University of Southern Queensland
Faculty of Engineering and Surveying

Hypersonic rocket manoeuvre in the TUSQ facility

A dissertation submitted by

Michael John Fowler

In fulfilment of the requirements of

Bachelor of Engineering (Mechanical)

Submitted 30 October 2014
Abstract

A number of methods are currently used for the design and testing of hypersonic vehicles, of particular focus in this research is those techniques employed in hypersonic wind tunnels. Current approaches to this testing use fixed sting mounted models, tethered models, and free-flight models to study its behaviour. An addition to these testing techniques and the focus of this project is the use of models with actuated control surfaces to allow the study of a hypersonic vehicle under dynamic conditions. This project aimed to design, construct, and validate; through demonstration of a pitching manoeuvre, a sub-scale model with an on-board control surface actuation system suitable for use in the University of Southern Queensland hypersonic wind tunnel (TUSQ). A tethered model with actuated control surfaces would indicate how the full-scale vehicle would behave whilst undertaking a manoeuvre.

The first phase of design was a development of a semi-analytical analysis to determine the expected forces and therefore response of the model. This provides data to later compare with the experimental results and parameters for the design of the model. The design of the model covered all components including the fin actuation system, tethering and support system and model housing design. The final phase was building and testing of the model in the TUSQ facility.

Two runs in the hypersonic facility were completed as part of the research. Unfortunately neither run resulted in a demonstration of an entire successful manoeuvre. Analysis of the results revealed that the motor controlling the fin operations was providing insufficient torque and the fin control was not occurring as expected. In addition the model exited the Mach cone of developed flow during its pitching manoeuvre. These two factors caused discrepancies between theoretical calculations and experimental data. Analysis of the results and high-speed footage of the model indicate that the technique has the potential to be valid however it will require some further work to make it practical and effective for use in design of hypersonic vehicles.
Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled “Research Project” is to contribute to the overall education within the student’s chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.
Certification

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Michael John Fowler

0061019338
Acknowledgements

I would like to thank my supervisor Professor David Buttsworth for his time in answering my many questions and guidance throughout the completion of my research. I would also like to thank my father Kevin Fowler for his assistance in the assembly of the model and associated release system, as well as some initial bench testing of these designs.
## Table Of Contents

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Certification</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>v</td>
</tr>
<tr>
<td>Table Of Contents</td>
<td>vi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>x</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xii</td>
</tr>
<tr>
<td>Nomenclature and Acronyms</td>
<td>xiii</td>
</tr>
<tr>
<td>Chapter 1 – Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Chapter Outline</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Purpose</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Objectives</td>
<td>2</td>
</tr>
<tr>
<td>1.5 Project Completion Plan</td>
<td>3</td>
</tr>
<tr>
<td>Chapter 2 – Literature Review</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Chapter Outline</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Hypersonic Flow</td>
<td>5</td>
</tr>
<tr>
<td>2.2.1 Thin Shock Layers</td>
<td>6</td>
</tr>
<tr>
<td>2.2.2 Entropy Layer</td>
<td>6</td>
</tr>
<tr>
<td>2.2.3 Viscous Interaction</td>
<td>6</td>
</tr>
<tr>
<td>2.2.4 High-Temperature Flows</td>
<td>6</td>
</tr>
<tr>
<td>2.2.5 Low-Density Flow</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Hypersonic Vehicle Testing</td>
<td>7</td>
</tr>
<tr>
<td>2.3.1 Computational Fluid Dynamics Analysis</td>
<td>7</td>
</tr>
<tr>
<td>2.3.2 Hypersonic Shock Tunnel</td>
<td>7</td>
</tr>
<tr>
<td>2.3.3 Sub-Scale Flight Testing</td>
<td>7</td>
</tr>
<tr>
<td>2.4 Models with Actuated Control Surfaces</td>
<td>8</td>
</tr>
<tr>
<td>2.4.1 Control of Model</td>
<td>8</td>
</tr>
<tr>
<td>2.4.2 Dynamic Scaling</td>
<td>9</td>
</tr>
<tr>
<td>2.5 Newtonian Flow Theory</td>
<td>10</td>
</tr>
<tr>
<td>2.5.1 Justification for Use of Newtonian Flow Theory</td>
<td>10</td>
</tr>
</tbody>
</table>
2.5.2 Newtonian Flow Theory Methods 13
2.5.3 Forces & Moments Location 16

Chapter 3 – Theoretical Rocket Performance 18
3.1 Chapter Overview 18
3.2 Body & Tail Fin Shape 18
  3.2.1 Body Shape 18
  3.2.2 Tail Fin Shape 19
3.3 Procedure for Analysis of Response to Flow 20
  3.3.1 Model Force Nomenclature 20
  3.3.2 Pitch Determination 21
  3.3.3 Angular Acceleration Determination 22
3.4 Theoretical Values at Different AOA 23
  3.4.1 Processing of Graphed Coefficient Data 23
  3.4.2 Determining Nozzle Exit Conditions 25
  3.4.3 Calculating Wing Coefficient Data 25
  3.4.4 Matlab Script for Tail Fin Actuation Required 26
  3.4.5 Matlab Script for Tail Fin Actuation Rate 27
  3.4.6 Matlab Script for Angular Acceleration & Manoeuvre Time 27
  3.4.7 Results of Theoretical Analysis 28

Chapter 4 – Methodology 33
4.1 Chapter Overview 33
4.2 Model Design Parameters 33
  4.2.1 Model Scale 33
  4.2.2 Test Section Mounting Points 34
4.3 Initial Design Concepts 34
  4.3.1 Tail Fin Actuation System Concepts 35
  4.3.2 Model Pitching Support Concepts 36
4.4 Design Choice & Development 37
  4.4.1 Tail Fin Actuation System 37
  4.4.2 Model Housing 43
  4.4.3 Model Pitching Support 44
4.5 Experimental Equipment 46
  4.5.1 TUSQ Facility 46
Reference List 71
Appendix A – Project Specification 74
Appendix B – Coefficient Data & Model Geometry 75
Appendix C – Matlab Scripts 79
Appendix D – Results of Theoretical Analysis 96
Appendix E – Risk Assessment 100
Appendix F – Actuation System Development 103
Appendix G – Model Drawings 124
Appendix H – Raw Experimental Data 135
List of Figures

Figure 1.1: Project task completion timeline .......................................................... 4
Figure 2.1: Free-flight testing in a full scale wind tunnel .............................................. 9
Figure 2.2: Viscous and inviscid flow regions over a body .......................................... 11
Figure 2.3: Flow separation at high angle of attack ..................................................... 11
Figure 2.4: Supersonic shock wave ........................................................................... 12
Figure 2.5: Hypersonic shock wave .......................................................................... 12
Figure 2.6: Newtonian flow theory for a flat surface ................................................... 14
Figure 2.7: Separation of normal force into lift & drag components .......................... 14
Figure 3.1: Basic body shape of model ....................................................................... 19
Figure 3.2: Simple wing profiles Investigated ............................................................. 19
Figure 3.3: Diagram of model force nomenclature ...................................................... 20
Figure 3.4: Wing nomenclature for lift & drag coefficient calculation ....................... 26
Figure 3.5: Required tail fin actuation to achieve a desired AoA ............................... 30
Figure 3.6: Model AoA & tail fin positions for a 10 degree manoeuvre .................. 31
Figure 3.7: Model angular acceleration for a 10 degree manoeuvre .......................... 31
Figure 3.8: Model body & wing loads for a 10 degree manoeuvre ............................ 32
Figure 4.1: Mach cone for Mach 5.8 flow in TUSQ facility ........................................ 34
Figure 4.2: Outlet nozzle mounting holes & mounting plate ....................................... 34
Figure 4.3: Typical linear solenoid ............................................................................. 35
Figure 4.4: Typical stepper motor ............................................................................. 36
Figure 4.5: Yoke type pitching support concept ......................................................... 37
Figure 4.6: Stepper motors purchased ...................................................................... 38
Figure 4.7: Determining approximate torque of a 15mm stepper motor .................. 39
Figure 4.8: Stepper motor with attached micro-metal gearbox & bevel gear.......... 40
Figure 4.9: Pololu A-Star 32U4 Micro ....................................................................... 41
Figure 4.10: Pololu DRV8834 Stepper Driver ............................................................. 41
Figure 4.11: D2F-L Omron Snap Action Switch ......................................................... 42
Figure 4.12: Stepper motor control circuit implemented ........................................... 42
Figure 4.13: Model cross-section showing componentry layout ................................ 43
Figure 4.14: 3D printed model housing with components assembled ..................... 43
Figure 4.15: Support stand features ........................................................................... 45
Figure 4.16: Schematic of the TUSQ Ludweig tube facility ................................47
Figure 4.17: TUSQ Mach 6 nozzle sketch ....................................................47
Figure 4.18: Body halves separated with connection leads showing ..........49
Figure 4.19: Two tethers in ‘X’ configuration ............................................49
Figure 4.20: Support system .....................................................................50
Figure 4.21: Sample image demonstrating determination of model angles ......51
Figure 5.1: Image sequence from RUN283 ................................................55
Figure 5.2: Expected & observed body & fin angles for RUN283..............56
Figure 5.3: Expected AoA using observed tail fin position of RUN283 .......57
Figure 5.4: Expected angular acceleration for RUN283 .............................57
Figure 5.5: Image sequence from RUN286 part 1 .....................................60
Figure 5.6: Image sequence from RUN286 part 2 .....................................61
Figure 5.7: Expected & observed body & fin angles for RUN286..............62
Figure 5.8: Expected AoA using observed tail fin position of RUN286 .......63
Figure 5.9: Expected & observed angular acceleration for RUN286 .........63
Figure B.1: Geometric details of considered hypersonic bodies ...............75
Figure B.2: Coefficient data for basic body, model I .................................76
Figure B.3: Fitted data comparison for body normal force coefficients .......77
Figure B.4: Fitted data comparison for body axial force coefficients .........77
Figure B.5: Fitted data comparison for body centre of pressure ...............78
Figure B.6: Lift & drag coefficients for an 8.6 degree wedge wing ..........78
Figure D.1: Body & fin angles for a 5 degree AoA manoeuvre .................96
Figure D.2: Angular acceleration for a 5 degree AoA manoeuvre .............97
Figure D.3: Body & control surface forces for a 5 degree AoA manoeuvre ....97
Figure D.4: Body & fin angles for a 15 degree AoA manoeuvre ...............98
Figure D.5: Angular acceleration for a 15 degree AoA manoeuvre ..........98
Figure D.6: Body & control surface forces for a 15 degree AoA manoeuvre ..99
Figure F.1: Stage 1 setup ..........................................................................103
Figure F.2: Stage 2 electrical schematic .....................................................106
Figure F.3: Stage 2 setup ..........................................................................107
Figure F.4: Stage 3 electrical schematic .....................................................111
Figure F.5: Stage 3 setup ..........................................................................112
List of Tables

Table 3.1: Selected model geometrical parameters ........................................ 28
Table 3.2: Determined model mass parameters ............................................. 28
Table 3.3: Actuation stepper motor parameters ............................................. 28
Table 3.4: Typical TUSQ flow conditions .................................................... 29
Table 3.5: Summary of results for 5, 10, and 15 degree AoA scenarios ........... 32
Table 5.1: Summary of all test run parameters .............................................. 53
Table E.1: Risk assessment matrix .............................................................. 101
Table E.2: Risk assessment results .............................................................. 102
Table G.1: Drawing numbers and descriptions ............................................ 124
Table G.2: Drawing revisions used for each run ......................................... 124
# Nomenclature and Acronyms

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>Vehicle angle of attack (AoA), nose up +ve</td>
<td>rad</td>
</tr>
<tr>
<td>$\dot{\alpha}$</td>
<td>Vehicle angular velocity</td>
<td>rad/s</td>
</tr>
<tr>
<td>$\ddot{\alpha}$</td>
<td>Vehicle angular acceleration</td>
<td>rad/s²</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Fin actuation relative to vehicle body, nose down +ve</td>
<td>rad</td>
</tr>
<tr>
<td>$\dot{\beta}$</td>
<td>Tail fin actuation rate</td>
<td>rad/s</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Mach cone angle</td>
<td>rad</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Wing AoA to flow</td>
<td>rad</td>
</tr>
<tr>
<td>$\psi$</td>
<td>Wing wedge angle</td>
<td>degrees</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Density</td>
<td>kg/m²</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Stress</td>
<td>Pa</td>
</tr>
<tr>
<td>$a$</td>
<td>Linear acceleration</td>
<td>m/s²</td>
</tr>
<tr>
<td>$A_p$</td>
<td>Total plan form area of control surfaces</td>
<td>m²</td>
</tr>
<tr>
<td>$A_s$</td>
<td>Cross-sectional area cylindrical body</td>
<td>m²</td>
</tr>
<tr>
<td>$C$</td>
<td>Force coefficient</td>
<td>-</td>
</tr>
<tr>
<td>$d_s$</td>
<td>Model body diameter</td>
<td>m</td>
</tr>
<tr>
<td>$F$</td>
<td>Force component</td>
<td>N</td>
</tr>
<tr>
<td>$g$</td>
<td>Gravitational acceleration constant</td>
<td>m/s²</td>
</tr>
<tr>
<td>$I$</td>
<td>Mass moment of inertia</td>
<td>kg.m²</td>
</tr>
<tr>
<td>$k$</td>
<td>Ratio of constant specific heats</td>
<td>-</td>
</tr>
<tr>
<td>$L$</td>
<td>Model overall length</td>
<td>m</td>
</tr>
<tr>
<td>$m$</td>
<td>Mass</td>
<td>kg</td>
</tr>
<tr>
<td>$M$</td>
<td>Moment about a point</td>
<td>N.m</td>
</tr>
<tr>
<td>$M_\infty$</td>
<td>Free stream Mach number</td>
<td>-</td>
</tr>
<tr>
<td>$p$</td>
<td>Pressure</td>
<td>Pa</td>
</tr>
<tr>
<td>$q_\infty$</td>
<td>Dynamic free stream pressure</td>
<td>Pa</td>
</tr>
<tr>
<td>$R$</td>
<td>Ideal gas constant</td>
<td>J/kg.K</td>
</tr>
<tr>
<td>$t$</td>
<td>Time</td>
<td>s</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature</td>
<td>K</td>
</tr>
</tbody>
</table>
$U$   Velocity \hspace{1cm} m/s
$W$   Weight \hspace{1cm} N
$x$   Location measured from model nose \hspace{1cm} m

**Subscripts**

0  Stagnation condition
$\infty$ Free stream condition
$A$ Axial force component
$\text{body}$ Main body of the vehicle
$c.\ g.$ Centre of gravity
$c.\ p.$ Centre of pressure
$D$ Drag force component
$L$ Lift force component
$N$ Normal force component
$pivot$ Pivot point
$\text{wing}$ Wing of the vehicle

**Acronyms**

3D  3-Dimensional
AoA  Angle of Attack
fps  Frames per second
ODE  Ordinary Differential Equation
TUSQ USQ Hypersonic Wind Tunnel
USQ  University of Southern Queensland
UQ   University of Queensland
Chapter 1 – Introduction

1.1 Chapter Outline
This chapter introduces the proceeding content of the report by providing a brief overview of the importance of hypersonic research. It then covers the purpose of the research contained in this report as well as defining the project objectives.

1.2 Background
Sustained hypersonic flight is an engineering feat that is yet to be completely accomplished. There is however a great deal of interest and research in being able to achieve these great speeds, in excess of about Mach 5 (Pritchard, 2011). Such speeds would allow travel times of about 90 minutes from London to Tokyo (Anderson, 2012). It is therefore becoming the next foreseeable step in manned flight.

Development into hypersonic flight has been triggered by the creation of progressively faster and faster aircraft. The first hurdle to overcome was breaking the sound barrier by travelling faster than Mach 1. Unfortunately as aircraft managed to obtain speeds of Mach 1 they would experience ‘high drag, buffeting, changes in structural loads, and even loss of control and in-flight breakups’ these affects appeared to indicate a barrier against faster than sound flight (Hallion, 2012).

