation operation by removing the exploitation bias of the arithmetic crossover. The simple transfer of parent features provides greater access to the search domain which can aid the global search on some functions.

Table A.1: Performance of the mutation operators in combination with whole arithmetic crossover, on a set of benchmark functions. The mean and standard deviation apply to the best solution f^*, from each of the 50 runs, and f^*_{best} is the best solution found.

Function	Non-uniform			Adaptive non-uniform			Adaptive range		
	f^*_{avg}	std. dev.	f^*_{best}	f^*_{avg}	std. dev.	f^*_{best}	f^*_{avg}	std. dev.	f^*_{best}
f_{1a}	2.716e-7	1.643e-7	4.43e-9	2.086e-7	1.36e-7	4.543e-8	8.198e-7	5.573e-7	1.540e-7
f_{1b}	2.25	1.13	0.740	0.2	0.734	0.00278	8.778e-7	5.387e-7	6.891e-8
f_2	0	0	0	0	0	0	0	0	0
f_3	39.37	37.09	27.32	1311.6	6064.8	26.43	116.6	174.7	26.78
f_4	9.303e-4	3.012e-4	3.844e-4	8.438e-4	3.229e-4	3.0e-4	1.611e-3	5.564e-4	6.688e-4
f_5	-9658.8	610.0	-10680.1	-10336.2	585.4	-11615.6	-10263.6	404.8	-11186.2
f_6	1253.2	2630.3	5.608	2689.8	3714.8	16.4	735.9	1263.0	0.0707

The sphere function f_1 emphasizes convergence velocity on convex problems. Though not a test for the global search capability of the algorithm, it provides a simple examina-

Table A.2: Performance of the mutation operators in combination with single point crossover, on a set of benchmark functions. The mean and standard deviation apply to the best solution f^*, from each of the 50 runs, and f^*_{best} is the best solution found.

Function	Non-uniform			Adaptive non-uniform			Adaptive range		
	f^*_{avg}	std. dev.	f^*_{best}	f^*_{avg}	std. dev.	f^*_{best}	f^*_{avg}	std. dev.	f^*_{best}
f_{1a}	2.514e-6	1.247e-6	6.505e-7	48.25	64.04	5.086e-7	5.712e-6	2.514e-6	1.438e-6
f_{1b}	0.151	0.0813	0.0245	32.6	10.25	15.71	4.91e-6	4.1e-6	1.30e-6
f_2	0	0	0	7660.7	7533.8	0	0	0	0
f_3	174.8	272.8	20.33	1.28e7	2.96e7	26.4	304.2	587.8	20.12
f_4	2.778e-3	9.084e-4	1.403e-3	12.56	8.148	2.181e-3	1.386e-3	4.623e-4	6.007e-4
f_5	-10761.2	366.3	-11497.7	-10479.5	539.7	-11495.8	-11019.7	428.3	-11799.6
f_6	1327.0	1434.1	0.0505	5778.7	4576.9	324.4	2139.1	2522.8	0.0196

tion of the search bias of the mutation and crossover operators. With the solution located in the centre of the search domain, greater solution precision is provided by the non-uniform operator of Michalewicz, due to the search bias and the lower maximum mutation possible for a given generation. For the edge solution (f_{1a}), the average error in the solution values from the 50 runs was 0.224, compared to 7.381×10^{-5}. The drop in solution precision reflects the time need to counter the mutation bias. In comparison, the adaptive range operator performance appears indifferent to the position of the solution, with solution errors of 1.321×10^{-4} and 1.273×10^{-4} for f_{1a} and f_{1b} respectively. Due to the mutation range being consistently scaled off the domain boundary, the maximum mutation possible throughout the evolution is greater than for the Michalewicz operator, and the solution precision for the f_{1a} is marginally reduced. As expected, the arithmetic crossover provides greater solution precision for the sphere problem, due to its exploitive nature.

The experimental verification performed by Neubauer used the sphere function, with the same dimension, solution set, and search range of f_{1a} and f_{1b}. It is worth considering therefore, why those experiments supported the adaptive non-uniform operator, while the results contained in Tables A.1 and A.2 clearly do not. In the results presented by Neubauer [162], the non-uniform and the adaptive non-uniform operator were shown to provide equivalent solutions for f_{1a}. For the edge solution of f_{1b}, the modified mutation operator of Neubauer provided superior results, though interestingly, a three orders of magnitude drop in the precision of the best function. The results generated benefited from the large number of generations, 10 000 were used, and the use of $\beta = 5$ to rapidly reduce the chances of large mutations. If there were no bias to the location of the solution in the search domain, then it would be reasonable to expect a similar return for the best function value. A similar drop in solution precision was observed when the RCGA used in this thesis was run over 10 000 generations. However, out of 100 test runs for f_{1a}, 9 returned solutions with one or two values on the search boundary rather than on the actual solution. The affect can be explained by search corruption due to the repeated cycle of mutation bias with crossover recovery. The single point crossover has no means of reintroducing values lost through mutation bias so, as shown in Table A.2, the solutions are consistently poor. Therefore, with due consideration to the simulation setup and the relative performance between the edge and central solution, the assertion that the modified mutation operator is more effective is inappropriate.

By discretizing the sphere model, as in f_2, the search is made considerably easier. In most cases the population converged to the global minimum within 800 generations. If the search length is reduced to 500 generations, the global minimum could likewise be found within 80 % of the available generations.

The non-uniformity of the mutations across the generations is expressed through β. In function minimization test problems this usually set to $\beta = 5$ for the fine-tuning benefits. When combined with arithmetic crossover the search can be hindered if the solution is away from the centre of the search domain. For functions with the solution located in near the centre of the search domain, f_{1a}, f_2, f_3, and f_4, the bias of the non-uniform operator provides superior results. This results in a reduction of the fine-tuning capability of the mutation operation. In the case of the flight control problem for this thesis, it was more desirable to remove search bias than to maintain the fine-tuning.

One of the potential concerns of the GA application to the flight control problem, is the sensitivity of the genetic algorithm to parameter epistatis. Epistatis refers to the interaction of variables with respect to the fitness function. Of the functions tested here, the Rosenbrock function exhibits a dependency on the relative value of neighbouring parameters. Results from Salomon [190] indicate that the high global convergence performance of the genetic algorithm relies on the independence of the parameters. For control problems the fitness of a set of parameters is generally highly dependent on the interaction of the parameters, and could therefore form a deceptive problem for GAs. It is not clear from the tests done whether the performance of the GA on the Rosenbrock is due to parameter epistatis or the general difficulty of finding the global minimum within the relatively flat valley that contains it.

None of the experiments conducted on the Rosenbrock function were able to find the global minimum. Tuning the parameterization of the RCGA can reduce the variation amongst the solutions found, but the global minimum remains elusive. Of the 21 forms of RCGA tested by Herrera *et al.* [99], only one provided an objective function which indicated a global optimum. The implementation included non-uniform mutation, a fuzzy connectives based crossover, and a very low probability of mutation ($p_m = 0.005$). Evolutionary strategies are suggested to be unaffected by epistatis. Using a population size of 100 and 20 000 generations, Yao *et al.* [236] reports a mean best from 50 runs of 5.06.

Table A.3: Solution quality for f_5. $N(x_i^*)$ represents the number of values per solution within ± 10 of x_i^* and $\bar{x}_{i,\text{err}}$ describes the average error of those values.

Mutation	$N(x_i^*)$	$\bar{x}_{i,\text{err}}$
non-uniform	18.16/30	1.984
adaptive-range	19.34/30	0.0357

Simulated annealing has also been applied with some success to the Rosenbrock function [197].