After about a decade aircraft managed to break into the supersonic speeds, achieved when the aircraft is travelling nominally between Mach 1 to 5. The first aircraft to break this speed of sound barrier was the Bell XS-1, with a speed of Mach 1.06 (Hallion, 2012). From here the F-104 fighter and the Concorde airliner are examples of Mach 2 capable vehicles, at Mach 3 upwards is the Lockheed SR-71 (Heppenheimer, 2007). At Mach 4 heating effects on the aircraft became severe and in addition no turbojet engine had been used at these speeds. This led to the development of the X-7 a ramjet vehicle, which achieved Mach 4.3. This was then follow by development of the X-15, which successfully reached the Mach 6 mark using a series of rocket engines (Heppenheimer, 2007).

One of the more recent advances in hypersonic flight is the HTV-2 tested in 2011. By use of rockets it reached speeds of Mach 20, however after 200 seconds at these speeds its skin peeled from the frame and the test was aborted (Anderson, 2012). Today there is still a lot of interest and research undertaken into sustainable hypersonic flight within our atmosphere. The University of Queensland (UQ) recently completed the design and build of Scramspace 1; a scramjet intended for operation at speeds of Mach 8 (Scramspace 1, 2013). The University of Southern Queensland (USQ) was also
involved in the testing of the project through use of their hypersonic wind tunnel designated TUSQ. This testing recently allowed Ennis (2013) to complete his project ScramSpace hypersonic aerodynamic drag coefficient \( (C_D) \) determined by free-flight experiments at TUSQ wind-tunnel, where he began development into a system for performing true free flight experiments in the hypersonic wind tunnel. The facility itself has also allowed much experimental research to be undertaken and due to the relatively long flow durations produced it makes it especially useful where unsteady vehicle aerodynamics are of interest.

1.3 Purpose
This project intends to validate the performance of manoeuvres in a hypersonic testing facility using a model with actuated control surfaces. The focus of the project will be on performing a pitching manoeuvre. It will utilise a tethered scaled model of a blunt-nose rocket, housing an on-board microcontroller package for control, and a high-speed camera to record the model change in attitude as a result of control surface actuation.

It seeks to determine the practicality and feasibility of installing an actuated control surface assembly into a small model for dynamic testing. The reliability and effectiveness of the tail-fin assembly will then be determined by comparison of high-speed camera data with semi-analytical calculations of the expected model response.

Successful research will be determined by; successful tail fin operation, attitude changes occurring as predicted, and a good correlation between experimental data and semi-analytical calculations. If this testing approach proves to be valid it would then present the opportunity for future work to experimentally determine the attitude change for more complex model geometries, which prove difficult to determine via other methods. In addition the research into performance of a simple pitching manoeuvre can also be applied to other manoeuvre types. Primarily if the approach proves feasible it can allow experimental determination of the behaviour of hypersonic vehicles by more accurately simulating the dynamics of different manoeuvres. This then builds on the current set of design techniques available for design of hypersonic vehicles.

1.4 Objectives
It is desired that the project will provide sufficient successful test data to allow future more complex models to be tested within the TUSQ facility and other hypersonic facilities. It will also provide an analysis of any limitations that the tunnel presents to testing of a simple manoeuvre and the effects they may have.

Primarily the objective of the project is to mount an actuated tail fin assembly into a rocket model of suitable scale for the TUSQ. Then it should demonstrate a successful manoeuvre within the steady run time of the tunnel flow. Secondly it is important to then compare these results with those determined theoretically
using a semi-analytical approach. If time and resources permit the model will also implement a nose mounted pressure transducer to detect model attitude. Also if practical multiple runs with different amounts of tail fin actuation and different rates of tail fin actuation will be undertaken to produce different sets of test data.

Project completion will require the following:

- Determination of expected model response to a tail fin actuation via a semi-analytical approach
- Design of a scaled rocket model suitable for the installation of a microcontroller package, nose mounted pressure transducer, and tail fin actuation assembly
- Design of a tail fin actuation assembly.
- Fabrication of previously designed model and tail fin assembly
- Testing of model in the TUSQ facility
- Comparison of semi-analytical data with tunnel data to draw conclusions.

1.5 Project Completion Plan

A project task completion timeline is presented in Figure 1.1 below. It has been split into 7 key tasks each provided with their expected start and end dates.

The initial phase of this project is the project scope, where the guidelines for the remainder of the project are laid. It defines the dissertation focus and also what outcomes are to be achieved. Occurring mid way through this process is the Literature Review, this will give the author the chance to look at useful research which will provide a starting point for the remainder of this project.

Towards the end of the Literature review it will be possible to begin theoretical calculations, which will allow determination of the model geometry and provide theoretical model attitude responses. Following the selection of model geometry the Methodology section will be started where manufacturing of the model will be covered. This section will also cover the experimental procedure used for testing.

Upon completion of the model design, manufacturing can begin. Directly following the manufacturing model testing can begin in the wind tunnel, this testing process will then be recorded in the methodology section of the report. Finally collected data from testing will be presented and analysed in the results and discussion section of the report.
Figure 1.1: Project task completion timeline
Chapter 2 – Literature Review

2.1 Chapter Outline
This chapter aims to research and compile current knowledge in the field of hypersonics with a focus on experimental testing of hypersonic vehicles in shock tunnels. It begins with a brief introduction to hypersonic flow, a discussion on the current techniques for testing & design of hypersonic vehicles, a look at dynamic models with movable control surfaces, and finally the applicability and feasibility of using Newtonian flow theory for calculations is examined.

2.2 Hypersonic Flow
Hypersonic flow is generally accepted as the point where the Mach number of a flow exceeds 5. However there is no instant transition as would occur when transitioning from subsonic speeds of less than Mach 1 to supersonic speeds greater than Mach 1. Instead it is manifested by an increase in energy flux and surface forces as a result of high temperature gases around the vehicle. It will usually result in strong shocks and either equilibrium or non-equilibrium gas chemistry (Bertin & Cummings, 2006). Special consideration for this high-speed flow is required to ensure the pressures and surface friction forces are properly considered in the determination of expected aerodynamic forces and moments. This in turn will allow the successful design of the vehicle ‘lift, drag, pitching moment, and control surface effectiveness’ (Bertin & Cummings, 2006). Research into hypersonic flow is undertaken to provide a bank of knowledge containing theory and any special considerations that need to be made in the design and testing of the model.

Mach 5 has been generally accepted as the point where a flow transitions from supersonic to hypersonic, however the Mach number itself has not yet been defined. It can be defined as the ratio of the vehicle speed (or fluid speed) to the speed of sound in that fluid, i.e. \( M = \frac{U}{c} \) where \( U \) (m/s) is the velocity of the vehicle and \( c \) (m/s) is the speed of sound in the fluid (Pritchard, 2011). The speed of sound \( c \) is a function of temperature and is defined by the following:

\[
c = \sqrt{kRT}
\] (2.1)

Where \( k \) is the ratio of constant specific heats, \( R \) (J/kg.K) is the gas constant, and \( T \) (K) is the temperature of the fluid (Pritchard, 2011). It can therefore be seen that the Mach number varies both with vehicle velocity and fluid temperature.

At these high speeds experienced in the hypersonic flows, and to a lesser extent in the supersonic flows cause intense aerodynamic heating to result. This means that a hypersonic vehicle design will differ from that of a subsonic or supersonic vehicle. Anderson (2006) notes that the Lockheed F-104 supersonic aircraft utilises a ‘sharp, needle-like nose and slender fuselage, very thin wings and tail surfaces with very sharp leading edges’ which successfully minimise
wave drag at supersonic speeds. This is in contrast to the X-20 Dynasoar hypersonic aircraft which utilised a ‘blunt, rounded leading edge, and a rather thick fuselage with a rounded nose’ (Anderson, 2006). Heppenheimer (2007) mentions in his research that many hypersonic designs utilise the blunt-nose design in order to minimise the amount of heat transfer into the vehicle and that there is an optimum nose bluntness which maximises nose cooling effects whilst minimising the production of the heat.

It has so far been discussed when hypersonic flow occurs and also why it is important to considerer separately, supersonic and subsonic flows, it is now important to identify the physical phenomena of theses flows. This understanding is critical to allow an accurate estimate of a selected design’s response and also to explain any results from the experimental analysis in this project. Anderson (2006) best summarises these different phenomena by discussing each in turn, they include; Thin Shock Layers, Entropy Layer, Viscous Interaction, High-Temperature Flows, and Low Density Flows.

2.2.1 Thin Shock Layers
As aircraft accelerate to the supersonic region they experience oblique shocks due to sudden compression and deflection of the flow (Pritchard, 2011). This oblique shock causes streamlines to rapidly change direction and usually follow the surface of the craft (Pritchard, 2011). As the Mach number of the aircraft increases the oblique shock becomes progressively more squashed onto the body. Towards the hypersonic Mach numbers this shock layer becomes quite thin and with a high Reynolds number Newtonian Flow Theory applies (Anderson, 2006).

2.2.2 Entropy Layer
Towards the tip of a blunt-nosed object in hypersonic flows there is a large curl to the shock layer. At this point there is a large change in entropy, this then flows down along the body for large distances. This can present problems in performing boundary-layer calculations (Anderson, 2006).

2.2.3 Viscous Interaction
As the hypersonic flow is slowed around the surface of the aircraft kinetic energy is transferred into the boundary layer. This results in a temperature increase and therefore an increase in viscosity and a decrease in density. To result in a decrease in density the thickness of the boundary layer must increase. As this boundary layer becomes sufficiently large it interacts with the surrounding inviscid flow, it can also become so thick that it will merge with the shock layer (Anderson, 2006).

2.2.4 High-Temperature Flows
As a result of the viscous interaction discussed previously high temperatures will result. In some cases these are sufficient to cause dissociation or ionization in the gas. Primarily this affect leads to issues with design of aircraft in order to
reduce or resist the convective and radiative heating effects from the intensely hot flow (Anderson, 2006).

2.2.5 Low-Density Flow
Low-density flow occurs at high altitudes where the spacing between gas molecules becomes large. Under these conditions alternative equations and theory must be applied ‘using concepts from kinetic theory’ (Anderson, 2006).

2.3 Hypersonic Vehicle Testing
Due to the largely unpredictable nature of hypersonic flows, especially in cases of complex or un-trialled geometries, it is important to undertake various simulations of the aircraft before making the financial commitment to build a prototype. Hicks (1993) covers in his work three main methods, these include; Computational Fluid Dynamics (CFD) Analysis, Hypersonic Shock Tunnel Testing, and Subscale Flight Testing. Chambers (2010) also places a strong recommendation towards the use of wind tunnel testing and drop testing of subscale models.

2.3.1 Computational Fluid Dynamics Analysis
The use of CFD in hypersonic aircraft design provides a manufacturing free method of testing a design. It can provide an initial indication as to how effective a design may be as well as any potential problems. Hicks (1993) states that the CFD software typically uses the Navier-Stokes fluid dynamics equations ‘which assume uniform, homogenous flow with no chemical reaction of species diffusion’. Although software does exist to allow consideration of some of these effects, the computation time is long, and has difficulties in simulating complex geometries or flows.

2.3.2 Hypersonic Shock Tunnel
Ground testing of hypersonic vehicles can contribute to their successful design and development. Typically these tests will make apparent any technical problems and allow them to be addressed and rectified before further investment in a design is made. Unfortunately these tests also have their limitations, especially in the hypersonic speeds. These can be due to ‘inaccuracies in scale or size, lack of incorporating or simulating critical systems, and lack of properly represented system integration’ (Hicks, 1993). In addition the facility itself can be limited by brief test durations and limited Mach numbers. Bertin and Cummings (2006) note that in many cases different tunnels need to be used, each designed for a different purpose, in order to obtain all the required data.

2.3.3 Sub-Scale Flight Testing
Typically this stage is begun after a CFD analysis and ground testing, and is the ultimate stage in proving a design (Hicks, 1993). Hicks (1993) provides two methods of achieving sub-scale flight testing. Subscale captive carry involves the model being permanently attached to an aircraft or rocket and accelerated to the desired Mach number and test data recorded. Alternatively air launched
free flight allows testing of larger scale models without interference of the transporting vehicle. Chambers (2010) also discusses similar methods for sub-scale flight-testing which tend to agree with those by Hicks (1993).

2.4 Models with Actuated Control Surfaces
Models with control surfaces can be used in various ways, these include; captive tests, single degree-of-freedom tests, free flight tests within a wind tunnel, and sub-scale flight-testing (Owens et al. 2006). Captive dynamic tests involve the use of a rigidly mounted model, which has either forced oscillation or a rotary balance test performed on it. In a single degree-of-freedom test the model is allowed to roll or pitch. Undertaking this particular test allows assessment of models stability, typically without any stability control from control surfaces. Free flight tests within a wind tunnel give the opportunity to see how the aircraft might respond when completely unconstrained. This method can provide data closely representing that of the prototype aircraft. Sub-scale flight-testing provides the best data of all tests, however it only occurs as a result of significant design and testing in previous methods, a summary of this test method has been discussed previously.

2.4.1 Control of Model
In performing a free-flight operation within a wind tunnel a great deal of coordination is required. Chambers (2010) and Owens et al. (2006) cover how these models are controlled when used in subsonic and supersonic wind tunnels. They are held with minimal restraint, the only connection is a flexible; partly steel cable that runs from the aircraft and attaches to roof of the tunnel. This cable provides air to the model for simulated thrust, power for operation of the control surfaces, and data cables for feedback of the models performance. This process requires multiple operators each controlling a small portion of the aircraft or tunnel, these include; a pilot for rolling and yaw control, a thrust pilot controlling compressed air, and a pitch pilot. Figure 2.1 shows the layout of such a free-flight test control centre. Splitting the piloting task between so many operators is required because the reduced scale of the model results in its dynamics being greatly accelerated and therefore requires rapid pilot response (Owens et al. 2006).
Figure 2.1: Free-flight testing in a full scale wind tunnel (Source: Owens et al., 2006)

Single degree-of-freedom tests require less coordination than the free-flight tests with the model being restrained about all axes except the one under consideration. A typical case for testing in wind tunnels at all speeds is to allow a single DOF in pitch only (Bernstein, 1975). Hicks (1993) discussed the use of a yoke-type setup, which permits 360 degrees of rotation in pitch. For testing, the model is positioned at a set attitude and held with light string, once the flow begins the model is released and its response recorded. Further data can be gained from this by using remotely operable controls that can be used to regain control of the aircraft thus simulating the aircrafts effectiveness in recovering from an unstable flight condition. Campbell (1963) researched the use of semi-free flight techniques for use in low speed wind tunnels, one technique was to supplement model thrust and support the models weight by use of two tethers. These tethers were then connected to a dynamometer to measure drag and lift. Bernstein (1975) discussed methods for testing in short-duration hypersonic facilities, two methods provided are; the use of a weakly restrained model, and a pivoted model. For a weakly restrained model a series of low stiffness springs are utilised to hold the model. These springs are sized so that they are sufficiently weak, and their effects on the model can be neglected. For a pivoted model a number of options are mentioned by Bernstein (1975), these include ball bearings, elastic flexures, gas-bearings and knife-edges. It is noted that the ball bearings and elastic flexures are not normally suited to hypersonic impulse tunnels due to the damping effects they both produce. In both the knife-edge and gas-bearing setups nylon string is used to initially position the model, once the flow starts the string is broken and the test begins.

2.4.2 Dynamic Scaling

Dynamic scaling is required to ensure that the data captured from a sub-scale model is most representative of the full-scale prototype. This means that the geometry of the model is scaled but also the dynamics of the model will need to have similitude with the full-scale prototype (Chambers 2010). If the dynamic
scaling process is undertaken correctly, the model will share geometrically similar flight paths and angular displacements with the full-scale prototype, however the time taken for these movements will be different (Chambers 2010). Furthermore, the flow angles, control surface deflections, Mach number, and if possible the Reynolds number should be matched between the model and full-scale prototype, for valid data (Owens et al. 2006). Depending on the speed of the flow and whether it can be considered compressible or incompressible will provide a different set of dimensionless scaling parameters to be used.

2.5 Newtonian Flow Theory

2.5.1 Justification for Use of Newtonian Flow Theory
Selection of a theory for analysis of a certain flow requires consideration of factors such as; the density of the flow, if the fluid can be considered inviscid or when viscous effects are dominant, whether the flow is considered incompressible or compressible, and the magnitude of the Mach number associated with the flow. Selection of Newtonian flow theory for use in the design of the model rocket will be justified by consideration of these points.