Ackley's function f_4 was generally well solved by the various combinations of crossover and mutation. If the search range is shifted such that the global minimum is no longer centrally located, the performance trend observed with the sphere functions is duplicated. For example, by setting the search range to [-10,30], f^*_{avg} using the Michalewicz operator becomes 0.53, while the adaptive-range operator maintains the solution accuracy with $f^*_{\text{avg}} = 1.03 \times 10^{-3}$.

Apart from the deceptive nature of the Rosenbrock function, functions f_5 and f_6 represent the most challenging problems in the set considered here. The results presented in Tables A.1 and A.2 are merely to compare the mutation operators, and do not represent the best arrangement for the RCGA. For both functions, the adaptive range operator offers improved performance over Michalewicz's original form. By examining the average error of those solution values which are near the global minimum ($x^*_{i,\text{err}}$), the behavior of the two operators is clear. If a near optimal value is within $\pm\, 10$ of the global optimum, then the results in Table A.3 shows the average number of near optimum values per run and the average error in those values. The differences in the function values reported in Table A.2 are therefore due to the reduction in solution precision due to bias of the non-uniform mutation.

Knowing the general properties of the Schwefel function, an algorithm configuration can readily be formed to dramatically improve the global search performance. Primarily, searching Schwefel's function benefits from a mix of exploitation and exploration in the crossover operator, so that the corner solution can be reliably reached. The BLX-α operator from Eshelman [67] provides an extension of the crossover range for a given set of parent values. It was implemented in the RCGA to provide two offspring, $\boldsymbol{C}'_1 and \boldsymbol{C}'_2$, from two parents $\boldsymbol{C}_j = (x_{1,j}, \ldots, x_{i,j}, \ldots, x_{n,j})\; j = 1, 2$. The BLX-$\alpha$ operator uniformly picks the new individual values $x'_{i,j}$ from the interval $[c_{\text{min}_i} - \alpha I_i, c_{\text{max},i} + \alpha I_i]$, where $c_{\text{min},i} = \min(x_{i,1}, x_{i,2})$, $c_{\text{max},i} = \max(x_{i,1}, x_{i,2})$, and $I_i = c_{max,i} - c_{min,i}$. The offspring values are therefore expressed as follows:

$$x'_{i,j} = c_{\text{min},i} + r_j\,(c_{\text{max},i} - c_{\text{min},i}),\ \text{for } i = 1, n;\ j = 1, 2, \tag{A.15}$$

Table A.4: Improved performance of RCGA for f_5 using the BLX-0.5 crossover operator. The results are taken from 50 runs with $(N_P = 30,\ N_G = 1000,\ p_c = 0.6,\ p_m = 0.01,\ \beta = 5)$.

Mutation	f^*_{avg}	std. dev.	f^*_{best}	$N(x^*_i)$	$\bar{x}_{i,\text{err}}$
non-uniform	-12175.0	314.7	-12569.3	28.34/30	0.3442
adaptive-range	-12345.6	236.3	-12569.5	28.68/30	0.0569

where r_1 and r_2 are uniform random numbers $\in [0, 1]$.

Using the BLX-α crossover operator, the results in Table A.4 were generated. Like the previous results, the effect of the mutation bias when using the non-uniform operator is a reduction in the solution precision when compared to the modified operator. It was also noted that the return of the global minimum from 50 runs is insensitive to the mutation rate.

Compared to the large number of published results for experiments involving the Schwefel function, there are far fewer examples of experiments with high order Fletcher-Powell functions. Bäck *et al.* [25] used the Fletcher-Powell function with $n = 30$ to compare the performance of evolutionary strategies with a binary-coded genetic algorithm. A population size of 100 was used and the search performed over 2000 generations. In a paper by Takahashi *et al.* [216] a real-coded genetic algorithm was applied to the Fletcher-Powell function for the purpose of examining an extension to the unimodal normal distribution crossover. A remarkable feature of the experiments was the use of populations ranging in size from 1500 to 12 000. The application of the RCGA in this thesis is targeting the rapid development of solutions, which it seems, is not readily attainable for the Fletcher-Powell function. It would also seem, from Bäck *et al.* [25], that the greater degree of freedom resulting from working with n different self-adaptive mutation parameters per individual is a significant advantage over the single uniform mutation rate typically used for genetic algorithms.

The purpose here is not to compete with alternative algorithms, but to provide an assessment of a modification to a flawed mutation operator. Results presented for the 10 variable Fletcher-Powell case, further support the adaptive range operator. Additional experiments showed that over 500 generations near optimal solutions were generated for mutation rates ranging from 0.05 to 0.9 and for the three crossover operators previously discussed (single point, arithmetic and BLX-α).

The issue of parameterization of evolutionary algorithms for optimal search performance has been addressed in many ways [64, 83]. Heuristic rules have been developed (typically applying to a certain class of problems), varying the activation rates of the perturbation operators with time has been recommended, self-adaptive schemes are used by evolutionary strategies, and parameters can be included in the the self-adaption of parameters during the evolutionary process. It is inevitable however that a trial and error process is applied, as one set of evolutionary mechanisms and parameters cannot be universally superior.

Across variations in population size, generation scale, mutation probabilities and crossover types, the modification to Michalewicz's mutation operator provided a general improvement in performance. One significant observation was that the adaptive-range operator is able to generate near optimal solutions out of set of experiments, for a broad range of mutation rates, generation scales and crossover types.

A.6 Summary

The aim of this appendix was to demonstrate the performance advantage of an alternative definition of the non-uniform mutation operator proposed by Michalewicz, for real-coded genetic algorithms. Established operators, through their search bias, were shown to adversely affect the reliability and precision of the algorithm. They are sensitive to the algorithm structure and the setting of parameters such as population size, mutation probability, and the rate of fine-tuning. In some cases, effective fine-tuning was delayed by the extra time needed to generate gene values lost by the bias of the mutation operator. By redefining the non-uniform mutation operator as an *adaptive range* mutation, the general performance of the genetic algorithm was improved. Though the objective was not the measuring of the RCGA performance with other evolutionary algorithms, favourable comparisons can be made with published results included those using genetic algorithms, evolutionary programming, and evolutionary strategies.

Favourable comparisons can be made with other published results such that the application of genetic algorithms to real-valued parameter optimization should not be discounted, see for example [42, 26].

Bibliography

[1] ——, *U.S. Standard Atmosphere*, U.S. Government Printing Office, Washington, D.C., 1976.

[2] Abzug, M. J. and Larrabee, E. E., *Airplane Stability and Control : A History of the Technologies That Made Aviation Possible*, Cambridge Aerospace Series ; 6, Cambridge University Press, 1997.

[3] Ackley, D. H., *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers, Boston, 1987.

[4] Allen, G. A. J., "Parametric study on thrust production in the two dimensional scramjet," Tech. Rep. AIAA Paper 91-0227, AIAA 29th Aerospace Sciences Meeting, 1991.

[5] Anderson, G., Kumar, A., and Erdos, J., "Progress in hypersonic combustion technology with computation and experiment," AIAA Paper 90-5254, AIAA Second International Aerospace Planes Conference, 1990.

[6] Anderson, J. D., *Hypersonic and High Temperature Gas Dynamics*, McGraw-Hill, 1989.

[7] Anderson, J. D., *Introduction to Flight*, McGraw-Hill Inc., 1989.

[8] Anderson, J. D., *Modern Compressible Flow : With Historical Perspective*, McGraw-Hill, New York, 2nd ed., 1990.

[9] Anderson, J. D., *A History of Aerodynamics and Its Impact on Flying Machines*, Cambridge University Press, 1997.

[10] Anderson, M. R., Emami-Naeini, A., and Vincent, J. H., "Robust control law development for a hypersonic cruise aircraft," *Proceedings of the 10th American Control Conference*, Vol. 1, IEEE, 1991, pp. 839–845.

[11] Antsaklis, P., "Defining intelligent control: Report of the task force on intelligent control," *IEEE Control Systems Magazine*, Vol. 14, No. 3, June 1994, pp. 4–5 & 58–66.

[12] Armengaud, F., Decher, R., Koopman, A., and Lafosse, B., "One dimensional modeling of hypersonic flight propulsion engines," AIAA Paper 89-2026, 1989.

[13] Ashby, G. C., Bernot, P. T., and Woods, W. C., "Pressure distributions on a 0.02-scale space shuttle orbiter nose at mach 21.5 in helium," Tech. Rep. NASA TM-4495, Langley Research Center, 1994.

[14] Åström, K. L., "Adaptive control around 1960," *Proceedings of the 34th Conference on Decision and Control*, IEEE, New Orleans, LA, December 1995, pp. 2784–2789.

[15] Atmar, W., "Notes on the simulation of evolution," *IEEE Transactions on Neural Networks*, Vol. 5, No. 1, January 1994, pp. 130–147.

[16] Austin, K. J. and Jacobs, P. A., "A newtonian solver for hypersonic flows," Technical Report 5/96, Dept of Mechanical Engineering, The Univeristy of Queensland, 1996.

[17] Austin, K. J. and Jacobs, P. A., "Benchmark control problem - evolutionary design of a robust controller," Technical Report 2002/5, Department of Mechanical Engineering: The University of Queensland, 2002.

[18] Austin, K. J. and Jacobs, P. A., "Numerical flight simulation of a hypersonic air-breathing vehicle," Technical Report 2002/3, Department of Mechanical Engineering: The University of Queensland, 2002.

[19] Austin, K. J. and Jacobs, P. A., "A real-coded genetic algorithm for function optimization," Technical Report 2002/4, Department of Mechanical Engineering: The University of Queensland, 2002.

[20] Baals, D. D. and Corliss, W. R., *Wind Tunnels of NASA*, National Aeronautics and Space Administration, Washington, D. C., 1981.

[21] Bäck, T., "Optimal mutation rates in genetic search," *Proceedings of the Fifth International Conference on Genetic Algorithms*, 1993, pp. 2–8.

[22] Bäck, T., "Selective pressure in evolutionary algorithms: A characterization of selection mechanisms," *First IEEE Conference on Evolutionary Computation*, IEEE Press, 1995, pp. 57–62.

[23] Bäck, T., *Evolutionary Algorithms in Theory and Practice : Evolutionary Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, New York, 1996.

[24] Bäck, T., Hammel, U., and Schwefel, H.-P., "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, April 1997, pp. 3–17.

[25] Bäck, T., Schütz, M., and Khuri, S., "Evolution strategies: An alternative evolutionary computation method," *Artificial Evolution*, edited by J. M. Alliot, E. Lutton, E. Ronald, M. Schoenhauer, and D. Snyers, Springer, Berlin, 1996.

[26] Bäck, T. and Schwefel, H.-P., "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, Vol. 1, No. 1, 1993, pp. 1–23.

[27] Beer, F. P. and Johnston, E. R., *Vector Mechanics for Engineers*, Vol. 2 Dynamics, McGraw-Hill Ryerson, 1981.

[28] Bessaou, M. and Siarry, P., "A new tool in electrostatics using a really-coded multipopulation genetic algorithm tuned through analytical test problems," *Advances in Engineering Software*, Vol. 32, 2001, pp. 363–374.

[29] Billig, F. S., "Research on supersonic combustion," *Journal of Propulsion and Power*, Vol. 9, No. 4, July-Aug 1993, pp. 499–514.

[30] Blickle, T., "Tournament selection," *Evolutionary Computation 1 : Basic Algorithms and Operators*, edited by T. Bäck, D. B. Fogel, and Z. Michalewicz, chap. 24, Institute of Physics Publishing, 2000, pp. 181–186.

[31] Booker, L. B., *Intelligent behaviour as an adaptation to the task environment*, Doctoral dissertation, University of Michigan, 1982, cited from Goldberg, 1989.

[32] Booker, L. B., Fogel, D., Whitley, D., Angeline, P. J., and Eiben, A. E., "Recombination," *Evolutionary Computation 1: Basic Algorithms and Operators*, edited by T. Bäck, D. B. Fogel, and Z. Michalewicz, chap. 33, Institute of Physics Publishing, 2000, pp. 256–307.

[33] Box, G. E. P., "Evolutionary operation: A method for increasing industrial productivity," *Journal of the Royal Statistical Society*, Vol. C 6, No. 2, 1957, pp. 81–101.

[34] Bremermann, H. J., "Optimization through evolution and recombination," *Self-Organizing Systems*, edited by M. C. Yovits, G. T. Jacobi, and G. D. Goldstein, Spartan, Washington DC, 1962.

[35] Brindle, A., *Genetic algorithms for function optimization*, Unpublished doctoral dissertation, University of Alberta, Edmonton, 1981, cited from Goldberg, 1989.

[36] Bryan, G. H., *Stability in Aviation*, MacMillan, London, 1911, cited in [2].

[37] Buschek, H. and Calise, A. J., "μ controllers: Mixed and fixed," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 34–41.

[38] Buschek, H. and Calise, A. J., "Uncertainty modeling and fixed-order controller design for a hypersonic vehicle model," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 42–48.

[39] Chamitoff, G. E., *Robust Intelligent Flight Control for Hypersonic Vehicles*, Ph.D. Thesis, Massachusetts Institute of Technology, 1992.

[40] Chavez, F. R. and Schmidt, D. K., "Analytical aeropropulsive/aeroelastic hypersonic vehicle model with dynamic analysis," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, 1994, pp. 1308–1319.

[41] Chavez, F. R. and Schmidt, D. K., "Uncertainty modeling for multivariable-control robustness analysis of elastic high-speed vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 1, Jan.–Feb. 1999, pp. 87–95.

[42] Chellapilla, K., "Combining mutation operators in evolutionary programming," *IEEE Transactions on Evolutionary Programming*, Vol. 2, No. 3, Sept. 1998, pp. 91–96.

[43] Christian, P. M., "Fuzzy logic control of a hypervelocity intercepter," *Second IEEE International Conference on Fuzzy Systems*, 1993, pp. 877–882.

[44] Corana, A., Marchesi, C., Martini, C., and Ridella, S., "Minimizing multimodal functions of continuous variables with the simulated annealing algorithm," *ACM Transactions on Mathematical Software*, Vol. 13, No. 2, September 1987, pp. 262–280.

[45] Council, N. R., *Hypersonic Technology for Military Application*, National Academy Press, 1990.

[46] Covault, C., Kandebo, S. W., and Lenorovitz, J. M., "Russian scramjet flight tests," *Aviation Week and Space Technology*, February 3 1992.

[47] Coverstone-Carroll, V., "Near-optimal low-thrust trajectories via micro-genetic algorithms," *Journal Guidnance, Control, and Dynamics*, Vol. 20, No. 1, 1996, pp. 196–198.

[48] Craddock, C., *Computational Optimization of Scramjets and Shock Tunnel Nozzles*, Ph.D. Thesis, Department of Mechanical Engineering, The University of Queensland, 1999.