Flow Density
A brief definition of low-density flows was provided previously however it will be repeated here for clarity. Low-density flow also known as free molecule flow can be compared to a continuum flow. In low-density flow such as that experienced at high altitudes, the spacing between the fluid (air) molecules becomes very large compared to the size of the vehicle travelling through them. This means the object travelling through the fluid will not consistently experience impacts from fluid molecules, and so a different approach must be taken in design of a vehicle suited to this flow type. A continuum flow in contrast is where the molecule spacing is sufficiently small so that the fluid impacts the vehicle as if it is continuous (Anderson 2001).

Flow Viscosity
All flows do exhibit viscosity effects, however in many cases these effects can be neglected as they are either very small or irrelevant, this results in two flow types namely, inviscid and viscous flows. When a flow is considered to be inviscid the effects of ‘friction, thermal conduction, and diffusion’ (Anderson 2001) are ignored. This assumption is fairly accurate at sufficiently high Reynolds numbers as the viscous portion of the flow becomes sufficiently small, and will provide accurate results for pressure distributions and lift on the vehicle (Anderson 2001). To consider the effects viscous drag has on the surface the flow can be divided into two regions (Pritchard 2011). These are the viscous boundary layer, which occurs along the surface of the vehicle, and the surrounding inviscid flow, Figure 2.2 shows where these two different flow types occur.
With a sufficiently high Mach number or low Reynolds number this boundary layer thickness will increase until it begins to become a dominant part of the flow, in these cases the viscous effects must be considered as the main flow type. Other cases where viscous effects become dominant are in high angles of attack. The effects of this can be seen in Figure 2.3 where the flow has separated from the aerofoil and created a wake.

**Flow Compressibility**

Although no fluid is truly incompressible it can be considered as such if the ‘variations in density are negligible’ (Pritchard 2011). In the case of air it can quite acceptably be considered incompressible for Mach numbers of less than 0.3 as the density variations are less than 5 percent (Pritchard 2011). When Mach numbers approach the sonic region the compressibility effects become quite important to consider and in hypersonic flow they must be appropriately accounted for.

**Mach Number Magnitude**

Anderson (2001) discusses all the flow regimes from subsonic, transonic, supersonic and up to hypersonic however of interest will be the supersonic and hypersonic flows. A typically case for the shock layers forming in supersonic and hypersonic flows is pictured in Figure 2.4 and Figure 2.5 respectively.

---

**Figure 2.2: Viscous and inviscid flow regions over a body (Anderson 2001)**

**Figure 2.3: Flow separation at high angle of attack (Anderson 2001)**
consideration. Anderson (2001) notes that is the most straightforward theory to Mach numbers where the shock layer is used for sharp wave angle and the shock wave itself must be attached (Anderson 2006). This suite

The tangent-wedge method, tangent-wedge, which have surface inclination angles less than the shock wave angle and the shock wave itself must be attached (Anderson 2006). This means it is used for sharp-nosed bodies. Newtonian flow theory applies at high Mach numbers where the shock layer lays sufficiently close to the body under consideration. Anderson (2001) notes that is the most straightforward theory to

In both flow cases an attached shock wave is formed on the front of the vehicle. The main difference between the two is the angle this shock wave forms with the surface of the vehicle. When approaching hypersonic speeds the angle between the shock wave and the surface are small, at even higher Mach numbers this layer becomes so close to the surface that interactions between the viscous boundary layer and shock wave occur. At these higher speeds the vehicle also experiences high temperatures; in some cases these are sufficiently high enough to cause chemical reactions in the air.

Based on a discussion of these different flow types it is possible to narrow down which category the rocket model for testing in the USQ hypersonic wind tunnel falls under. The test conditions of the model will mean the flow will be a continuum, it will be compressible due to the flow velocity, and it will be in the hypersonic region; although it is in the lower range of the hypersonic region at Mach 6. It is unknown how significant the viscous effects along the rocket will be. These effects are only required for determination of drag and surface heating effects, in the case of this research the model will be restrained so that drag is of little concern and surface heating is not of interest. For this reason the flow will be considered inviscid. Anderson (2006) outlines a number of theories suited to these flow conditions, these include: Newtonian Flow theory, tangent-wedge method, tangent-cone method, and shock expansion method.

The tangent-wedge, tangent-cone and shock expansion methods are most suited to bodies, which have surface inclination angles less than the shock wave angle and the shock wave itself must be attached (Anderson 2006). This means it is used for sharp-nosed bodies. Newtonian flow theory applies at high Mach numbers where the shock layer lays sufficiently close to the body under consideration. Anderson (2001) notes that is the most straightforward theory to
use for prediction of surface pressures on a hypersonic vehicle. Although not as
accurate as other methods its simplicity makes it most desirable for rough
calculations of the rocket model body and control fin sizing.

2.5.2 Newtonian Flow Theory Methods
Now that the selection of Newtonian flow theory for design of the rocket model
has been justified, the theory itself will be further examined. As a result this
section aims to identify the techniques, which should be used to analyse bodies
in hypersonic flow. This will provide a base point from which a useful model can
be designed.

Newtonian flow theory was developed by Isaac Newton in 1687 and was initially
intended for use in low-speed flows; in these cases it produced very erroneous
results (Anderson 2006). More recently it was found that his theories provide a
simple approach to the determination of surface pressure over a hypersonic
vehicle. The theory assumes that as the fluid impacts a surface it transfers all of
its momentum in the direction normal to the surface but maintains its
momentum tangential to the surface. For a plain flat plate surface such as that
pictured in Figure 2.6, the following expression is obtained (Anderson, 2006):

\[ \frac{F_N}{A} = \rho_\infty U_\infty^2 \sin^2 \alpha \]  

(2.2)

Where, \( F_N \) (N) is the force normal to the surface, \( A \) (m\(^2\)) is some reference area,
\( \rho_\infty \) (kg/m\(^3\)) is the freestream density, \( U_\infty \) (m/s) is the freestream velocity, and \( \alpha \)
(rad.) is the vehicles angle of attack (AoA).

The \( \frac{F_N}{A} \) term in Eq. (2.2) is a pressure which is equal to the difference between
the surface pressure, \( p \) (Pa) and the freestream static pressure \( p_\infty \) (Pa):

\[ \frac{F_N}{A} = p - p_\infty \]  

(2.3)

With some manipulation this can all be represented in terms of a pressure
coefficient \( C_p \):

\[ C_p = \frac{p - p_\infty}{\frac{1}{2} \rho_\infty U_\infty^2} = 2 \sin^2 \alpha \]  

(2.4)

During testing the stagnation properties are measured and so the following
relations from Pritchard (2011) are required to determine the freestream
properties \( p_\infty \), \( \rho_\infty \), \( U_\infty \) from the known stagnation pressure, \( p_0 \) stagnation
density, \( \rho_0 \) stagnation temperature, \( T_0 \) and the freestream Mach number, \( M_\infty \).
\[
\frac{p_0}{p_\infty} = \left[1 + \frac{k - 1}{2} M_\infty^2 \right]^{\frac{k}{k-1}} \\
\frac{T_0}{T_\infty} = 1 + \frac{k - 1}{2} M_\infty^2 \\
\frac{p_0}{p_\infty} = \left[1 + \frac{k - 1}{2} M_\infty^2 \right]^{\frac{1}{k-1}}
\]

Also using the relationship from (2.1)

\[U_\infty = M_\infty \sqrt{kRT_\infty}\]  

For air \(k\) will be 1.4, and \(R\) will be 287 J/kg.K

![Figure 2.6: Newtonian flow theory for a flat surface (Anderson 2001)](image)

Equation (2.4) is the pressure coefficient acting perpendicular to the surface of the body and becomes most accurate as the Mach number goes to infinity. The term on the bottom of this equation is known as the free stream dynamic pressure \(q_\infty\) (Pa):

\[q_\infty = \frac{1}{2} \rho_\infty U_\infty^2\]  

![Figure 2.7: Separation of normal force into lift & drag components (Anderson 2001)](image)
Substituting this and the relationship \( F_N/A = p - p_\infty \) into the coefficient of pressure equation, results in the coefficient of Normal force equation:

\[
C_N = \frac{F_N}{q_\infty A} = 2 \sin^2 \alpha
\]  

(2.10)

This Normal force coefficient can further be broken-down into lift and drag components as shown in Figure 2.7. For the case of a plain flat plate the coefficient of lift can be defined as:

\[
C_L = \frac{F_L}{q_\infty A} = 2 \sin^2 \alpha \cos \alpha
\]  

(2.11)

and, the coefficient of drag as:

\[
C_D = \frac{F_D}{q_\infty A} = 2 \sin^3 \alpha
\]  

(2.12)

Grimminger, Williams and Young (1950) provide a derivation of these formulas applicable for use on a cylinder at an angle of attack to the flow. They show that again assuming the Mach number goes to infinity these formulas become quite close to experimental data. The normal force coefficient that they derived for a cylinder is:

\[
C_N = \frac{F_N}{q_\infty A} = \frac{16 l_s}{3 \pi d_s} \sin^2 \alpha
\]

This is then split into lift and drag components, defined as:

\[
C_L = \frac{F_L}{q_\infty A} = \frac{16 l_s}{3 \pi d_s} \sin^2 \alpha \cos \alpha
\]

and,

\[
C_D = \frac{F_D}{q_\infty A} = \frac{16 l_s}{3 \pi d_s} \sin^3 \alpha
\]

In the previously shown coefficient calculations it has been assumed that the Mach number is approaching infinity and therefore the maximum coefficient of pressure at the stagnation point on the nose of the body is 2. Mudford, O’Byrne, Buttsworth, Balage, & Neely (2013) undertook a study which derived a formula for a simple body with a semi-spherical nose and cylindrical body; they also note that Newtonian flow theory provides a good approximation to model behaviour. For their theoretical calculations they used modified Newtonian flow theory, which accounts for the more realistic cases where the Mach number is not infinity and therefore the maximum coefficient of pressure is not 2. Instead it
adds a $C_{p,0}$ term into the equations. This term represents the maximum pressure coefficient on a body, this will occur at the stagnation point on the nose of a cylindrical body. For the normal force on a cylindrical body using modified Newtonian flow theory, we obtain:

$$C_N = \frac{F_N}{q_\infty A} = \frac{4l_s}{3\pi r_s} \sin^2 \alpha$$

Where $l_s$ is the cylinder length, $r_s$ is the cylinder radius and the maximum pressure coefficient is defined by Anderson (2006) as:

$$C_{p,0} = \frac{2}{\gamma M_{\infty}^2} \left\{ \frac{(\gamma + 1)^2 M_{\infty}^2}{4\gamma M_{\infty}^2 - 2(\gamma - 1)} \right\}^{\gamma-1} \left[ \frac{1 - \gamma + 2\gamma M_{\infty}^2}{\gamma + 1} - 1 \right]$$

The lift and drag coefficients are then correspondingly:

$$C_L = \frac{F_L}{q_\infty A} = C_{p,0} \frac{4l_s}{3\pi r_s} \sin^2 \alpha \cos \alpha$$

$$C_D = \frac{F_D}{q_\infty A} = C_{p,0} \frac{4l_s}{3\pi r_s} \sin^3 \alpha$$

Ashby (1961) investigated the aerodynamics of 5 different hypersonic rocket configurations, this research showed good correlation between Newtonian flow theory and the experimental results. This research also provided rocket dimensions and data such as coefficients of drag, lift and pitching moment, and centre of pressure for the models. This information will prove useful in the design of the basic body for initial calculations.

2.5.3 Forces & Moments Location

Now that methods have been provided for determination of the pressures and forces on a surface, the point at which these forces act is required. Such a point is needed in order to determine how the model will respond to the oncoming flow. Critically it will allow an estimate to be made as to the forces expected on control surfaces and the body of the rocket but also provide a method to determine the models response.

Anderson (2001) defines the centre of pressure as the point where a single resultant force can be placed so that it represents all the distributed loads on the body due to pressure. Because the location of this point depends on the loads imposed on the body it will of course vary with control surface operation and with the models angle of attack in the flow. It can therefore prove difficult to find the centre of pressure location on a body. The location is frequently determined via experimental methods and data is provided for the particular model at various AoA. In the case of the standard NACA aerofoil profiles their
centre of pressure is not provided, rather they give lift, drag and moments about either the quarter chord point of the aerofoil or the aerodynamic centre (Anderson 2001).

If the pitching moment about a point and normal force are known for a particular body it is then possible to determine the centre of pressure location. Anderson (2001) provides the general relationship for the centre of pressure $x_{c.p.}$ (m) to be:

$$x_{c.p.} = -\frac{M}{F_N}$$

(2.13)

Where, $x_{c.p.}$ is the distance relative to the point where the moment $M$ (N.m) was recorded.

Furthermore for a vehicle to be regarded as statically stable the location of the centre of pressure must lay further towards the rear of vehicle than its centre of gravity. This ensures that without control surface input the vehicle will inherently return to a neutral angle of attack. Ennis (2013) notes that the value of this separation distance of the two points relative to the body length must be about 3 to 5 percent for uncontrolled hypersonic vehicles.
Chapter 3 – Theoretical Rocket Performance

3.1 Chapter Overview
The chapter will select a hypersonic body and fin design based upon available documentation for model coefficients and geometry. It then develops a generalised procedure for analysis of the model to determine tail fin actuation required to reach a specified pitch and the angular acceleration whilst undergoing the manoeuvre. Based on the model parameters determined later in Chapter 4 it will document some theoretical results at different AoA.

3.2 Body & Tail Fin Shape
Many different shapes and designs for hypersonic vehicles currently exist; the focus of this particular project however will be a classic rocket shaped body. This particular body is of interest for a number of reasons; it is geometrically simple thereby allowing a simple semi-analytical analysis of the body using well defined research, geometric simplicity also allows easy fabrication with available equipment, its response in the flow duration will be easy to monitor and the cross section of the body provides plenty of space to mount equipment and control surface actuators.

Rather than using only Newtonian flow theory to produce values of lift, drag, pitching moment, and centre of pressure at various AoA for the body, previous research will be utilised. Of particular interest is research by Ashby (1961) titled *Longitudinal Aerodynamic Characteristics of Five Hypersonic Missile Configurations at Mach Numbers from 2.01 to 6.01*, and research by McLellan, Bertram, and Moore (1951) titled *Investigation of Four Wings of Square Plan Form at a Mach Number of 6.9 in the Langley 11-inch Hypersonic Tunnel*.

3.2.1 Body Shape
Ashby (1961) investigated five different hypersonic vehicle configurations at a Mach number of 6.01. Both experimentally and through the use of Newtonian flow theory he produced plots of coefficients for pitching moments, axial force, centre of pressure, and normal force for all five bodies. The five models utilised the same basic body, which consisted of a tapered section and circular arc for the nose and a cylindrical after body. The first model was just this basic model, the second employed a 10 degree flared after body, the third had 4 equally spaced 10 degree fins, the fourth had 4 equally spaced 5 degree fins, and the fifth had a longer body with a 10 degree flared after body.

The basic first body used in this study is of interest here because it provides a basic platform to which control surfaces can be added. Ashby's (1951) data also compared his coefficient values with other documented values at various other Mach numbers for the same models. The results demonstrated that for this particular body there was little variation in coefficient values with Mach number change. This means that the Mach number at the TUSQ facility is not required to be exactly 6.01, as was used in the study, for the data to apply. The shape of
this basic body can be seen in Figure 3.1. For the original plots of pitching moment, axial force, centre of pressure, and normal force coefficients of this model see Appendix B. Also included are further geometric details on this model and the other 4 models used.

![Figure 3.1: Basic body shape of model (Dimensions in inches) (Ashby, 1951)](image)

### 3.2.2 Tail Fin Shape

McLellan, Bertram, and Moore (1951) undertook a similar study to Ashby (1961) where they investigated several simple wing profiles by comparing theoretical Newtonian calculations with experimental data. Investigated in this study were 4 wing profiles, which included a diamond, half-diamond, wedge, and half-circular-arc, these can be seen in Figure 3.2. Each of the profiles had a 4-inch chord length with an aspect ratio of 1 and a thickness of 5 percent of the chord.