[49] Cruz, C. I. and Wilhite, A. W., "Prediction of high-speed aerodynamic characteristics using the aerodynamic preliminary analysis system (APAS)," AIAA 89-2173, 1989.

[50] Curran, E. T., Heiser, W. H., and Pratt, D. T., "Fluid phenomena in scramjet combustion systems," *Annual Review of Fluid Mechanics*, Vol. 28, 1996, pp. 323–360.

[51] Darwin, C., *On the origin of species by means of natural selection*, London; Murray, 1859.

[52] Davidson, J., Lallman, F., McMinn, J. D., Martin, J., Pahle, J., Stephenson, M., Selmon, J., and Bose, D., "Flight control laws for NASA's Hyper-X research vehicle," AIAA Paper 99-4124, 1999.

[53] Davis, L., *Handbook of genetic algorithms*, Van Nostrand Reinhold, 1991.

[54] Davis, M. C., Pahle, J. W., White, J. T., Marshall, L. A., Mashburn, M. J., and Franks, R., "Development of a flush airdata sensing system on a sharp-nosed vehicle for flight at mach 3 to 8," Tech. Rep. AIAA Paper 2000-0504, 38th Aerospace Sciences Meeting and Exhibit.

[55] De Jong, K., *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*, Ph.D. Thesis, University of Michigan, 1975.

[56] De Jong, K., "Learning with genetic algorithms: An overview," *Machine Learning*, Vol. 3, 1988, pp. 121–138.

[57] de Selding, P. B., "Russians seek French Francs for engine test," *Space News*, October 18-24 1993.

[58] de Silva, C. W., *Intelligent Control : Fuzzy Logic Applications*, CRC Press, 1995.

[59] Denna, M., Mauri, G., and Zanaboni, A. M., "Learning fuzzy rules with tabu search - An application to control," *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 2, April 1999, pp. 295–318.

[60] Divan, P. E., "Aerodynamic analysis system for conceptual and preliminary analysis from subsonic to hypersonic speeds," AIAA 80-1897, 1980.

[61] Doyle, J. C., Glover, K., Khargonekar, P. P., and Francis, B. A., "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Transactions on Automatic Control*, Vol. 34, No. 8, August 1989, pp. 831–847.

[62] Doyle, J. C. and Stein, G., "Multivariable feedback design: Concepts for a classical/modern synthesis," *IEEE Transactions on Automatic Control*, Vol. AC-26, No. 1, February 1981, pp. 4–16.

[63] Driankov, D., Hellendoorn, H., and Reinfrank, M., *An Introduction to Fuzzy Control*, Springer-Verlag, New York, USA, 1993.

[64] Eiben, A. E., Hinterding, R., and Michalewicz, Z., "Parameter control," *Evolutionary Computation 2: Advanced Algorithms and Operators*, edited by T. Bäck, D. B. Fogel, and Z. Michalewicz, chap. 20, Institute of Physics Publishing, 2000, pp. 170–187.

[65] Engeln-Müllges, G. and Uhlig, F., *Numerical Algorithms with C*, Springer-Verlag, 1996.

[66] Engelund, W. C., Holland, S. D., Cockrell, C. E., and Bittner, R. D., "Propulsion system airframe integration issues and aerodynamic database development for the Hyper-X flight research vehicle," *Fourteenth Internatioal Symposium on Air Breathing Engines*, 1999.

[67] Eshelman, L. J. and Schaffer, J. D., "Real-coded genetic algorithms and interval-schemata," *Foundation of Genetic Algorithms 2*, edited by L. D. Whitley, Morgan Kaufmann, San Mateo, 1993, pp. 187–202.

[68] Etkin, B., *Dynamics of Atmospheric Flight*, Wiley, 1972.

[69] Falempin, F. H., "Overview of French research center ONERA activities on high-speed airbreathing propulsion," *12th International Symposium on Airbreathing Engines*, 1995, XII ISABE.

[70] Ferri, A., "Mixing-controlled supersonic combustion," *Annual Review of Fluid Mechanics*, Vol. 5, 1973, pp. 301–338.

[71] Ferri, A., Moretti, G., and Slutsky, S., "Mixing processes in supersonic combustion," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 13, No. 1, March 1965, pp. 229–258.

[72] Fletcher, R. and Powell, M. J. D., "A rapidly convergent descent method for minimization," *Computing Journal*, Vol. 6, 1963, pp. 163–168.

[73] Fogarty, T. C., "Varying the probability of mutation in the genetic algorithm," *Proceedings of the Third International Conference on Genetic Algorithms*, edited by J. D. Schaffer, Morgan Kaufmann, San Mateo, Calif., 1989, pp. 104–109.

[74] Fogel, D., *Evolutionary Computation I*, chap. Mutation Operators, Institute of Physics Publishing, Bristol and Philadelphia, 2000.

[75] Fogel, L. J., *Intelligence Through Simulated Evolution : Four Decades of Evolutionary Programming*, New York : Wiley, 1999.

[76] Fogel, L. J., Owens, A. J., and Walsh, M. J., *Artificial Intelligence Through Simulated Evolution*, New York: Wiley, 1966.

[77] Fonseca, C. M. and Fleming, P. J., "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, Vol. 3, No. 1, 1995, pp. 1–16.

[78] Fraser, S. M., "Calculation of Mach number from given turning angle in supersonic isentropic flow," *Aeronautical Journal*, February 1975, pp. 95–96.

[79] Freeman, D. C., Reubush, D. E., McClinton, C. R., Rausch, V. L., and Crawford, J. L., "The NASA Hyper-X program," *48th International Astronautical Congress*, 1997.

[80] Friedberg, R. M., "A learning machine: Part I," *IBM Journal*, Vol. 2, No. 1, Jan. 1958, pp. 2–13.

[81] Fukuda, T. and Kubota, N., *Fuzzy and Neuro-Fuzzy Systems in Medicine*, chap. Fuzzy Control Methodology: Basics and state of the art, CRC Press, 1999.

[82] Gentry, A. E., "Hypersonic arbitrary-body aerodynamic computer program (Mark III version), vol. 1-User's manual," Rep. DAC 61552, McDonnell-Douglas Corp., April 1968.

[83] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company Inc., 1989.

[84] Goldberg, D. E. and Deb, K., "A comparative analysis of selection schemes used in genetic algorithms," *Foundations of Genetic Algorithms*, edited by G. J. E. Rawlins, Morgan Kaufmann, San Mateo, Calf., 1991.

[85] Gracey, W. H., "Measurement of aircraft speed and altitude," NASA RP-1046, 1980.

[86] Grefenstette, J., "Proportional selection and sampling algorithms," *Evolutionary Computation 1 : Basic Algorithms and Operators*, edited by T. Bäck, D. B. Fogel, and Z. Michalewicz, chap. 23, Institute of Physics Publishing, 2000, pp. 172–180.

[87] Grefenstette, J., "Rank-based selection," *Evolutionary Computation 1 : Basic Algorithms and Operators*, edited by T. Bäck, D. B. Fogel, and Z. Michalewicz, chap. 25, Institute of Physics Publishing, 2000, pp. 187–194.

[88] Grefenstette, J. J., "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 16, 1986, pp. 122–128.

[89] Grefenstette, J. J., "Incorporating problem specific knowledge in GA's," *Genetic Algorithms and Simulated Annealing*, edited by L. Davis, Morgan Kaufmann Publishers, 1987, pp. 42–60.