![Figure 3.2: Simple wing profiles Investigated (McLellan, Bertram, & Moore, 1951)](image)

The finished model should desirably have no lift at 0 degrees AOA and 0 degrees of tail fin actuation. This is more typical of a rocket or missile type configuration and will also make analysis simpler. This suggests a symmetric wing profile should be used which means either the half-diamond or wedge should be chosen. In the case of the wedge shape wing profile the centre of pressure location data indicates that the centre of pressure does not vary considerably at varying angles of attack. This property makes theoretical analysis a lot simpler as its location can be assumed constant and in theory should reduce the aerodynamic torque about the wing pivot axis.

Because the thickness of the wing is only 5% of its chord, the thickness at the middle is very small. This makes it difficult to attach the wing to a pivot shaft, as there is a risk of the wing breaking off the shaft. Therefore this thickness will need to be increased and values for lift and drag determined analytically.
3.3 Procedure for Analysis of Response to Flow

A set of data has been obtained for the basic body of the model and calculations to determine wing data will be undertaken; this now needs to be combined in order to obtain a relationship between a desired model pitch and required tail fin actuation. Such information will be useful in validating the success of the experiment and will also give design parameters for the actuation system. The models expected angular acceleration whilst undergoing the manoeuvre will also be determined, this can later be compared to experimental data captured during testing. Furthermore it will allow the determination of time taken to perform the manoeuvre, thereby ensuring that it can be performed during the steady run time of the flow.

3.3.1 Model Force Nomenclature

To best assist the reader in understanding the forces the model is undergoing Figure 3.3 below displays the forces and their locations on a generalised model. The direction of the forces as shown in the diagram indicates their positive sense.

The locations shown are; $x_{c.g}$ the location to the centre of gravity, $x_{pivot}$ the location to the point where the model is allowed to pitch, $x_{cp.body}$ the location of the centre of pressure for the basic body (excluding fins), and $x_{cp.wing}$ the location to the centre of pressure for the wings. The forces shown are; W the weight of the model which acts at the centre of gravity, $F_{N.body}$ and $F_{A.body}$ which are the normal and axial forces on the body respectively, and $F_{L.wing}$ and $F_{D.wing}$ which are the lift and drag forces respectively, acting on the tail fin assembly.
The normal and axial force components are selected for use on the basic body because their direction is always perpendicular and parallel to the body longitudinal axis respectively, this simplifies calculations significantly. Whilst undergoing a manoeuvre the tail fin will be at an angle to the body centreline, this means that the use of normal and axial force components would require significant trigonometry to resolve them appropriately. Because the centre of pressure is assumed to remain constant for the selected wedge shape wing, the use of lift and drag force components simplifies the analysis. By placing the aerofoil pivot at the centre of pressure these force components will always act through the tail fin pivot and thereby simplifies the analysis of forces in the following sections.

3.3.2 Pitch Determination

With reference to Figure 3.3 an expression will be developed for the models required tail fin actuation to achieve a desired pitch. This equation will be developed based on the principle that the model will reach a steady pitch angle when the moment produced by the tail fin about the pivoting axis is balanced by the restorative moment produced by the basic body of the model. This will mean:

\[
\sum M_{\text{pivot}} = 0
\]

Now writing out all the forces and their respective moment arms we obtain the following expression:

\[
+ \sum M_{\text{pivot}} = -W(x_{\text{pivot}} - x_{c.g.})\cos(\alpha) - F_{N,\text{body}}(x_{c.p,\text{body}} - x_{\text{pivot}}) \\
+ F_{L,\text{wing}}(x_{c.p,\text{wing}} - x_{\text{pivot}})\cos(\alpha) \\
- F_{D,\text{wing}}(x_{c.p,\text{wing}} - x_{\text{pivot}})\sin(\alpha)
\]

(3.1)

It is also known from previous research that in general the dimensionless force coefficient can be presented as:

\[
C_x = \frac{F_x}{q_\infty A}
\]

Rearranging this expression in terms of the force:

\[
F_x = C_x q_\infty A
\]

(3.2)

Now substituting equation (3.2) into equation (3.1):

\[
+ \sum M_{\text{pivot}} = -W(x_{\text{pivot}} - x_{c.g.})\cos(\alpha) - C_{N,\text{body}} q_\infty A_s(x_{c.p,\text{body}} - x_{\text{pivot}}) \\
+ C_{L,\text{wing}} q_\infty A_p(x_{c.p,\text{wing}} - x_{\text{pivot}})\cos(\alpha) \\
- C_{D,\text{wing}} q_\infty A_p(x_{c.p,\text{wing}} - x_{\text{pivot}})\sin(\alpha)
\]

(3.3)
It can be seen from the above equation that there is no inclusion of the tail fin actuation amount $\beta$, this is because this value is determined via lift and drag coefficient charts for the wing. This is of course an iterative approach where a tail fin angle to flow amount is selected, the corresponding lift and drag coefficients are used to recalculate the above balance. When the moment balance is acceptably close to 0 the model is balanced at the desired AoA. The amount of tail fin actuation $\beta$ required, is then simply $\alpha$ plus the value read from the chart.

3.3.3 Angular Acceleration Determination
The analysis undertaken previously only considers the case where the model has reached an equilibrium position, there is of course a period where the model undergoes angular acceleration to change direction. Consideration of this will provide additional data for comparison with experimental results. Through further analysis it will also provide the expected time taken to undergo the manoeuvre, which can be used to confirm the suitability of the manoeuvre during the steady run time of the flow. In general the angular acceleration of a body about a fixed axis is related to the moment by:

$$\sum M_{\text{pivot}} = I_{\text{pivot}} \alpha$$

Where $\alpha$ (rad/s$^2$) is the angular acceleration of the body and $I_{\text{pivot}}$ (kg.m$^2$) is the mass moment of inertia for the body about the pivot axis.

Expressed in the form of a second order ordinary differential equation it is:

$$\ddot{\alpha} = \frac{\sum M_{\text{pivot}}(\alpha)}{I_{\text{pivot}}}$$

It should also be apparent that the sum of moments about the pivot is not constant and is a function of the pitch angle as was found in the previous section. This means that to determine the instantaneous acceleration at a certain pitch the function for the moments must be re-evaluated. Because of the iterative and repetitive nature of this calculation it is best performed via a computer program.

To determine the mass moment of inertia of the model about the pivot axis the use of 3D solid modelling software will need to be employed. This is due to the fact that the model is composed of many individual pieces many of which are of arbitrary shape and would prove difficult to evaluate manually.

The complexity of the integration required in obtaining $\alpha$, the model pitch as a function of time requires the implementation of Matlab's ODE solvers. To utilise these solvers requires that any ODE equation higher than first order be
represented as a set of coupled first order equations. For the equation of angular acceleration this can be achieved by defining two variables:

\[ x_1 = \alpha \]
\[ x_2 = \dot{\alpha} \]

Where \( \dot{\alpha} \) (rad/s) is the angular velocity and the first derivative of the AoA, \( \alpha \).

This means that:

\[ x_1' = \dot{\alpha} = x_2 \]
\[ x_2' = \ddot{\alpha} = \frac{\Sigma M_{\text{pivot}}(x_1)}{I_{\text{pivot}}} \]

The second order ODE is now presented in terms of two first order ODE’s and will allow straightforward implementation into Matlab’s ODE solvers.

### 3.4 Theoretical Values at Different AOA

Determination of theoretical dynamics for the model whilst undergoing a manoeuvre requires the implementation of a computer program. The particular program utilised was Matlab R2012a Student Edition for the creation of all scripts and functions. There are a number of sections to the development of the script documented here; they include processing of graphed data from research papers, determination of wing coefficient data, calculation of tail fin actuation for a specified pitch, model angular acceleration and manoeuvring time. Following this are some tabulated results, which were calculated using model design parameters defined in Chapter 4. All scripts are included in Appendix C.

#### 3.4.1 Processing of Graphed Coefficient Data

The body profile selected was chosen mainly because of the availability of force coefficient data. The experimental data (see Appendix B) is quite old and is only available in a graph format. In order for Matlab to utilise this data in calculations it must be manually entered. The general process for conversion of the data was to first carefully read each experimental data point by using a steel rule to measure its location in an attempt to maintain accuracy. Following this it was entered into a Matlab array as raw data and a function file was created. The function file was created to allow simple access to the data from other functions and the main Matlab script. Its basic operation is to accept an input angle in radians or an array of angles, which is then used to interpolate the corresponding coefficient value/s. A Piecewise Cubic Hermite Interpolating Polynomial or ‘pchip’ proved to be the best fit for interpolating the data points in all cases. Following this the interpolated data is provided as output from the function. Each of the function files had to be customised depending on the type...
of coefficient and how it will be used. The reproduced data points and fitted functions can be seen in Appendix B and the Matlab function files for each of the coefficients are reproduced in Appendix C.

**Body Normal Force Coefficient**
The normal force on the body should change direction as the body passes from a positive to negative AoA. Because this particular body is symmetrical the magnitude of this normal force coefficient should remain the same. This means that at positive AoA the function returns a positive coefficient value, and at negative AoA it will return the negative of the same values. The original data was provided in two forms, the first was a theoretical value calculated from Newtonian theory and the second was experimental force data. Both correlated well however the pressure data was selected as the functions raw data as it is most representative of the actual body.

**Body Axial Force Coefficient**
The body axial force will only vary in magnitude and not direction at varying AoA. The function file replicates this by using the absolute value of the angle entered. As with the normal force coefficients two different types of data were provided and the force data was selected.

**Body Centre of Pressure Location**
Theoretically for a symmetrical body the centre of pressure location should be the same for positive or negative AoA. This allows the provided data, which ranges from 0 to 30 degrees to be used at negative angles in this same range. To achieve this the function file uses the absolute value of the input angle to interpolate. Furthermore because the centre of pressure data is provided as a fraction of model length the function converts it to a position in meters by multiplying the data by the length.

Some of the original graphed data points for the centre of pressure as a fraction of body length were missing, however the normal force and pitching moment coefficients were available. The following simple equation allows the determination of centre of pressure \( x_{c.p.} \) as a fraction of body length using the pitching moment coefficient \( C_M \), normal force coefficient \( C_N \), body diameter \( d_s \) and body overall length \( L \) for the original model:

\[
\frac{x_{c.p.}}{L} = \frac{C_M d_s}{C_N L} + 0.5
\]

This was done for the missing values at 1 and 2 degrees AoA. At low angles of attack (between 1 to 4 degrees) the experimental centre of pressure location has large fluctuations causing it to vary from 50%, 20%, 40%, and 38% of the body length at 0, 1, 2, and 4 degrees AoA respectively. Although this fluctuation is large its effect on pitching moment will be minimal because the normal force coefficient is very small at these AoA and the tail fin moments will dominate.
3.4.2 Determining Nozzle Exit Conditions
Professor David Buttsworth (2014a) provided the researcher with a simple script for the determination of the nozzle exit conditions for the hypersonic facility. The recorded data for each run is the stagnation pressure, stagnation temperature, and the Mach number, for analysis in proceeding calculations the freestream conditions are required. The script uses the hypersonic flow relationships presented earlier in Equations (2.5), (2.6), (2.7), and (2.8).

3.4.3 Calculating Wing Coefficient Data
Buttsworth (2014b) also provided the researcher with a script that determines the surface pressure to freestream static pressure ratio, \( p/p_\infty \) on the surface of a simple wedge shape wing. In its original form the script estimated these pressure ratios for the upper surface of the wing facing the flow and lower surface of the wing facing away from the flow at AoA from 0 degrees to half of the wedge angle. The script would not calculate any further values on the lower side of the wing as once the wing was at the wedge angle the lower surface would not be facing the flow and would be subject to a pressure ratio of less than 1, indicating that the surface pressure is now lower than the static freestream pressure. A condition that the theory used could not predict.

Data for the wing is required at AoA in a greater range than this and so the researchers solution was to make a simplifying assumption about the behaviour on the lower side of this wing. Interpolation of the result from where the pressure ratio of 1 occurs provides a good approximation for pressure ratios at higher AoA where the ratio is less than 1. Once the ratio decreased even further it was held at a value of 0.5, this value was selected because McLellan, Bertram, and Moore (1951) used this for their analysis of a wedged wing and their results showed good correlation with experimental data.

The pressure ratio data that was obtained now needs converting into normal force coefficient data to allow determination of lift and drag coefficients. This is achieved through use of Equations (4.1) and (2.10) where

\[
C_N = \frac{F_N}{q_\infty A}
\]

and

\[
\frac{F_N}{A} = p - p_\infty
\]

Combining the two equations and simplifying it can be seen that

\[
C_N = \frac{p - p_\infty}{q_\infty}
\]
Where $p_\infty$ and $q_\infty$ are calculated from the flow conditions and $p$ is determined from the ratio $p/p_\infty$. Figure 3.4 shows where these upper and lower normal force coefficients act on the wing.

![Figure 3.4: Wing nomenclature for lift & drag coefficient calculation](image)

**Wing Lift Force Coefficient**

With reference to Figure 3.4 some simple trigonometry will allow the $C_{N,\text{upper}}$ and $C_{N,\text{lower}}$ to be resolved into a total lift coefficient component. Assuming a downward force is positive:

$$C_{L,\text{wing}} = C_{N,\text{upper}} \cos \left( \phi + \frac{\phi}{2} \right) - C_{N,\text{lower}} \cos \left( \phi - \frac{\phi}{2} \right)$$

Where $\phi$ is the wing AoA to the flow, and $\phi$ is the wedge angle of the wing.

Similar to the normal force coefficient for the body the wing lift coefficient will only vary in direction and not magnitude for positive and negative AoA due to the symmetrical shape of the wing.

**Wing Drag Force Coefficient**

The drag force coefficient is determined in a similar way to the lift force where a force to the right is assumed positive:

$$C_{D,\text{wing}} = C_{N,\text{upper}} \sin \left( \phi + \frac{\phi}{2} \right) - C_{N,\text{lower}} \sin \left( \phi - \frac{\phi}{2} \right)$$

The drag force on the wing will only vary in magnitude and not direction regardless of wing AoA to the flow.

**3.4.4 Matlab Script for Tail Fin Actuation Required**

Determination of the tail fin actuation required is achieved using the moment summation about the pivot point defined in equation (3.3). This moment summation is turned into a function file, with the moment summation being entered exactly as it appears in equation (3.3). This means that the direction of angles and forces are positive for the directions shown in Figure 3.3. If a negative AoA or pitch down manoeuvre (opposite to that Figure 3.3) was desired to be represented by the function the reverse of sign for the different
forces would occur within the function files for the force coefficients described earlier. The function accepts inputs of AoA and tail fin actuation, $\alpha$ and $\beta$ respectively in Figure 3.3 and outputs the resulting moment about the pivot axis.

By trialling a constant desired AoA value and a range of tail fin actuation values in the above-defined function a series of resulting moments about the pivot axis will be obtained. The value of tail fin actuation for steady state at the desired AoA is the one that makes the pivot moment closest to zero. Because only a selected range of values for control fin actuation are used the calculation is limited to AoA, which do not require actuation outside this range. This is not of concern, as the range of tail fin actuation values used will exceed the expected operating angles.

3.4.5 Matlab Script for Tail Fin Actuation Rate
The high speed with which the manoeuvre will occur means that the time taken for the fin to actuate is an important factor to consider. The script is set up to simulate a case where the fin would actuate linearly to the required position, it will then hold for a chosen duration, then it would begin returning to its initial position at the same linear rate.

It accepts an input of time, which is used to determine what stage the fin is at, either; actuation, delay, or returning and at what point of the stage it is at. The time by which the fin should be in the fully actuated position is determined from the required actuation amount, $\beta$ divided by the rate at which the fin can be actuated, $\dot{\beta}$. While time is still less than this value the value of fin actuation rate is multiplied by the time. When this time is exceed but by no more than the delay, the fin angle is held constant at the required actuation amount. When this is then exceeded the fin begins to return to its initial position.

3.4.6 Matlab Script for Angular Acceleration & Manoeuvre Time
By evaluating the pivot moment and dividing by the mass moment of inertia the angular acceleration will be obtained. The pivot moment is dependant upon the current AoA and the current tail fin actuation amount, which in turn are both dependant upon time.