[90] Grefenstette, J. J., "Credit assignment in rule discovery systems based on genetic algorithms," *Machine Learning*, Vol. 3, 1988, pp. 225–245.

[91] Gregory, I. M., Chowdhry, R. S., McMinn, J. D., and Shaughnessy, J. D., "Hypersonic vehicle model and control law development using h_∞ and μ synthesis," NASA tm-4562, Langley Research Center, NASA, October 1994.

[92] Griffith, B. J., Maus, J. R., and Majors, B. M., "Addressing the hypersonic simulation problem," *Journal of Spacecraft and Rockets*, Vol. 24, No. 4, 1987, pp. 334–341.

[93] Gupta, M. M., "Part V: Applications to aircraft control problems," *Adaptive Methods for Control System Design*, edited by M. M. Gupta, IEEE, 1986.

[94] Hancock, P. J. B., "A comparison of selection mechanisms," *Evolutionary Computation 1 : Basic Algorithms and Operators*, edited by T. Bäck, D. B. Fogel, and Z. Michalewicz, chap. 29, Institute of Physics Publishing, 2000, pp. 212–227.

[95] Harris, R., "On the threshold - the outlook for supersonic and hypersonic aircraft," AIAA 89-2071, 1989, AIAA/AHS/ASEE Aircraft Design, Systems and Operations Conference.

[96] Hattis, P., Malchow, H., Shaugnessy, J., and Chowdhry, R., "Integrated trajectory and control analysis for generic hypersonic vehicles," *AIAA Third International Aerospace Planes Conference*, 1991, AIAA Paper 91-5052.

[97] Heiser, W. H. and Pratt, D. T., *Hypersonic Airbreathing Propulsion*, AIAA Education Series, Washington, USA, 1994.

[98] Henry, J. R. and Anderson, G. Y., "Design considerations for the airframe-integrated scramjet," *First International Symposium on Air Breathing Engines*, 1972.

[99] Herrera, F., Lorenzo, M., and Verdegay, J. L., "Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis," *Artificial Intelligence Review*, Vol. 12, No. 4, 1998, pp. 265–319.

[100] Hicks, J. W., "International efforts scram into flight," *Aerospace America*, June 1998, pp. 28–33.

[101] Holland, J. H., *Adaption in Natural and Artificial Systems : An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, Ann Arbor : University of Michigan Press, 1975.

[102] Holmblad, L. P. and Ostergaard, J. J., *Fuzzy Information and Decision Processes*, chap. Control of a Cement Kiln by Fuzzy Logic, 1982.

[103] Homaifar, A. and McCormick, E., "Simulaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms," *IEEE Transactions on Fuzzy Systems*, Vol. 3, No. 2, May 1995, pp. 129–139.

[104] Huber, P. W., Schexnayder, C. J., and McClinton, C. R., "Criteria for self-ignition of supersonic hydogren-air mixtures," NASA Technical Paper 1457, 1979.

[105] Hunt, J. L., "Hypersonics - volume 1. Defining the hypersonic environment," *Hypersonic airbreathing vehicle design (focus on aero-space plane)*, edited by J. J. Bertin, R. Glowinski, and J. Periaux, Birkhauser, Boston, 1989, pp. 205–262.

[106] Hunt, J. L., "Systems challenges for hypersonic vehicles," AGARD interpanel symposium on future aerospace technology in service to the alliance, NASA Langley Research Center, 1997.

[107] Isaka, S. and Sebald, A. V., "An optimization approach for fuzzy controller design," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 22, No. 6, Nov./Dec. 1992, pp. 1469–1473.

[108] Isakowitz, S. J., *International Reference Guide to Space Launch Systems*, AIAA, 1995.

[109] Isakowitz, S. J., Hopkins, J. P., and Hopkins, J. B., *International Reference Guide to Space Launch Systems*, American Institute of Aeronautics and Astronautics, 3rd ed., 1999.

[110] Jacobs, P. A., "MB_CNS: A computer program for the simulation of transient compressible flows," Tech. Rep. 6/96, Department of Mechanical Engineering, The University of Queensland, Brisbane, July 1996.

[111] Jacobs, P. A., "MB_CNS: A computer program for the simulation of transient compressible flows," Tech. Rep. 7/98, Department of Mechanical Engineering, The University of Queensland, Brisbane, June 1998, 1998 Update.

[112] Jacobs, P. A. and Craddock, C. S., "Simulation and optimization of heated, inviscid flows in scramjet ducts," *Journal of Propulsion and Power*, Vol. 15, No. 1, Jan.–Feb. 1999, pp. 73–81.

[113] Jacobs, P. A., Macrossan, M. N., and Stalker, R. J., "ARC collaborative research grant, final report - effects of scale on scramjet vehicles," Report 13/96, Department of Mechanical Engineering, The University of Queensland, September 1996.

[114] Jager, R., *Fuzzy Logic in Control*, Ph.D. Thesis, Delft University of Technology, 1995.

[115] Jakab, P. L., *Visions of a Flying Machine : The Wright Brothers and the Process of Invention*, Smithsonian Institution Press, 1990.

[116] Janikow, C. Z. and Michalewicz, Z., "An experimental comparison of binary and floating point representations in genetic algorithms," *Proceedings of the Fourth International Conference on Genetic Algorithms*, 1991, pp. 31–36.

[117] Johnston, I. A., *Simulation of Flow Around Hypersonic Blunt-Nosed Vehicles for the Calibration of Air Data Systems*, Ph.D. Thesis, Department of Mechanical Engineering, The University of Queensland, 1999.

[118] Jones, R. A. and Huber, P. W., "Toward scramjet aircraft," *Astronautics and Aeronautics*, February 1978, pp. 38–48.

[119] Kandebo, S. W., "Franco-Russian tests may spur new projects," *Aviation Week and Space Technology*, December 14/21 1992.

[120] Kandebo, S. W., "Russian space initiatives," *Aviation Week and Space Technology*, March 30 1992, pp. 18–20.

[121] Kerrebrock, J. K., *Aircraft Engines and Gas Turbines*, The MIT Press, London, England, 1992.

[122] King, E. G., Freeman, L. M., Whitaker, K. W., and Karr, C. L., "Two-dimensional thrust vectoring nozzle optimization techniques," AIAA Paper 91-0473, 29th Aerospace Sciences Meeting, 1991.

[123] Kortmann, D. P., "PD-like fuzzy controllers compared with conventional PD controllers," *Transactions of the Institute of Measurement and Control*, Vol. 18, No. 3, 1996, pp. 135–148.

[124] Kosko, B., *Neural Networks and Fuzzy Systems*, Prentice Hall, Englewood Cliffs, NJ, 1992.

[125] Kosko, B., *Fuzzy Thinking*, Flamingo, 1994.

[126] Kosko, B. and Isaka, S., "Fuzzy logic," *Scientific American*, July 1993, pp. 62–67.

[127] Kreisselmeier, G., "Perspectives on the application of adaptive control to aircraft systems," *Applications of Adaptive Control*, edited by K. S. Narendra and R. V. Monopoli, chap. Aircraft Systems, Academic Press, 1979, pp. 269–290.

[128] Krishnajumar, K. and Goldberg, D. E., "Control system optimization using genetic algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 3, May-June 1992, pp. 735–740.

[129] Kubota, N. and Fukuda, T., "Genetic algorithms with age structure," *Soft Computing*, 1997, pp. 155–166.

[130] Kubota, N., Shimojima, K., and Fukuda, T., "The role of virus infection in a virus-evolutionary genetic algorithm," *Applied Mathematics and Computer Science*, Vol. 6, No. 3, 1996, pp. 415–429.