This is a second order ODE problem and so the equations are presented in terms of two first order ODE’s as shown in the previous section. This function is setup to suit Matlab’s built in ODE solvers; therefore it takes inputs of time, and position. It outputs a vector, which contains the first and second derivatives of AoA. The ODE45 solver was selected, as it is the most commonly used and the others provided no advantages for this application. The solver requires inputs, which include the function containing the ODE, the time range for analysis, and initial conditions for the start of the time range. These initial conditions are the initial AoA and the initial angular velocity.
3.4.7 Results of Theoretical Analysis
Theoretical calculations undertaken here were directly influenced by information found throughout design of the model covered in Chapter 4. Also design of the model required information from the theoretical calculations and this necessitated that both aspects be considered simultaneously. For this reason the model parameters selected for use in the Matlab script are presented here with the full justification of their selection covered in Chapter 4.

Model Parameters
The general dimensions of the model require scaling from the dimensions that Ashby (1961) used in his research to suit the usable manoeuvre space in the flow, the scale required for the body is 5/9 of the original used. In addition the wing dimensions were selected based on manufacturing constraints, as they need to be have sufficient thickness for mounting onto a shaft.

Table 3.1: Selected model geometrical parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body Diameter, $d_s$</td>
<td>25.4 mm</td>
</tr>
<tr>
<td>Body Cross sectional area, $A_s$</td>
<td>506.7 mm$^2$</td>
</tr>
<tr>
<td>Body Length, $L$</td>
<td>254 mm</td>
</tr>
<tr>
<td>Wing Chord</td>
<td>50 mm</td>
</tr>
<tr>
<td>Wing Width</td>
<td>25 mm</td>
</tr>
<tr>
<td>Wing Planform Area, $A_p$</td>
<td>2500 mm$^2$ (total for both wings)</td>
</tr>
<tr>
<td>Wing Wedge Angle, $\phi$</td>
<td>8.6 degrees (15% of chord)</td>
</tr>
<tr>
<td>Wing Pivot Location, $x_{piv}$</td>
<td>222 mm (from model nose)</td>
</tr>
</tbody>
</table>

Determination of these geometrical parameters and in addition the on-board equipment weights and positions allowed determination of the following mass related parameters (See Appendix G for Detail Drawings).

Table 3.2: Determined model mass parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centre of Gravity, $x_{cg}$</td>
<td>159.1 mm (from model nose)</td>
</tr>
<tr>
<td>Body Pivot Location, $x_{piv}$</td>
<td>96 mm (from model nose)</td>
</tr>
<tr>
<td>Total Mass, $m$</td>
<td>119.8 g</td>
</tr>
<tr>
<td>Mass moment of Inertia, $I_{piv}$</td>
<td>$1.049 \times 10^3$ kg.mm$^2$ (about pivot)</td>
</tr>
</tbody>
</table>

In addition to these parameters there are also some limitations on the rate at which the motor can actuate the fins. The motor performance was determined through testing. The time which the motor will hold for at the fully actuated position is nominally set at 10 ms as it is unknown what effects this will have on the experiment.

Table 3.3: Actuation stepper motor parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor Actuation Rate, $\dot{\beta}$</td>
<td>180 degrees/s (1:10 Gearbox)</td>
</tr>
<tr>
<td>Hold Time Delay</td>
<td>10 ms</td>
</tr>
</tbody>
</table>
Widodo and Buttsworth (n.d.) undertook testing in the TUSQ facility, which is the facility that will be used for this work. They recorded typical flow conditions presented in Table 3.4 that will be used for this initial analysis.

Table 3.4: Typical TUSQ flow conditions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freestream Mach Number, $M_\infty$</td>
<td>5.84</td>
</tr>
<tr>
<td>Stagnation Pressure, $p_0$</td>
<td>0.92 MPa</td>
</tr>
<tr>
<td>Stagnation Temperature, $T_0$</td>
<td>572 K</td>
</tr>
<tr>
<td>Ratio of Specific Heats, $k$</td>
<td>1.4</td>
</tr>
<tr>
<td>Ideal Gas Constant, $R$</td>
<td>287 J/kg.K</td>
</tr>
</tbody>
</table>

Results

Now that the parameters are defined for the model and entered into the script it is possible to simulate the response for different scenarios. Of interest are those responses, which occur when the model is undertaking realistic manoeuvres therefore the AoA scenarios to be investigated are 5, 10, and 15 degrees.

The information of interest is the required total tail fin actuation for each of these responses, the total time taken to complete the manoeuvre, AoA during the simulation, angular acceleration of the body, and forces on the model surfaces. This selection of data provides enough information to make informed design choices in Chapter 4 and also provides a starting point for comparison with experimental results. In addition to this information Figure 3.5 shows the calculated values for the amount of tail fin actuation required to achieve a certain AoA. This provides a quick reference to see expected requirements that the actuation system must deliver.

Rather than analyse all three AoA scenarios here the 10-degree case will be investigated, with the data for the other cases included in Appendix D. Key properties of all three cases are then summarised in Table 3.5.

Figure 3.6 shows plots of $\alpha$ the models AoA, $\beta$ the tail fin position relative to the body, and $\phi$ the tail fin position relative to the flow. Starting from a time of 0 the tail fin begins actuating, its position relative to the flow peaks at 19 ms and a value of 1.8 degrees. After this point the body is changing its AoA very rapidly and actually overtakes the tail fin movement at 35 ms. It reaches the desired angle of attack at 46 ms but due to angular momentum it overshoots this position by 2 degrees. The body then begins its return to a 0 degree AoA and gets to its equilibrium position at about 150 ms, 15 ms after the tail fin is finished actuating. In actual fact the body will not quite get back to its original 0 degree AoA as when the fin is at a 0 degree angle the body will sit pointing upwards about 1 degree. This is because the centre of mass of the body does not coincide with pivot point. In addition the effects of not including damping also become clear after this point as the simulation indicates that the body will continue to oscillate indefinitely. Because it is not possible to readily estimate
the damping effects they will be inferred from the experimental data obtained by testing. By observing the rate at which the actual oscillations die out this can be replicated through trial and error in the simulation.

In conjunction with Figure 3.6, Figure 3.7 displays more data that will be compared with the experimental results. This plot shows expected angular acceleration of the model during time. Again after 145 ms when the manoeuvre is complete the model still continues to oscillate because of the exclusion of damping in the analysis.

Figure 1.8 provides the expected body and control surface loads, it is not intended to use this data for comparison with experimental results but rather for the design of the model and tethering system. The body axial force and wing drag force both stay approximately constant whereas the body normal force and wing lift force both have large changes. Body normal force peaks when the models AoA peaks and the restorative moment is at its maximum. Similarly the peak tail fin forces occur when the tail fins position to flow is at a maximum.
Figure 3.6: Model AoA & tail fin positions expected during a 10 degree manoeuvre

Figure 3.7: Model angular acceleration expected during a 10 degree manoeuvre
In summary the following values for 5, 10, and 15 degree positive, nose up manoeuvres, were found:

Table 3.5: Summary of results for 5, 10, and 15 degree AoA scenarios

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Angle of Attack Desired</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5°</td>
</tr>
<tr>
<td>Tail Fin Actuation Required</td>
<td>4.5 degrees</td>
</tr>
<tr>
<td>Total Manoeuvre Time</td>
<td>70 ms</td>
</tr>
<tr>
<td>Tail Fin Lift Max.</td>
<td>6.6 N</td>
</tr>
<tr>
<td>Tail Fin Drag Max.</td>
<td>1.3 N</td>
</tr>
<tr>
<td>Body Normal Force Max.</td>
<td>4.9 N</td>
</tr>
<tr>
<td>Body Axial Force Max.</td>
<td>0.9 N</td>
</tr>
</tbody>
</table>

A few points to note about the results presented in Table 3.5, firstly for the 5 degree manoeuvre the tail fin only requires 4.5 degrees of actuation. This means that when the model is at 5 degrees AoA the fin will be pointing upwards attempting to rotate the model back to a 0 degree AoA. This is because the models C.G sits behind the pivot point and at low angles of attack the fin has to compensate for this. The durations for the 5 and 10 degree manoeuvres are both sufficiently short to occur within the flow duration however the 15-degree manoeuvre will likely not be completed during the steady flow time.
Chapter 4 – Methodology

4.1 Chapter Overview
Using the theoretical calculations developed and undertaken in the previous chapter the model will be designed. A few concepts are discussed with the selected design developed in detail. Detailed working drawings for the model and support system are also produced at this stage and presented in Appendix G. A brief description of the TUSQ facility and experiment equipment is provided. The steps used in preparing the model and undertaking a typical run is discussed. Finally the post processing of collected data is covered.

4.2 Model Design Parameters
Design of the model is controlled by the available area for testing in the flow, the size of equipment to be mounted inside the model, and the requirement to maintain the length to diameter aspect ratio of the model for previously documented experimental data to apply. After these parameters are determined the model housing which will encase all the equipment can be developed. Finally the support system for the model restraint can be designed.

4.2.1 Model Scale
TUSQ tunnel cross section places a limitation on the overall model scale and more importantly, the effective core diameter of the flow produced by the Mach 6 nozzle provides a limit to the available space for a manoeuvre in the tunnel. Another factor to consider is the size of sensory, actuation, and microcontroller equipment, which needs to fit inside the model cavity. The body design chosen for the model utilises an aspect ratio of 10, for the coefficient data utilised in previous calculations to apply the scale of the model must maintain this ratio.

The flow has an effective core diameter of approximately 160mm at the exit of the Mach 6 nozzle. As the flow exits the nozzle it develops into a Mach cone. The area inside of this cone is at the desired testing velocity whilst outside of the cone is not fully developed and unsuitable for testing. The flow Mach number $M_{\infty}$ and angle $\delta$ that the cone forms with the flow are given in Pritchard (2011) as:

$$\delta = \sin^{-1}\left(\frac{1}{M_{\infty}}\right) \quad (4.1)$$

For the Mach 5.8 flow experienced in the TUSQ facility this gives an angle of 9.9 degrees and a cone shape as shown in Figure 4.1, where the hatched area indicates the desired testing space.

For the experimental data to be valid the model must remain within this core flow whilst performing its manoeuvre. It was found from drawing different scale models at various AoA that a 10-inch (254 mm) long model should be capable of performing up to a 10 degree pitching manoeuvre without exiting the Mach
cone. A 10-degree pitching manoeuvre should be sufficient for replicating the dynamics of any real hypersonic vehicle. The scale, which provides a 254 mm overall length, is 5/9 of the original used by Ashby (1961). This means the original outside diameter of 1.8 inches (45.7 mm) will be reduced by 5/9 resulting in a 1 inch (25.4 mm) outside diameter.

![Mach cone for Mach 5.8 flow in TUSQ facility](image)

**Figure 4.1:** Mach cone for Mach 5.8 flow in TUSQ facility

### 4.2.2 Test Section Mounting Points
To tether the model there is a number of drilled and tapped mounting holes on the end of the nozzle parallel to the flow (Figure 4.2 (a)) as well as a mounting plate with drilled and tapped holes perpendicular to the flow (Figure 4.2 (b)). The plate can be positioned above or below the nozzle outlet.

![Outlet nozzle mounting holes](image)

- **Figure 4.2:** (a) Outlet nozzle mounting holes, (b) Mounting plate positioned below nozzle

### 4.3 Initial Design Concepts
Two main components of the system are to be developed. These are the tail fin actuation system and the support system for inside the test section. Constraints on the models size and the available test section mounting points were identified previously and will provide scope to the design of these two systems.
4.3.1 Tail Fin Actuation System Concepts

Design of this system is an original part of this research, and is a critical component to allow the experiment to proceed. To be successful this system must meet a number of key design parameters, including:

- Allow incremental rotation between 0 and 20 degrees relative to the model axis, this will allow AoA in the range of about 0 to 15 degrees
- Be sufficiently compact to fit inside the models cavity of 25.4mm outside diameter
- Durable enough to withstand expected control surface forces
- Ability to be triggered via the onset of flow
- Have a quick actuation time which occurs well within the steady flow period of approximately 200 ms duration
- Hold the fin steady at the desired actuation position
- Be manufactured from readily available and economically viable componentry.

To achieve this two systems have been considered for use as the control surface actuator. The first was based around the use of a linear solenoid and the second based around a stepper motor.

Linear Solenoid

The main component of this system would be a linear solenoid similar to that pictured in Figure 4.3. These operate by energizing a coil, this causes the iron bar through the centre to extend or contract depending on the current through the coil. These solenoids are available in small sizes normally about a 20mm x 20mm cross-section and normally extend about 10mm. It is not possible to control the amount of extension that this solenoid provides and so for this application it would require a mechanical stop which can be pre-set at the desired control surface actuation angle.

![Figure 4.3: Typical linear solenoid (Trossen Robotics, n.d)](image)

The solenoid would be mounted parallel to the longitudinal axis of the model and would actuate a linkage connected to the tail fins. This will allow the linear motion to be converted to the desired rotational motion.

This particular option benefits from a low cost solenoid, which has a fairly simple operation. It is expected to be reliable and provide accurate tail fin
positioning. It is unfortunately limited in a few aspects; adjustment of the control surface deflection will likely require disassembly of the model, a short stroke length of about 10mm will mean it will be difficult to rotate the control surface to some of the higher angles required, and it is not possible to effectively adjust the actuation speed.

**Stepper Motor**
This system would utilise a stepper motor similar to that in Figure 4.4. A stepper motor is an electric motor with the ability to be positioned at a number of equally sized steps in a full rotation. They achieve this by utilising two sets of coils inside the motor, by alternately energising these sets of coils the motor can be stepped through its rotation and has the ability to be stopped and held at any of these steps. Typical step sizes include 1.8 degrees (200 steps/rev) and 0.9 degrees (400 steps/rev). It is also possible to increase the step resolution to 1/2, 1/4, 1/8, 1/16, and 1/32 of a regular step by varying the current in each of the coils; this is achieved via a microcontroller. Stepper motors are readily available in sizes of 28x28 mm and 20x20 mm.

![Figure 4.4: Typical stepper motor (Pololu: Robotics & Electronics, 2014a)](image)

The stepper motor, like the linear solenoid, would be installed with its axis parallel to the longitudinal axis of the model. It would then need to transfer its motion through 90 degrees to allow rotation of the tail fins.

The benefits from this design are the large versatility arising from the ability to position the fins at a large range of angles. It will also have accurate positioning as this is controlled via a microcontroller. The implementation of this concept will however prove more complex due to; the 90-degree gearbox required and additional programming to enable positioning of the stepper motor and the use of increased step resolutions. Stepper motors of a suitable size are also less common with a readily available 20x20 mm stepper motor still too large for the 25.4 mm cylindrical body.

### 4.3.2 Model Pitching Support Concepts
Equipment mounted inside the model is relatively costly in both fabrication time and component cost and so damage to it should be avoided, for this reason it is not desired to test the model in free flight. This means that some form of restraint must be provided, whilst still allowing the desired pitching motions to occur. Two different setups have been considered the first is a yoke type setup,
which would allow rotational motion only; the second is a tethered setup that would allow rotational motion, and potentially some yawing.

**Yoke Support**
The yoke support would hold a shaft between the two prongs of the yoke; the model would then mount and pivot about this shaft. This general concept is illustrated in Figure 4.5. The yoke support would need to sit clear of the flow interactions that will occur around the body and fins of the model. The shaft will also need to be thick enough to withstand the bending force caused by drag and lift on the model.

![Figure 4.5: Yoke type pitching support concept](image)

It is likely that flow over the relatively thick shaft (around 2 – 3 mm) will result in downstream disturbances. These disturbances will affect flow over the remainder of the body and critically, over the wings. If sufficiently large this could cause errors in the performance of the model, making the experimental data inexplicably vary from the semi-analytical analysis.

**Tether Support**
A tether support would utilise thin wire, which would essentially simulate model thrust, and supplement the models lift. It would allow pitching motions to occur and to some extent would allow roll and yaw of the body. Two wires are required as initial calculations show that relying on a single horizontal tether will allow excessive changes in the models elevation with changes in pitch and as a result would not remain in the Mach cone for the manoeuvre duration. A tether system with sufficiently thin support wire should not cause any large disturbances to the downstream flow.

### 4.4 Design Choice & Development

**4.4.1 Tail Fin Actuation System**
Through consideration of the two options it was decided that a stepper motor-based system would be utilised. This was chosen because of its versatility in range of possible angles, ability to control the speed of the movement, and ability to control the oscillation of the movement. Although the creation of such a system is more complex than if it incorporated a linear solenoid it will provide a platform for future expansion because of its versatility. This system can be
further divided into 3 main sections; the stepper motor, the 90 degree gear system, and the control electronics.

**Stepper Motor**

Because the 20x20 mm motor is still too large to fit inside the model cylindrical body a much smaller stepper motor had to be sourced. Searching online revealed a range of much smaller stepper motors with body diameters of 6 mm, 10 mm, and 15 mm all of which would be suitable to fit inside the model. Typically these small stepper motors would be used in consumer electronics such as DVD/CD drives, camera lens auto focus mechanisms, scanners, and printers.

Availability of these small motors was only through websites such as Ebay and delivery was from China. Information available on the motors was also limited and lacked data such as rated voltage, rated current, step angle, torque, and operating speed. Because of long delivery times and low cost a selection of different motors were purchased. Ten each of the 6 mm, 10 mm and 15 mm motors were purchased and a sample of each can be seen in Figure 4.6.

![Stepper motors purchased: from left to right 15mm, 10mm & 6mm](image)

The 15mm stepper motors were focused on initially as they would provide the greatest torque. Some testing of the motors whilst developing the microcontroller system (see Appendix F) allowed determination of some suitable motor operating parameters. At a battery pack voltage of about 8.2 VDC the motors current draw was about 250 mA and it never got hot to touch. This suggests that the coils were not operating at excessively high power, which would cause damage and therefore these are suitable parameters. A current draw of 250 mA is also low enough for the batteries to supply.

The other parameter of interest was the motors torque at these operating conditions. An approximate figure for this was found by using the setup seen in Figure 4.7. By attaching lead weights to a lever arm attached to the motors shaft a maximum torque could be calculated. For the motor at a voltage of 8.2 VDC and a current draw of 250 mA the motor could lift a single 3.1 g lead weight at a distance of 75 mm from the motor shaft centre. This is a torque of 232 g.mm or 2.3 mN.m. It is also known from preliminary theoretical calculations that expected wing lift forces are in the order of 7 N (see Table 3.5)
at their greatest. However, because the wings centre of pressure stays almost constant at 50% of the chord length the actual torque produced about the wing pivot will be very low. Nonetheless it is still likely this may deviate even a few millimetres either side of the expected location which would cause torque values of about 14 mN.m assuming a 2 mm shift of the centre of pressure either side of the pivot.

![Figure 4.7: Determining approximate torque of a 15mm stepper motor](image)

At an attempt to increase the available torque from the motor the current was increased to about 600 mA whilst maintaining the same voltage of 8.2 VDC. This higher current draw causes the motor to rapidly heat and it is likely that extended periods of operation would cause motor damage. However an increase in torque was observed as the motor could now lift two of the 3.1 g lead weights at a distance of 75 mm. This corresponds to double the torque that was produced before with a value of about 4.6 mN.m. Unfortunately these values are still insufficient to overcome the expected control surface torque of about 14 mN.m and this will need to be addressed through gearbox selection.

**Gearbox System**

Due to the size of the stepper motor and available mounting space its orientation must be so its shaft is parallel with the longitudinal axis of the model. This necessitates the use of a 90-degree gearbox to connect the motor to the wing pivot axis. In addition the motors torque output is very low and needs to increase about 8 times to match the expected control surface forces. This will be achieved through two stages; the first is the attachment of a reduction gearbox to the stepper motor. The second is the use of a pair of identical bevel gears, one attached to the gearbox output shaft, the other attached to the pivot shaft of the control surfaces.

The reduction gearbox will increase the output torque at the expense of angular velocity. Rather than manufacture the gearbox it has been sourced online. It is a micro-metal gearbox, which would typically attach to a small electric motor however it is a direct fit to the 15 mm stepper. The stepper motor with gearbox and bevel gear attached can be seen in Figure 4.8. Ratios of interest are a 1:10 and a 1:30 reduction; this will mean each 18-degree step of the motor will result in a 1.8 or 0.6 degree step for the two gearboxes respectively. In addition the output torque will be increased by a factor of 10 or 30 via the two gearboxes.
The limiting factor is the decreasing angular velocity, as the actuation must occur within the 200 ms flow duration. From investigation of the motors (see Appendix F) it was found the highest step rate that could be achieved without skipping steps was 6 ms. For a large pitching manoeuvre (about 10 degrees) the expected required fin actuation is about 15 degrees, with a 1:10 reduction this would mean 8.3 steps of the motor would need to be taken which gives a duration of 50 ms. This is an acceptable time as the fin could actuate and return during the flow duration. With a 1:30 reduction this time will be tripled to 150 ms for actuation and there would likely be insufficient flow time to capture the return of the fin. If testing shows control surface loads necessitate the use of the 1:30 gearbox the fin actuation amount and therefore the pitch angle will need to be reduced.

![Stepper motor with attached micro-metal gearbox & bevel gear](image)

*Figure 4.8: Stepper motor with attached micro-metal gearbox & bevel gear*

A small bevel gear will be attached to the gearbox output shaft and a second mating gear will be attached to the control surface pivot shaft perpendicular to the models longitudinal axis. The gear itself is a 16 tooth plastic bevel gear with a diameter of 11 mm and modulus of 0.5 (pitch circle diameter of 8 mm). It will fit easily inside the models cavity and will press fit onto the control surface and gearbox shafts. This gear system arrangement can be seen in the component layout Figure 4.13.

**Control Circuitry**

To drive the stepper motor motion and receive trigger inputs from the test environment requires a few electrical components. In particular there are 4 key components used, these are the microcontroller, stepper motor driver, microswitch trigger, and the power source. Appendix F details the development and testing of the package that will be described here.

The microcontroller is the central component, which has the ability to receive inputs, and based on these inputs performs specified actions. These functions are preprogramed into the board by the user and allow a vast array of actions to be achieved. The platform chosen was Arduino because of its simple programming interface, large amount of online technical support, and because of the researchers previous experience with this platform. Further information on the Arduino platform can be found at 'http://arduino.cc/' (Arduino, 2014). The board selected is an Arduino compatible A-Star 32U4 Micro manufactured by
Pololu Robotics and Electronics. The main reason for selecting this board was because of its size, Figure 4.9 shows the board and its main dimensions. It measures 1” (25.4 mm) long by 0.6” (15.2 mm) wide by 0.18” (4.6 mm) thick and weighs only 1.3 grams, altogether the unit is very light, compact, and fits easily inside the model's cavity. In addition, the board has an inbuilt voltage regulator to allow a supply voltage of 5.5 V to 15 V meaning it does not require a separate voltage regulator. Further details and data sheets for this board can be found at the Pololu Robotics & Electronics (2014b) website.

![Figure 4.9: Pololu A-Star 32U4 Micro, dimensions in Inches (Pololu: Robotics & Electronics, 2014b)](image)

A microcontroller on its own could not provide sufficient current to power a stepper motor and so a stepper motor driver needs to be employed. The driver serves two purposes the most important is its ability to allow a higher current draw to operate the motor, however it also allows for simple control of the stepper motor. The particular driver chosen is seen in Figure 4.10 and is a DRV8834 Low-Voltage stepper motor driver carrier from Pololu Robotics & Electronics. Similarly this unit was selected because of its compact design measuring 0.8” (20.3 mm) long by 0.6” (15.2 mm) wide by 0.12” (3 mm) thick, and weighing only 1.6 grams. The driver allows for a wide range of operating voltages between 2.5 V up to 10.8 V and can handle peak currents of up to 2 A. Control is also very simple as the driver accepts inputs for step and direction as well as different microstep resolutions of 1/2, 1/4, 1/8, 1/16, and 1/32. Further details and data sheets for this board can be found at the Pololu Robotics & Electronics (2014c) website.

![Figure 4.10: Pololu DRV8834 Stepper Driver (Pololu: Robotics & Electronics, 2014c)](image)

With the two above-mentioned boards it is possible to provide full control of the stepper motor. The device also needs an input from its environment to signal when the test flow has started and then as a result triggers the control surface actuation. This is to be achieved via a microswitch which will have its contacts held closed via a thin support string, which will be attached, inside the test section. At the start of the flow the string will be broken, which in turn allows the
switch to release and the program will be triggered to begin. To achieve this a small lever type micro switch was selected, in particular it was a D2F-L Omron snap action switch available through Little Bird Electronics (2014) seen in Figure 4.11. Overall the switch is compact and is about 14 mm long (including the lever) by 6.5 mm high (not including the terminals) by 6 mm thick, and weighs only 0.5 grams.

Figure 4.11: D2F-L Omron Snap Action Switch

The final electronic component is the battery pack. To be acceptable the battery pack used must provide a voltage greater than 5.5 V to power the microcontroller, it must be light so as to not create excessive weight, it must be compact to fit inside the models cavity and finally it must be rechargeable to permit reuse of battery pack for multiple runs. To meet these criteria a custom battery pack was formed from two CR2, 3.6 V lithium ion rechargeable batteries. The batteries are linked in series to provide a nominal voltage of 7.2 V with a capacity of 600 mAh. Each battery has a total length of 27 mm, a diameter of 15 mm, and a weight of 10 grams.

Connections between the components were made as per Figure 4.12 below. The pin layouts shown and references are representative of the actual boards. The developments, which lead to this circuit layout, are discussed in Appendix F.
4.4.2 Model Housing

Housing of all the components proves quite difficult because of the constraints on available space. Furthermore components such as the batteries, microswitch, and microcontroller must be accessible to allow setup of the model. The weight of each item and its position also needs to be considered to ensure the models total weight remains low for a quick response. The housing will be manufactured by use of 3D printing, this has proven in the past to be a quick and accurate method of producing complex geometries that would be difficult to create via traditional machining.

After much consideration the layout in Figure 4.13 was decided upon.

Figure 4.13: Model cross-section showing componentry layout

Because the microswitch, batteries, and microcontroller all need to be accessible the model must be made from multiple pieces which can be split apart for access. The easiest option would be to provide a split along the longitudinal axis of the housing, however previous experience by Ennis (2013) showed that during 3D printing the body warped due to variations in the print environment. For this reason the body will be split perpendicular to the longitudinal axis. The splits will be made at the base of the nose section where it joins to the body and also at the very rear of the models body. The nose is kept in one continuous piece as a join along its surface will likely lead to flow disturbances that would affect the resulting data.

Figure 4.14: 3D printed model housing with components assembled

The split at the base of the nose will allow access to remove the batteries for charging. On the other side of the split will be the microcontroller with its USB port orientated towards the opening. This will allow access for modifying the
program without removing the microcontroller. Behind this is the microswitch, which has a thin wire, attached to its actuator, this leads out the rear of the model, through the motor end cap. The removable end piece is required for the installation of the stepper motor and fin drive system. It should not be necessary to remove this piece unless a component was to fail. Joins between the sections will be made to be a light push fit and to ensure they do not separate a pin will be slid through to constrain the sections. The completed and assembled model can be seen in Figure 4.14.

4.4.3 Model Pitching Support

Of the two alternatives considered the tether-based system is going to be utilised. It is selected because it is thought that the disturbances from the potentially 3 mm shaft in the yoke based system will be too large. A few factors need to be considered in the design of this tether system, mainly material for the tether, attachment of the tether to the model and test section, and holding the model at its initial AoA.

Tether Selection

The forces experienced by the model for some typical manoeuvres are tabulated in Table 3.5. Although these maximum values of forces do not directly relate to the forces a tether will experience they do give an indication to the magnitude of forces that should be designed for, with some factor of safety placed on top. Typically the values are a maximum on the body normal force coefficient which is the force attempting to correct the models AoA to a 0 degree position, although this value is partly balanced by wing lift it will be considered a maximum. Doubling this value for safety gives a force of 32 N for the 15 degree AoA case.

The wire material selected is guitar string, also known as piano wire, or music wire. The actual material used for manufacture of these strings is typically an ASTM A228 cold drawn carbon steel wire (MatWeb, 2014). The material has exceptionally high tensile strengths. The diameters that are being considered are two commonly available sizes; a 0.008 inch (0.2 mm) and 0.012 inch (0.3 mm) wire with corresponding tensile strength ranges from 2750 – 3040 MPa and 2600 – 2880 MPa respectively. Using the lowest of these values and assuming all load is carried by a single piece of wire we see that the force required to break it \( F_{\text{max}} \) is:

\[
F_{\text{max}} = \sigma_{\text{max}} A_{\text{wire}} = 2600 \left( \frac{0.2^2 \pi}{4} \right) = 81 \text{ N}
\]

This equates to an additional factor of safety of 2.5 on the 32 N load assuming worst-case tensile values and worst case loading condition. So the 0.2 mm wire will be selected, as it is suitably strong and should cause the least flow disturbance.
**Pivot Location**

Position of the pivot point needs to be in front of the centre of pressure for the model to be stable. The centre of pressure for the model sits furthest forward at low AoA however the data fluctuates a lot at these low angles so a nominal value of about 40% body length or 101 mm from the bodies nose has been assumed. The pivot will therefore be positioned further forwards at 96 mm, this is a clear area between the batteries which will allow the two tethers to be fed through (See Figure 4.13). It was decided that the tethers will be placed in an ‘X’ configuration, with both tethers attaching to the nozzle outlet.

**Support System**

Finally because the model is supported only by a pivot it will rotate during setup if not restrained, of course any restraint utilised must no longer support the model once the flow begins. A tether type support will be utilised, and will serve two purposes; the first is it will position the model at an initial angle of attack and hold it there until the flow is established. Secondly it will serve as a method to trigger the actuation system so that fin movement will begin only once the flow is established.

![Support stand features](image-url)
The general setup will have a wire passing out the rear of the model, one end is attached internally to the microswitch trigger, and the other end will be tied to a thin cotton string. This cotton string is then attached to a sliding rod assembly. The rod will have a small disc facing the flow to create high drag. The sleeve, which it slides through, is then mounted to a frame, which in turn is anchored to the test section. This support stand assembly is shown in Figure 4.15.

The cotton string was sized to be sufficiently strong to support the models weight whilst still being weak enough to break. A single strand of cotton string was chosen, and through testing it was found the breaking load ranged from about 5 to 10 Newtons. The drag disc needs to be suitably sized so that it will have sufficient force to break the string and sufficient velocity for it to occur quickly. Assuming a disc of 30 mm x 30 mm square gives an area of 900 mm$^2$, we can then calculate the force it will experience. This will be equal to the static pressure of the test section times the disc area. From the typical flow parameters in Table 3.4 it is known that static pressure during a run is about 30 kPa. This gives a force of

$$F = 0.03 \times 900 = 27 \text{ N}$$

Correspondingly assuming the rod and disc weigh about 50 grams, and the resulting force will be 27 N minus the 10 N required in breaking the string. This gives acceleration equal to 17 N divided by 0.05 kg or 340 m/s$^2$. Finally assuming constant acceleration the time taken for the rod to slide back enough to break the string can be found. It needs to slide about 50 mm.

$$s = \frac{1}{2} at^2$$

Where $s$ is position, $a$ is the acceleration, and $t$ is time.

For 50 mm of movement it’s found that the time taken is

$$t = \frac{2s}{\sqrt{a}} = \frac{2(0.05)}{\sqrt{340}} = 0.017 \text{ s}$$

So it will take at most about 17 ms for the tether to break and therefore the 30 mm x 30 mm drag disc is acceptable.

4.5 Experimental Equipment

4.5.1 TUSQ Facility

The TUSQ wind tunnel is a Ludwieg tube facility with free piston compression heating; it is installed at the University of Southern Queensland in Toowoomba. This facility is unique when compared to other facilities in Australia as it provides comparatively long test flow durations, which allow the study of
unsteady conditions such as vehicle dynamic stability and scramjet inlet startability (Buttsworth, 2009). Typical test times are in the order of about 200 ms. A schematic of the facility is seen in Figure 4.16.

![Figure 4.16: Schematic of the TUSQ Ludweig tube facility (Buttsworth, 2009)](image)

The facility can be operated in a number of different configurations however to produce the nominally Mach 6 flow the facility is operated with the Ludweig tube using free piston compression of the air. In this mode of operation air is pressurised in the reservoir and the test section is evacuated. A high speed, high flow rate actuated ball valve opens to release the high pressure air, at this point the piston is pushed towards the nozzle pressurising the test gas that lays ahead of it. When this pressure is sufficient it will rupture a diaphragm and allows the test gas to travel through the Mach 6 nozzle and into the test section, creating the test flow. The nozzle seen in Figure 4.17 has a throat diameter of 28.8 mm, an exit diameter of 217.5 mm, and a length of 1057 mm.