[131] Lee, C. C., "Fuzzy logic in control systems: Fuzzy logic controller - Part I," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 2, March/April 1990, pp. 404–418.

[132] Lee, C. C., "Fuzzy logic in control systems: Fuzzy logic controller - Part II," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 2, March/April 1990, pp. 419–435.

[133] Lewis, M. J., "Designing hypersonic inlets for bow shock location control," *Journal of Propulsion and Power*, Vol. 9, No. 2, Mar-Apr 1993, pp. 313–321.

[134] Liepmann, H. W. and Roshlo, A., *Elements of Gas Dynamics*, John Wiley and Sons, New York, 1957.

[135] Lin, C.-L. and Su, H.-W., "Intelligent control theory in guidance and control system design: An overview," *Proceedings of the National Science Counceil, Republic of China*, Vol. 24, No. 1, 200, pp. 15–30.

[136] Linkens, D. A. and Nyongesa, H. O., "Genetic algorithms for fuzzy control, Part 1: Offline system development and application," *IEE Proceedings - Control Theory and Applications*, Vol. 142, No. 3, May 1995, pp. 161–176.

[137] Maciejowski, J. M., *Multivariable Feedback Design*, Addison-Wesley, Wokingham, England, 1989.

[138] Mahfoud, S. W., "Boltzmann selection," *Evolutionary Computation 1 : Basic Algorithms and Operators*, edited by T. Bäck, D. B. Fogel, and Z. Michalewicz, chap. 26, Institute of Physics Publishing, 2000, pp. 195–200.

[139] Malki, H. A., Li, H., and Chen, G., "New design and stability analysis of fuzzy proportional-derivative control systems," *IEEE Transactions on Fuzzy Systems*, Vol. 2, No. 4, 1994, pp. 245–254.

[140] Mamdani, E. H., "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Transactions on Computers*, Vol. C-26, 1977, pp. 1182–1191.

[141] Mamdani, E. H., "Twenty years of fuzzy control: experiences gained and lessons learnt," *Second IEEE Conference on Fuzzy Systems*, 1993, pp. 339–344.

[142] Mamdani, E. H. and Assilian, S., "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, Vol. 7, 1975, pp. 1–13.

[143] Marcos, M. and Artaza, F., "A methodology for the design of rule-based supervisors," *European Journal of Control*, Vol. 2, 1996, pp. 305–312.

[144] Marrison, C. I. and Stengel, R. F., "Robust control system design using random search and genetic algorithm," *IEEE Transactions on Automatic Control*, Vol. 42, No. 6, June 1997, pp. 835–839.

[145] Marrison, C. I. and Stengel, R. F., "Design of robust control systems for a hypersonic aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 58–63.

[146] Matarrese, M. D., Messiter, A. F., and Adamson, T. C., "Control of hypersonic aerodynamic forces with surface blowing," *AIAA Journal*, Vol. 29, No. 12, December 1991, pp. 2094–2104.

[147] Maughmer, M., Ozoroski, L., Straussfogel, D., and Long, L., "Validation of engineering methods for predicting hypersonic vehicle control forces and moments," *A.I.A.A Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 4, 1993, pp. 762–769.

[148] McLean, D., *Automatic Flight Control Systems*, Prentice Hall, 1990.

[149] McLean, D. and Zaludin, Z. A., "Stabilization of longitudinal motion of a hypersonic transport aircraft," *Transactions of the Institute of Measurement and Control*, Vol. 21, No. 2/3, 1999, pp. 99–105.

[150] McMinn, J. D., "Extension of a Kolmogorov atmospheric turbulence model for time-based simulation implementation," AIAA Paper 97-3532, 1997, AIAA Guidance, Navigation and Control Conference.

[151] Mehta, U., "Hypersonic propulsion flight tests as essential to air-breathing aerospace plane development," *AIAA Sixth International Aerospace Planes and Hhypersonics Technolgies Conference*, 1995, AIAA Paper 95-6156.

[152] Mendel, G., "Experiments in plant hybridisation," 1965, Reprint of Mendel's original 1865 paper in English translation.

[153] Mendenhall, M., Lesieutre, D., Caruso, S., Marnix, F., and G., K., "Aerodynamic design of Pegasus: Concept to flight with computational fluid dynamics," *Journal of Spacecraft and Rockets*, Vol. 31, No. 6, November-December 1994, pp. 1007–1015.

[154] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 3rd ed., 1996.

[155] Michalewicz, Z. and Janikow, C. Z., "Genetic algorithms for numerical optimization," *Statistics and Computing*, Vol. 1, 1991, pp. 75–91.

[156] Michalewicz, Z., Logan, T. D., and Swaminathan, S., "Evolutionary operators for continuous convex parameter spaces," *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, 1994, pp. 84–97.

[157] Miyazawa, Y. and Motoda, T., "Stochastic parameter tuning applied to space vehicle flight control design," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 3, May–June 2001, pp. 597–604.

[158] Mooij, E., "Numerical investigation of model reference adaptive control for hypersonic aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 315–323.

[159] Moore, M. and Williams, J., "Aerodynamic prediction rationale for analyses of hypersonic configurations," AIAA Paper 89-0525, 27th Aerospace Sciences Meeting, 1989.

[160] Morgan, R. G. and Wallace, G. A., "Ethane combustion in a hypersonic duct," *Shock Tunnel Studies of Scramjet Phenomena*, Vol. 1 of *NASA Grant NAGW 674*, 1988.

[161] Nelder, J. A. and Mead, R., "A simplex method for function minimization," *The Computer Journal*, Vol. 7, 1965, pp. 308–313.

[162] Neubauer, A., "Adaptive non-uniform mutation for genetic algorithms," *Lecture Notes in Computer Science: Compuational Intelligence, Theory and Applications*, edited by B. Reusch, Vol. 1226, Springer, 1997, pp. 24–34.

[163] Nichols, R. A., Reichert, R. T., and Rugh, W. J., "Gain scheduling for H_∞ controllers: A flight control example," *IEEE Transactions on Control Systems Technology*, Vol. 1, No. 2, June 1993, pp. 69–79.

[164] Northam, G. B. and Anderson, G. Y., "Supersonic combustion ramjet research at Langley," AIAA Paper 86-0159, January 1986.

[165] Olsson, D. M. and Nelson, L. S., "The Nelder-Mead simplex procedure for function minimization," *Technometrics*, Vol. 17, No. 1, February 1975, pp. 45–51.

[166] O'Neill, R., "Algorithm as 47. function minimization using a simplex procedure," *Applied Statistics*, Vol. 20, 1971, pp. 338–345.

[167] Paredis, J., "Coevolutionary computation," *Artificial Life*, Vol. 2, No. 4, 1995, pp. 355–375.

[168] Parker, L. R., "Comparison of simplex algorithms," *Analytica Chimica Acta*, Vol. 175, 1985, pp. 231–237.

[169] Patton, R. J. and Liu, G. P., "Robust control design via Eigenstructure assignment, genetic algorithms, and gradient-based optimisation," *IEE Proceedings - Control Theory and Applications*, Vol. 141, No. 3, 1994, pp. 202–208.

[170] Paull, A., "Hyshot flight program," *RAMAC, International Ram Accelerator Workshop, Tokyo, Japan*, 2000.

[171] Paull, A., Stalker, R. J., and Mee, D. J., "Scramjet thrust measurement in a shock tunnel," *Aeronautical Journal*, April 1995, pp. 161–163.

[172] Pierce, J. R., *An Introduction to Information Theory : Symbols, Signals, and Noise*, Dover Publications, New York, 2nd ed., 1980.

[173] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipes in C : The Art of Scientific Computing*, Cambridge University Press, Cambridge, England, 1988.

[174] Pruett, C. D., Wolf, H., Heck, M. L., and Siemers, P. M., "An innovative air data system for the space shuttle orbiter: Data analysis techniques," AIAA Paper 81-2455, Nov. 1981.

[175] Rabiner, L. R. and Gold, B., *Theory and Application of Digital Processing*, Prentice Hall, Englewood Cliffs, N. J., 1975.

[176] Raney, D. and Lallman, F., "Control concept for maneuvering in hypersonic flight," AIAA Paper 91-5055, 1991.

[177] Rausch, V. L., McClinton, C. R., and Crawford, J. L., "Hyper-X: flight validation of hypersonic airbreathing technology," *XIII ISABE (International Society for Airbreathing Engines)*, Chattanooga, Tennessee, 1997, ISABE 97-7024.

[178] Ray, L. R. and Stengel, R. F., "Application of stochastic robustness to aircraft control systems," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 6, 1991, pp. 1251–1259.

[179] Rechenberg, I., *Cybernetic Solution Path of an Experimental Problem*, Ministry of Aviation, Royal Aircraft Establishment (U.K.), 1965.

[180] Reichert, R. T., "Dynamic scheduling of modern-robust-control autopilot designs for missiles," *IEEE Control Systems Magazine*, Vol. 12, No. 5, 1992, pp. 35–42, Presented at 1991 Conference on Decision and Control.

[181] Renders, J. and Bersini, H., "Hybridizing genetic algorithms with hill-climbing methods for global optimization: Two possible ways," *The First IEEE Conference on Evolutionary Computation*, Vol. 1, 1994, pp. 312–317.

[182] Rosenbrock, H. H., "An automatic method for finding the greatest or least value of a function," *The Computer Journal*, Vol. 3, No. 3, 1960, pp. 175–184.

[183] Roudakov, A. S., Semenoc, V. L., and Hicks, J. W., "Recent flight test results of the joint CIAM-NASA Mach 6.5 scramjet flight program," NASA TP 98-206548, NASA, Dryden Flight Research Center, 1998.

[184] Roudakov, A. S., Semenov, V. L., Kopchenov, V. I., and Hicks, J. W., "Future flight test plans of an axisymmetric hydrogen-fueled scramjet engine on the hypersonic flying laboratory," *7th International Spaceplanes and Hypersonics Systems and Technology Conference*, AIAA, 1996, AIAA 96-4572.

[185] Routh, M. W., Schwartz, P. A., and Denton, M. B., "Performance of the super modified simplex," *Analytical Chemistry*, Vol. 49, 1977, pp. 1422–1428.

[186] Rycroft, M., editor, *The Cambridge Encyclopedia of Space*, Cambridge University Press, Cambridge, New York, 1990.

[187] Rynaski, E. G., "Adaptive control application to aircraft," *Applications of Adaptive Control*, edited by K. S. Narendra and R. V. Monopoli, chap. Aircraft Systems, Academic Press, 1979, pp. 245–267.

[188] Sachs, G., Schoder, W., and Heller, M., "Robust control concept for a hypersonic test vehicle," *AIAA Sixth International Aerospace Planes and Hypersonic Technologies Conference*, 1995, AIAA Paper 95-6061.

[189] Sait, S. M. and Youssef, H., *Iterative Computer Algorithms with Applications to Engineering : Solving Combinatorial Optimization Problems*, IEEE Computer Society, 1999.

[190] Salomon, R., "Increasing adaptivity through evolutionary strategies," *From Animals to Animats 4, Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, edited by P. Maes, M. J. Mataric, J.-A. Meyer, J. Pollack, and S. W. Wilson, MIT Press, 1996, pp. 411–420.

[191] Schmidt, D., "Integrated control of hypersonic vehicles," AIAA Paper 93-5091, 1993, AIAA/DGLR Fifth International Aerospace Planes and Hypersonics Technologies Conference.

[192] Schmidt, D. K., Mamich, H., and Chavez, F., "Dynamics and control of hypersonic vehicles - The integration challenge for the 1990's," *AIAA Third International Aerospace Planes Conference*, 1991, AIAA Paper 91-5057.

[193] Schwefel, H.-P., *Evolution and Optimum Seeking*, John Wiley & Sons, 1995.

[194] Semple, E. C., "Trajectory and performance of an expendable scramjet powered launcher," *Seventh National Space Engineering Symposium*, The Institution of Engineers, Australia, 1992, pp. 37–41.

[195] Shannon, C. E., "A mathematical theory of communication," *The Bell System Technical Journal*, Vol. 27, No. 7 and 10, July and October 1948, pp. 379–423 and 623–656.

[196] Shaughnessy, J. D., Zane-Pinckney, S., McMinn, J. D., Cruz, C. I., and Kelley, M. L., "Hypersonic vehicle simulation model: Winged-cone configuration," Tech. Rep. NASA TM 102610, November 1990.

[197] Siarry, P., Berthiau, G., Durbin, F., and Haussy, J., "Enhanced simulated annealing for globally minimizing functions of many continuous variables," *ACM Transactions on Mathematical Software*, Vol. 23, No. 2, June 1997, pp. 209–228.

[198] Smith, A. L., Johnston, I. A., and Austin, K. J., "Comparison of numerical and experimental drag measurement in hypervelocity flow," *The Aeronautical Journal of the Royal Aeronautical Society*, November 1996, pp. 385–388.

[199] Smith, B. A., "Elevon failure precedes loss of first X-43A," *Aviation Week & Space Technology*, Vol. 154, No. 24, June 2001, pp. 50–51.

[200] South, M. C., Wetherill, G. B., and Tham, M. T., "Hitch-hiker's guide to genetic algorithms," *Journal of Applied Statistics*, Vol. 20, No. 1, 1993, pp. 153–175.

[201] Spendley, W., Hext, G. R., and Himsworth, F. R., "Sequential application of simplex designs in optimisation and evolutionary operation," *Technometrics*, Vol. 4, 1962, pp. 441–461, cited in [161].

[202] Staff, N. F. R. C., "Experience with the X-15 adaptive flight control system," Tech. Rep. NASA TN D-6208, NASA, 1971.

[203] Stalker, R. J., "Thermodynamics and wave processes in high Mach number propulsive ducts," AIAA Paper 89-0261, January 1989.

[204] Stalker, R. J., "Scaling laws and the launch vehicle market," *Seventh National Space Engineering Symposium 1992*, BARTON, A.C.T., Institution of Engineers, Australia, 1992.

[205] Stanhope, S. A. and Daida, J. M., "An individually variable mutation-rate strategy for genetic algorithms," *Evolutionary Programming VI : Proceedings of the Sixth Annual Conference on Evolutionary Programming*, Vol. 1213, Springer-Verlag, Indianapolis, Indiana, USA, 1997, pp. Angeline, P. J. and Reynolds, R. G. and McDonnell, J. R. and Eberhart, R.

[206] Steinberg, M., "A fuzzy logic based F/A-18 automatic carrier landing system," AIAA-92-4392-CP, 1992.

[207] Steinberg, M., "Development and simulation of an F/A-18 fuzzy logic automatic carrier landing system," *Second IEEE International Conference on Fuzzy Systems*, Vol. 2, 1993, pp. 797–802.