![Figure 4.17: TUSQ Mach 6 nozzle sketch (Buttsworth, 2009)](image)

4.5.2 Image Capture

To capture data a high-speed camera is used. It is positioned to capture images from the side of the test section. This allows images to be captured in 2 axes, displacement in the direction of the flow and vertical changes.

The camera used is an Olympus i-Speed 3 colour with high-speed digital video (HSDV). In addition because the frame rates used are quite high the model requires extra lighting to compensate for the decreased exposure time of the camera. A halogen work light was also aimed at the model to provide the clearest possible image with highest useable frame rate. Further to this a small reflective strip was placed on the edge of the models control fin, which the camera would be viewing. This makes the fin stand out on the body and will assist with analysis of its motion in the post processing phase.
4.6 Experimental Procedure

4.6.1 TUSQ Facility Operation

Experienced staff members undertook operation of the facility. It is not necessary to reproduce this procedure in full here. Instead the general procedure of operation is summarised:

- Create vacuum on dump tanks
- Open Ludweig tube at nozzle end and remove piston and diaphragm from previous run
- Clean Ludweig tube with cleaning slug
- Fit new diaphragm and close tube at nozzle end
- Open primary valve end of tube and place piston inside
- Close test section after model setup is complete
- Pull vacuum on test section
- Prepare air for piston compression
- Release flow

It is worth noting here that on application of vacuum to the test section the nozzle and Ludweig tube are pulled forwards, into the test section, until they contact some mechanical stops. If the model is setup before these stops are contacted the application of vacuum will cause the nozzle to shift and the tethers to go slack. Therefore these stops should be engaged before the models initial positioning.

4.6.2 Model Preparation & Assembly

Before preparing for a test the model must be loaded with suitable actuation profiles. This is done via the Arduino programing interface where the original microcontroller script was written. It has been setup so that 3 separate actuation profiles can be programed. The profiles are controlled by varying the step time, step number, and delay before the fin changes direction. It is known that each step of the motor is 18 degrees, this in turn will rotate the gearbox a fraction of this amount depending on the gear ratio. For a 1:10 gearbox this will be 1.8 degrees per step, therefore the number of steps required is the required actuation divided by the gearbox step size of 1.8 degrees. For numbers that are not whole steps they are rounded to the nearest integer, this assumption is corrected for in the results analysis. The final code used for all testing is included at the end of Appendix F. The actuation profiles shown there are samples however they were varied as required.

At the beginning of a test the models electronics must be switched on. Separating the nose from the body will reveal the battery leads and controller leads; these are then connected together whilst respecting correct polarity (Figure 4.18). Once connected the microcontroller is active.
The model actuation profile is then selected by use of the triggering switch. At first power on a single pull of the switch goes to a programing state for 5 seconds. If the switch is not pressed during this time it goes to mode 1, if it is pressed once it goes to mode 2, if it is pressed twice it goes to mode 3. After the mode is selected and the 5-second programming period is over it will run a demonstration of the actuation type selected to allow visual confirmation of the correct setting. Following this the model is ready for the switch to be pressed and held closed by its support tether, upon release it will begin actuation. Finally at the conclusion of the actuation it will flash an LED to confirm successful operation. Pressing the switch again whilst the LED is flashing resets the controller back to its initial power up state.

4.6.3 Model Setup in Test Section
There are two aspects to setting the model up in the test section, they are fitting the pivot tethers to the test section nozzle, and fitting & attaching the support system.

**Pivot Tethers**
The steel tether wires attach in an ‘X’ configuration onto the nozzle face via four M6 bolts (Figure 4.19). One end of each wire is firmly clamped under a bolt; it is then fed through the pivot point on the model.
Wire length is then adjusted to set the models vertical and horizontal position, when the desired position is achieved the other end of the wire is clamped under the second bolt. It was set so that horizontally about the first 20 mm of the nose protrudes into the nozzle. Vertically it should be set slightly above centre for a nose up manoeuvre or slightly below vertical for a nose down manoeuvre. This will maximise the available space in the Mach cone.

**Attaching Support System**

The support system consists of a stand mounted to the test section and a drag disc on a sliding rod (Figure 4.20, Figure 4.15 & Appendix G for detail drawings). The stand is bolted to the test section mounting plate via 2 M8 bolts and has adjustable slots used to fine-tune the models initial angle. The stand is first slid as far forwards as possible, the string is then tied through the wire connected to the microswitch, the other end of the string is then tied to the drag disc and sliding rod assembly. Finally the stand can be slid backwards in the slots until the model is close to the desired initial angle of attack.

4.6.4 Tunnel Flow Data

During each run various flow parameters for the tunnel operation are recorded. This data can later be utilised in theoretical calculations when modelling expected behaviour. The flow parameters will not vary greatly between runs and so for the design work undertaken in previous sections nominal parameters from a typical run have been utilised.

4.6.5 High Speed Camera Data

The high-speed camera is constantly recording a set length of video, nominally about 3 to 20 seconds depending on the frame rate selected. From when the stop record button is pressed the previous 3 to 20 seconds will be kept. For this reason one person triggers the tunnel whilst another presses the stop button after the flow finishes.
After this the data is saved locally on the camera, it can then be uploaded to a storage drive and then onto a computer. On the computer it is saved as a series of TIFF images for later analysis.

4.7 Data & Post Processing Required
Following the conclusion of a run the raw image data requires certain post processing to convert it to a form suitable for analysis. After obtaining the images from the camera they are a series of TIFF images, in total there are a couple thousand images depending on the frame rate selected. To cut down the size of this image set, images before the flow start, and after the flow conclusion are removed. This typically leaves only a few hundred photos.

These remaining photos are then loaded into basic photo editing software. The various levels of the photo are then adjusted to improve the image quality as much as possible, with the goal being to improve clarity of the model and fin surface.

4.7.1 Creation of Video
The first piece of information to be made from these photos is a short video. The photos are loaded into a basic movie-making program where they can be compiled. Basic selections such as time per frame can be made, this value is set so that the video plays quickly enough that the overall motion can be best understood but still slowly enough that any small details can be picked up on and investigated further. The videos are very useful as they allow a holistic view of how the particular run went, something which still images do not provide.

4.7.2 Interpreting Position from Image
To determine the models AoA and the control surface deflection the images are loaded into AutoCAD as a background image. Following this the image is scaled so that the drawing of the model lines up with the image of the model.

![Figure 4.21: Sample image demonstrating determination of model angles](image-url)
An outline of the model and its control surface is placed onto the image and then rotated and positioned so that the two line up. After this it is possible to measure the control surface deflection angle relative to the body and the models AoA relative to some reference position. The reference position is a point before the test flow begins, where it is known the body is at zero degrees AoA. A sample image of this process is provided in Figure 4.21.

The starting time for the manoeuvre is assumed to be when the control surface is first seen to move. Not every image is looked at, instead only every second image is viewed to reduce processing time. Finally the time for each frame is simply determined from the inverse of the frame rate. For example if a frame rate of 1000 fps is used this corresponds to 0.001 ms per frame. This data is then all recorded in a table and included in Appendix H for each run.

This approach obviously has some limitations, which will affect the accuracy. Determining the models position is up to the researches discretion as to whether the profile on the image lines up sufficiently well. In addition it can be quite difficult to detect the wing even with a reflective marker on its edge. Also the reference line for a zero degrees AoA is only as accurate as the models original position.

4.7.3 Implementing into Matlab

The data is entered into a Matlab function file as an array. The first column contains time points, the second column contains AoA data, and the third column contains control surface deflection. Positive AoA means a nose up movement, and negative AoA means a nose down movement. Positive control surface deflection is downwards relative to the body and would tend to cause a positive nose up manoeuvre whilst negative values are the opposite. The file is then named in the format ‘RUNXXX’ where ‘XXX’ is the run number.

The data is used as is to provide AoA information, and control surface deflection information. It can then also be used to give angular acceleration data. Angular acceleration is simply the second derivative of the AoA with respect to time. Before taking the derivative of this raw data it must first be filtered by fitting the data with a function. If this is not performed then the noise in the data will be amplified after taking the derivative. A polynomial is selected to fit to the data; its order is adjusted depending on the data set.

Upon running the script there will be a prompt to plot experimental data, if ‘y’ is selected then the file name for the data will need to be entered. The script will then provide a plot of the observed and expected AoA data and control surface deflection on the same plot for comparison. It also provides a plot with the observed angular acceleration and expected angular acceleration.
Chapter 5 – Results and Discussion

5.1 Chapter Outline
This chapter presents results and analysis of the testing undertaken in the TUSQ facility. It will provide a brief overview of each run configuration for reference. Following this each runs processed data is presented, described, and analysed with a full discussion. Runs will then be compared and discussed together to develop an opinion on the validity of testing hypersonic vehicles with actuated control surfaces. The limitations of these results and the previous design work undertaken are then going to be considered followed by recommendations for further work that could be undertaken.

5.2 Overview of All Runs
Table 5.1 provides details of each run, including pre run setup for the model, model recorded mass parameters, and tunnel recorded flow conditions. The required tail fin actuation is selected as an integer number of steps available from the stepper motor and gearbox configuration. For example if the stepper is fitted with a 1:10 ratio gearbox one step of the motor means 1.8 degrees of fin movement, therefore the available actuation amounts will be multiples of 1.8. The pitch then corresponding to this tail fin actuation is found using the Matlab script.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>RUN283*</th>
<th>RUN286*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fin actuation, $\beta$</td>
<td>12.6 degrees</td>
<td>7.8 degrees</td>
</tr>
<tr>
<td>Theoretical pitch, $\alpha$</td>
<td>11.2 degrees</td>
<td>7.46 degrees</td>
</tr>
<tr>
<td>Gearbox ratio used</td>
<td>1:10</td>
<td>1:30</td>
</tr>
<tr>
<td>Motor step rate</td>
<td>10 ms/step</td>
<td>7.5 ms/step</td>
</tr>
<tr>
<td>Fin actuation rate, $\dot{\beta}$</td>
<td>180 deg/s</td>
<td>80 deg/s</td>
</tr>
<tr>
<td>Hold time at full actuation</td>
<td>10 ms</td>
<td>10 ms</td>
</tr>
<tr>
<td>Model c.g., $x_{c,g}$</td>
<td>159.1 mm</td>
<td>150.5 mm</td>
</tr>
<tr>
<td>Model mass, $m^\wedge$</td>
<td>119.8 g</td>
<td>123.0 g</td>
</tr>
<tr>
<td>Inertia, $I_{pivot}^\wedge$</td>
<td>1.049e3 kg.mm^2</td>
<td>1.057e3 kg.mm^2</td>
</tr>
<tr>
<td>Mach number, $M_\infty$</td>
<td>5.84</td>
<td>5.84</td>
</tr>
<tr>
<td>Stagnation press., $P_0$</td>
<td>0.92 MPa</td>
<td>0.92 MPa</td>
</tr>
<tr>
<td>Stagnation temp., $T_0$</td>
<td>572 K</td>
<td>572 K</td>
</tr>
<tr>
<td>Tether Wire Diameter</td>
<td>0.2 mm (0.008&quot;)</td>
<td>0.4 mm (0.016&quot;)</td>
</tr>
<tr>
<td>Camera frame rate</td>
<td>1500 fps</td>
<td>3000 fps</td>
</tr>
<tr>
<td>No. of frames kept</td>
<td>600</td>
<td>3328</td>
</tr>
</tbody>
</table>

* mass, c.g. location, freestream Mach number, stagnation pressure, and stagnation temperature for this run are estimates only, no measured values were recorded.
^ Values are estimates only based on solid modelling.

The gearbox ratio, motor step rate, fin actuation rate, and hold duration when the fin is at full actuation, are all used to run the Matlab script when producing
theoretical data for comparison with experimental results. Finally some measured parameters for each particular run are given, these are: the models centre of gravity location, model mass. For the runs undertaken freestream Mach number, stagnation pressure, and stagnation temperature values were not kept, as the runs were not completely successful. Instead the nominal values are used as they will be sufficient for this analysis. The values recorded in the table are also used to run the Matlab script for production of theoretical results. All remaining geometrical parameters were constant between runs and can be interpreted from the detail drawings in Appendix G or alternatively from Table 3.1.

The data that is entered in the above table and the fixed geometric data are entered into the Matlab script when producing theoretical and experimental data comparisons for each run. By matching as many parameters as possible it is thought that the theoretical calculations will maintain greater accuracy and be more reliable for comparison with experimental data.

5.3 Run 283
Run 283 was the first run undertaken and so some preliminary tests were performed with the model set in the test section to confirm it would operate as intended. The test was to manually trigger the support system by pulling the drag rod to break the cotton string. This was then recorded with the high-speed camera to confirm the fin operated as expected, and to ensure appropriate frame rates and exposure were set. No recorded footage was kept however it appeared as if all systems were functioning as expected and so the run proceeded.

5.3.1 Run 283 Results
The run itself had some promising and also unfortunate results. Because it is impractical to provide all frames from a run in the content of this report a sequence of frames from the flow start to the end of the run are provided in Figure 5.1. For an indication of scale every 10 frames represents 6.67 ms of real time. Frame 190 is when the flow is just entering the test section; between this frame and frame 200 a successful breaking of the cotton string occurs. From frame 200 onwards the model is now supported by the flow and the front tethers only.

At about frame 220 the first noticeable fin movement occurs and by frame 240 the fin has stopped movement and the model has not yet begun pitching upwards. However from frame 250 to 270 the model is pitching upwards but by the time it reaches its maximum pitch at frame 270 the fins have returned to being inline with the body axis. In frame 280 it can be seen that the fins have been sharply caught by the flow and pulled at a large drag inducing angle to the flow. The body pitches rapidly to a nose down position by frame 290 and in frame 300 the rear body section actually separates from its nose.
The nose is now unstable without the body attached, as the centre of pressure location will have shifted forwards in front of the models pivot. This causes the nose to rotate 90 degrees by frame 310 and then the excessive drag loading causes the front upper tether to break and then the lower tether to break by frame 320. The nose is then seen drifting down the test section in the final frame.

Figure 5.1: Image sequence from RUN283
Although the whole of run 283 was not successful there is still much information to be gained from it. Most importantly it was observed that the model did indeed pitch as intended during the initial part of the run. For this reason the AoA and tail fin angle were determined for every second frame starting at frame 215 up until frame 273. This series of frames are the period from which the tail fin began actuating up until just prior to the model losing control, overall it represents about 40 ms of data. The raw data for these recorded values is included in Appendix H.

Plotting this observed AoA and tail fin angle data obtained against the theoretical results determined using the parameters of Table 5.1 gives the graph in Figure 5.2.

![Body & Fin Angles for a 11.2 degree AoAManeuvre, and RUN283 data](image)

**Figure 5.2: Expected & observed body & fin angles for RUN283**

At first inspection of the graph in Figure 5.2 it appears as if there is an acceptable correlation between the expected and observed AoA. However the expected and observed tail fin actuations differ greatly meaning that the assumed fin actuation is incorrect and as a result this is going to affect the theoretically determined AoA. For this reason the theoretical AoA is recalculated using the actual tail fin actuation profile of run 283. The figure is then replotted showing only the first 40 ms of data, as the fin actuation is not known for the remainder of the simulation. This gives the result in Figure 5.3.
Figure 5.3: Expected AoA corrected to use observed tail fin position of RUN283

Figure 5.4: Expected and observed angular acceleration for RUN283
Similarly it is desirable to compare theoretical and experimental model angular acceleration. Determination of this by fitting the models AoA data with a polynomial is covered in Section 4.7. The polynomial used was fourth order and so after taking the second derivative the result will be a quadratic polynomial. Again looking at only the first 40 ms of the run, Figure 5.4 is obtained.

5.3.2 Run 283 Discussion
A number of important results and design improvements can be identified from run 283. The first is a successful release of the support system. This occurred at about frame 195, relative to the flow start this only took 5 frames to occur or at 0.66 ms per frame at total of 3.3 ms. This occurred sufficiently fast and therefore release of the model uses only a minimum of test flow duration. In addition it appears as if the shock loading on the cotton string when it is being broken has caused minimal disturbance to the models AoA, although in Figure 5.3 between time 0 to 18 ms the model is at a negative, nose down AoA with a maximum of about -2 degrees. It is believed that the support system has caused this affect.