[208] Steinberg, M., "An initial assessment of neural network and fuzzy logic technology for flight control systems," *Proceedings of the American Control Conference*, June 1994, pp. 173–177.

[209] Stengel, R. F., *Optimal Control and Estimation*, Dover Publications, New York, 1994.

[210] Stengel, R. F. and Marrison, C. I., "Robustness of solutions to a benchmark control problem," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 5, September 1992, pp. 1060–1067.

[211] Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*, Wiley, New York, 1992.

[212] Stewart, B., Morgan, R. G., and Jacobs, P. A., "The RHYFL facility as a high performance expansion tube for scramjet testing," AIAA Paper 2000-2595, 21st AIAA Aerodynamic Measurement Technology and Ground Testing Conference, 2000.

[213] Sutton, G. P., *Rocket Propulsion Elements : An Introduction to the Engineering of Rockets*, John Wiley and Sons, Inc, 6th ed., 1992.

[214] Swithenbank, J., "Hypersonic air-breathing propulsion," *Progress in Aeronautical Sciences*, Vol. 8, 1966, pp. 229–294.

[215] Takagi, T. and Sugeno, M., "Derivation of fuzzy control rules from human operator's control actions," *Proceedings of the IFAC Symposium of Fuzzy Information, Knowledge Representation and Decision Analysis*, 1983, pp. 55–60, cited in Lee [131].

[216] Takahashi, O., Kita, H., and Kobayashi, S., "A real-coded genetic algorithm using distance dependent alternation model for complex function optimization," *Proceedings of the Genetic and Evolutionary Computation Conference*, 2000, pp. 219–226.

[217] Tauber, M. E. and Meneses, G. P., "Aerothermodynamics of transatmospheric vehicles," AIAA 86-1257, 1986.

[218] Tomayko, J. E., *Computers Take Flight: A History of NASA's Pioneering Digital Fly-by-Wire Project*, NASA, 2000.

[219] Tsujikawa, Y., Tsukamoto, Y., and Fujii, S., "Performance analysis of scramjet engine with quasi-one-dimensional flow model," *International Journal of Hydrogen Energy*, Vol. 16, No. 2, 1991, pp. 135–142.

[220] Vincent, J. H., Emami-Naeini, A., and Khraishi, N. M., "Case study comparison of linear quadratic regulator and h_∞ control synthesis," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 5, Sept.–Oct. 1994, pp. 985–965.

[221] Voland, R. T., Auslender, A. H., Smart, M. K., Roudakov, A. S., Semenov, V. L., and Kopchenov, V., "CIAM/NASA Mach 6.5 scramjet flight and ground test," AIAA Paper 99-4848, 1999.

[222] Voland, R. T., Rock, K. E., Huebner, L. D., Witte, D. W., Fischer, K. E., and McClinton, C. R., "Hyper-X engine design and ground test program," *AIAA 8th International Space Planes and Hypersonic Systems and Technologies Conference*, 1992, AIAA Paper 98-1532.

[223] Wadawadigi, G., Tannehill, J. C., and Lawrence, S. L., "Three-dimensional computation of the integrated aerodynamic and propulsive flowfields of a generic hypersonic space plane," AIAA Paper 94-0633, 1994.

[224] Wang, C.-H., Hong, T.-P., and Tseng, S.-S., "Integrating fuzzy knowledge by genetic algorithms," *IEEE Transactions on Evolutionary Computation*, Vol. 2, No. 4, November 1998, pp. 138–149.

[225] Wang, Q. and Stengel, R. F., "Robust nonlinear control of a hypersonic aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 4, 2000, pp. 577–585.

[226] Weber, R. J. and Mackay, J., "An analysis of ramjet engines using supersonic combustion," NACA TN 4386, 1958.

[227] Weidner, J. P., "Hypersonic propulsion – breaking the thermal barrier," *Proceedings of the Institute of Mechanical Engineers. Part G, Journal of Aerospace Engineering*, Vol. 207, No. G1, 1993, pp. 47–59.

[228] Weilmuenster, K. J., Gnoffo, P. A., and Greene, F. A., "Navier-Stokes simulations of Orbiter aerodynamic characteristics including pitch trim and bodyflap," *Journal Spacecraft and Rockets*, Vol. 31, No. 3, 1994, pp. 355–366.

[229] Wendt, M. N. and Jacobs, P. A., "Modular composite scramjet motor (C.S.M) testing," Report 2/96, Department of Mechanical Engineering, The University of Queensland, February 1996.

[230] Wendt, M. N., Jacobs, P. A., and Stalker, R. J., "Design and testing of a modular scramjet vehicle," *20th International Symposium on Space Technology and Science*, May 1996, Paper 96-a-2-12.

[231] Whitmore, S. A., Davis, R. J., and Fife, J. M., "In-Flight Demonstration of a Real-Time Flush Airdata Sensing (RT-FADS) System," Tech. Rep. NASA TM-104314, NASA Dryden Flight Research Center, 1995.

[232] Whitmore, S. A. and Moes, T. R., "Measurement uncertainty and feasibility study of a flush airdata system for a hypersonic flight experiment," Tech. Rep. NASA TM-4627, Dryden Flight Research Center, 1994.

[233] Wie, B., *Space Vehicle Dynamics and Control*, AIAA Education Series, 1998.

[234] Wilhite, A. W., Bush, L. B., Cruz, C. I., Lepsch, R. A., Morris, W. D., Stanley, D. O., and Wurster, K. E., "Advanced technologies for rocket single-stage-to-orbit vehicles," *Journal of Spacecraft and Rockets*, Vol. 28, No. 6, Nov-Dec 1991, pp. 646–651.

[235] Wolpert, D. H. and Macready, W. G., "No free lunch theorem for optimization," *IEEE Transcactions on Evolutionary Computation*, Vol. 1, No. 1, April 1997, pp. 67–82.

[236] Yao, X., Liu, Y., and Lin, G., "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 2, 1999, pp. 82–102.

[237] Yasunobu, S. and Miyamoto, S., *Industrial Application of Fuzzy Control*, chap. Automatic Train Operation by Predictive Fuzzy Control, 1995, pp. 1–18.

[238] Yen, J., Liao, J. C., Lee, B., and Randolph, D., "A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol. 28, No. 2, April 1998, pp. 173–191.

[239] Ying, H., "Practical design of nonlinear fuzzy controllers with stability analysis for regulating processes with unknown mathematical models," *Automatica*, Vol. 30, No. 7, 1994, pp. 1185–1195.

[240] Zadeh, L., "Fuzzy sets," *Information and Control*, Vol. 8, 1965, pp. 338–353.

[241] Zadeh, L., "A rationale for fuzzy control," *Transactions ASME Journal Dynamics, Systems, Measurement and Control*, Vol. 94, 1972, pp. 3–4.

[242] Zadeh, L., "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-3, No. 1, January 1973, pp. 28–44.

[243] Zeutzius, M., Beylich, A. E., Matsuo, S., and Setoguchi, T., "Experimental investigation of a gas dynamic thrust vector control for hypersonic space planes," *JSME International Journal. Series B, Fluids and thermal engineering*, Vol. 39, No. 1, 1996, pp. 101–111.

[244] Zhou, Z., Lin, C.-F., and Burken, J., "Fuzzy logic based flight control system for hypersonic transporter," *Proceedings of the 36th Conference on Decision and Control*, December 1997, pp. 2730–2735.

[245] Zitzler, E. and Thiele, L., "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Transactions of Evolutionary Computation*, Vol. 3, No. 4, Nov. 1999, pp. 257–271.