Following this it appears as if the fin had begun actuating as expected however upon comparison with the theoretical results of Figure 5.2 it can be seen that the observed fin rate does not even closely align with the expected rate. Because the actuation rate is controlled by a microcontroller the only way it can step either faster or slower than the expected amount is if the load on the motor is exceeding its torque capabilities. It is therefore believed that what is actually occurring is a combination of some control from the motor and some influence from the aerodynamic wing loads on the motor. The torque capabilities of the motor and requirements for control were a concern earlier on in the design phase and it was thought that a 1:10 ratio gearbox would be sufficient to overcome these problems, this was not however the case.

Although the motor did not function as expected it is still possible to use the data obtained to verify the applicability of the theoretical calculations that have been developed. Using the observed fin actuation rate to determine a theoretical model AoA response and comparing with the observed AoA response gave Figure 5.3. This showed very good correlation between the models expected and observed AoA for the first 40 ms of useful data. The two data sets follow the same slope for the duration however the theoretical data is about 3 degrees greater than the observed results for the duration. Two factors have likely influenced this difference; the first is the initial negative AoA at the beginning of the run caused by the support system as it breaks the cotton string to release the model. The initial negative AoA immediately offsets it from the theoretical data and so it will lag with this constant amount. The other contributing factor is the exclusion of damping from the analysis; unfortunately for this run there is insufficient data to make a suitable estimate of the damping present. Even with these possible errors the results here are quite good when compared to other research that has compared theoretical and experimental
results for even much more simple body geometries than this rocket configuration.

Because the slope of the expected and observed AoA graphs is almost identical, the angular acceleration is expected to have a good correlation. Figure 5.4 confirms this and shows an excellent correlation between the two.

The design improvements are identified just after this successful pitching up manoeuvre. After this point the insufficient torque of the motors becomes a large problem as the control fin is sharply rotated backwards. Following this excessive drag produces sufficient force to separate the body from the tethered nose. Originally it was intended that these two sections would be constrained by a small pin however the fit between the two components was so firm that it was thought this would not be required, obviously it should have been added.

Following this the nose lost control, when it became 90 degrees to the flow the excessive drag caused the tethers to break. The upper tether broke first and failed on the stress raising edge as it went through the pivot sleeve on the model. The second tether failed where it was bolted to the test section flow nozzle.

It is strongly believed that had the wing control not failed, the body would not have separated from the nose, the tethers would not have been broken, and overall the test would have been more successful. Regardless this run has made apparent some necessary design improvements for future runs.

5.4 Run 286
Run 286 used the insight gained from run 283, the initial run, to make certain critical design improvements. Because the last run showed a lack of torque available from the motor the gearbox used was changed from a 1:10 ratio to a 1:30 ratio with the intention of tripling torque output. Because this now results in slower actuation times the step time was reduced slightly from 10 ms/step to 7.5 ms/step. This is achieved by shortening the length of the LOW pulse signal created by the microcontroller from 5 ms to 2.5 ms. The HIGH signal time was left at 5 ms as this signal causes the motor to move and shortening it can cause the motor to miss steps (See Appendix F for a full discussion on this) whereas the LOW signal is just a pause between HIGH signals and only needs to be long enough for the stepper driver to recognise it.

Other more minor changes included the increase of tether wire diameter from 0.2 mm (0.008”) to 0.4 mm (0.016”) to prevent failure in the event of unexpected loadings. The body was more rigidly connected to the nose by use of retaining pins to prevent separation of the two components. Finally the wings and bevel gear drive, which were originally a press fit onto their shafts, were held in place with superglue to try and prevent undesired movement.
5.4.1 Run 286 Results
Run 286 showed some improvements over run 283 however it was still not able to demonstrate a successful manoeuvre. It does however provide results, which can be compared to run 283 for the purpose of drawing conclusions. Again to give an overview for the run a sequence of frames is provided in Figure 5.5 and Figure 5.6. Every 20 frames in the sequence is 6.67 ms in real time. Frame 370 is when the flow just enters the test section; this is followed by the tether breaking in frame 386. After this point onwards the model is support only by its tethers and the flow.

Figure 5.5: Image sequence from RUN286 part 1
Following release the body begins to pitch nose up until frame 490 and during this time the first noticeable fin movement occurs at frame 462 however it moves opposite to the expected direction. From frame 490 until 570 the model then starts to pitch nose down. During this period the fin also begins to start actuating in the correct direction at frame 500. At frame 598 the fin stops actuating and holds steady. After frame 570 the body starts to pitch in the positive nose up direction and continues to do so until frame 650.
Between frame 630 and 650 the fin is rapidly rotated in the negative direction causing a rapid change to model AoA resulting in a large nose down pitch between frames 670 to 830. Now out of control the model starts to pitch nose up again however at around frame 930 one of the control fins breaks away, allowing the other control fin to rotate and bend the pivot shaft. The shaft is actually bent inside the model and as a result it pushes the stepper motor and gearbox assembly backwards causing the motor carrier to be pushed out.

Similarly to run 283 this run was not a complete success however data recorded can still be analysed. The run is deemed to have begun at the point where the tether breaks, this is at frame 386. From this point up to loss of control at frame 650 the images will be analysed to determine AoA and control fin angles. It was deemed sufficient to investigate every fourth frame from this set of images; this is equivalent to a time step of 1.33 ms (each frame is 0.33 ms). The raw tabulated data from completing this process can be found in Appendix H.

The first graph produced in Figure 5.7 is a plot of the expected AoA and tail fin actuation with the observed AoA and tail fin actuation shown for comparison. Similarly to run 283 the correlation of expected and observed tail fin actuation amounts is not great. To attempt correction to this the theoretical analysis is run again using the observed tail fin position to determine an expected AoA.

Figure 5.7: Expected & observed body & fin angles for RUN286
The plot in Figure 5.8 shows this revised graph and the correlation between the results.

**Figure 5.8:** Expected AoA corrected to use observed tail fin position of RUN286

**Figure 5.9:** Expected & observed angular acceleration for RUN286
Finally Figure 5.9 shows the comparison of theoretical and experimental model angular acceleration. Determination of this by fitting the models AoA data with a polynomial is covered in Section 4.7. The polynomial used was fifth order and so after taking the second derivative the result will be a cubic polynomial.

5.4.2 Run 286 Discussion
Some unique and difficult to explain results were found through examination of the results of run 286, the analysis presented here will attempt to justify them. To begin at the start of the run there is a successful, steady release of the support system at frame 386. When compared to the point where the flow entered the test section at frame 370 a total of 16 frames or 5.3 ms has passed. This is quicker than the time determined during design and requires only minimal flow duration for execution. It also appears as if the disturbance to the model caused by the breaking of the cotton string is minimal. This is shown by no perceptible change in AoA in Figure 5.7 or Figure 5.8.

Following release the body begins to pitch nose up due to is mass and centre of gravity location being behind the pivot. No visible fin motion occurs until 30 ms after release (Figure 5.7) where a fin motion opposite to that expected occurs. The size of this reverse fin motion peaks at about -2 degrees and it is thought that this is a result of the tolerance between the internal bevel gear drive. Because of difficulties in setting the two bevel gears to mesh properly there remains about a 3 to 5 degree backlash in the fins. This would mean as the body pitches without a firm mesh between bevel gears the fins can sharply deflect backwards until this backlash is taken up. Following this the fin starts actuating in the expected direction with an essentially linear profile. The actual rate of this actuation is comparable to the expected rate in Figure 5.7 however the fin still shows minimal control. At 70 ms the fin begins to hold its position steady at 7 degrees for 10 ms duration. This aligns with the amount of time the fins were programmed to remain in a fixed position for however it occurs 25 ms to early. It is unclear why this pause would occur early unless the internal clock on the microcontroller is not very accurate.

After this sharp fin deflection the model pitches nose down but with some delay after. It takes about 10 ms for the body to respond to the fin motion. In addition with reference to Figure 5.8 the magnitude of the nose down pitching which results is seen to be a lot larger than what the theoretical analysis using observed tail fin actuation predicts. The theoretical calculations indicate that the negative fin actuation would not have caused a negative AoA. A possible cause of this problem is thought to be with the tether system. As the model makes these rapid direction changes one of the tethers is observed to go slack. Whilst a tether is slack the model is almost in a free flight and therefore during this period the pivot location is not providing a completely fixed point to pivot about. This means the model will no longer behave as expected, as it will tend to pivot about its centre of gravity. This will give results, which the theoretical analysis cannot predict.
After this the body is pitching in the positive direction corresponding to the fin input. The body reaches a pitch of nearly 20 degrees, far greater than the prediction of Figure 5.8, which indicates an approximate 12 degree pitch. It is possible that the body exited the testing Mach cone; if this were the case the body would be exposed to a lower pressure than that present in the flow causing unpredictable motion. This could also explain how shortly after this large pitch angle the body rapidly lost control. Although it is not possible to see at what point the model exited the Mach cone it is known from the design process that at a pitch of 15 degrees the control fins will no longer be within the cone.

Figure 5.9 shows the expected angular acceleration based on the observed fin actuation against the angular acceleration determined from AoA data for the run. Because the rates of change of the observed AoA are vastly different from the predicted values it is not expected that the theoretical and experimental angular accelerations would correlate well, and they do not. As there appears to be many external factors not accounted for by the simulation this is included mainly for interest.

Most of the improvements made as a result of run 283 appear to have been effective. However it still appears as if motor torque is a problem with the issue worsened due to bevel gear backlash and possible slipping of bevel gears on their shafts. The new problem identified is a setup and testing limitation of the Mach cone. Exiting this Mach cone or approaching its boundary will not only invalidate test data but also cause loss of control of the model and potentially damage to the model.

5.5 Validity of Testing Models with Actuated Control Surfaces

There are various aspects, which can be discussed to decide whether or not testing of models with actuated control surfaces is a valid technique. It is clear that a complete controlled manoeuvre was not demonstrated in this work however sufficient results have been gathered to come to a decision. To begin, the two runs completed will be compared to find those portions that were successful, and those that remain problems. The results produced will then be compared to theoretical predictions of the response and finally the challenges faced through creation of a suitable model will be presented.

5.5.1 Comparison of Runs

The support system, which holds the model at its initial AoA, releases the model at flow initiation, and triggers the actuation system to start, worked effectively in both runs. Trigger time was about 3.3 ms and 5.3 ms for the two runs, which represents only a small portion of total flow duration. The system is relatively easy to setup and provides a smooth model release. Disturbance to the models initial AoA was observed to cause a nose down pitching of -2 degrees in run 283 and in run 286 no disturbance was observed.
The lack of disturbance in the run 286 caused by the release system is attributed to the change in tether wire diameter. It could be seen that when using the thinner 0.2 mm diameter wire in run 283 that the tethers were fairly straight. When increasing this to 0.4 mm diameter wire in run 286 the wires had some permanent curve and spring to them. This extra spring in the wire has a shock absorbing effect on the release system forces, when the system pulls the model to break the cotton string this force is dissipated into the wires and thereby causing a less noticeable nose down pitching. The other affect the thicker wires had was minimisation of body roll. In run 283 the body was seen to roll early on in the test, likely due to fin misalignment. When the thicker wires were used in run 286 the body still tried to roll however the magnitude of the rolling was much smaller as the thicker wire provided a much stiffer restraint.

Run 283 implemented a 1:10 ratio gearbox attached to the stepper motor. The results of this run showed clearly that the torque provided was insufficient and therefore resulted in the control fins loss of control. Run 286 tried to improve on this by using a 1:30 ratio gearbox, the results seen when comparing Figure 5.2 and Figure 5.7 show an improvement in control with the higher torque gearbox. There still however appears to be some skipping of motor steps or slipping of components leading to unexpected fin actuation.

5.5.2 Correlation of Experimental & Theoretical Results
Both runs undertaken show not great correlation between the theoretical prediction and the test results. This is mainly a result of the observed fin actuation not aligning with what was expected. When modifying the results to use the observed fin actuation in determination of the theoretical AoA the correlation was vastly improved for run 283 (See Figure 5.3). When the same treatment was given to run 286 data the correlation was slightly improved however the data overall did not correlate well (See Figure 5.8). The lack of correlation improvement when using observed fin actuation was likely caused by the model control fins exiting the testing Mach cone. This would result in unpredictable pitching motions. Run 286 was more prone to leaving the Mach cone than run 283 as the model was positioned further downstream of the nozzle. This means a reduction in the allowable pitching range would occur.

If the effects and limits of actual versus predicted control fin position and available space in the Mach cone are adhered to the experimental data obtained is going to be valid. Some other potential factors, which may affect the accuracy of resulting data, are disturbances caused by the tethers, friction caused by the tethers in rotation, and potential nose down pitching cause by the release system.

5.5.3 Test Model Development
During this research the model to be investigated was a simple missile or rocket shaped body, it was selected because of its desirable shape. The shape made it quite straightforward to have a long cylindrical cavity inside to mount all testing equipment. This is of course not always going to be the case as many
hypersonic vehicles have very flat plate type geometries. Combining this factor with the allowable scales suitable for hypersonic testing facilities is going to result in difficulties in implementing sufficiently small drive systems to operate the control surfaces. The stepper motor used during this research was the largest available which could fit inside the model cross-section and this still provided insufficient torque. Similar difficulties will be faced when sourcing other componentry such as batteries, microcontrollers and motor drivers. These components are however, typically available in a wide variety of geometries.

The other issue faced is involved in the assembly of the model itself. To achieve an accurate surface which aerodynamically represents the full-scale vehicle and provides an internal cavity for mounting of actuation system components necessitates the use of 3D printing. There is significant time placed into correctly modelling these internal cavities so that the components will fit. Once printed the parts then required hand finishing to open holes out to correct diameters, allow fitment between push fit components, and then final assembly of the components. Overall if the technique were used to design a hypersonic vehicle it would only be undertaken for a small number of final designs due to the large costs and assembly times associated with it.

5.6 Experimental Limitations
Through completion of the experimental runs there has been limitations that ultimately affect recording of resulting data, or have the potential to introduce error into the results.

5.6.1 Data Capture
Data capture is purely via high-speed camera and this is converted to a tabulated form via a manual process of tracing the outline of the body and control fin to determine angles. The process takes quite long to complete and the accuracy of how the outlines are placed is up to the discretion of the researcher. In addition the tether release times, and fin actuation start times are all estimated based upon the high-speed images. These factors all introduce errors into the graphed data.

5.6.2 Data Type
It is difficult to determine the models yaw purely from a side view and if the body was undergoing any significant yawing motion this would affect results. In addition with no flow visualisation utilised it is not possible to see the flow disturbance effects that the tether system has on the body.

5.6.3 Mach Cone
As has been noted a few times previously the Mach cone size limit for the facility places a large constraint on the test. Leaving the Mach cone introduces error into the data and can cause loss of model control. Remaining inside the Mach cone means a model of suitably small scale must be selected so that it can still manoeuvre in the available space. This sets a constraint on the size of actuation equipment that is to be mounted inside the model. Depending on the
scale selected the AoA range will also be limited as when the model pitches its aft, and therefore the control fins, tend to exit the Mach cone.

5.6.4 Number of Runs
During this work it was intended that at least 3 experimental runs would be undertaken. Unfortunately due to time constraints and model repair setbacks it was only possible to complete two runs in the hypersonic test facility. This does not provide a great amount of data for comparison and confirmation of observations. In addition it was hoped that at least one test could demonstrate an entire successful manoeuvre as this would provide strong evidence that the testing technique is possible.
Chapter 6 – Conclusions

6.1 Research Aims
The project intended to design a model with actuated control surfaces for testing dynamic manoeuvres in a hypersonic wind tunnel. To achieve this it needed to predict a theoretical response based on control surface input. It then needed to design a control and motor package to operate the control surfaces, which needed to be suited to the models scale. Finally fabrication followed by testing of the model in the TUSQ facility would provide experimental results to compare with theoretical calculations. Altogether if successful this would validate the testing of models using actuated control surfaces and could be applied to various other hypersonic vehicle designs of interest.

6.2 Summary of Findings
Theoretical analysis of the bodies response was achieved by first selecting a body and wing profile. Through the use of some hypersonic aerodynamic theory and tabulated coefficient data it was possible to determine a model response based on control surface inputs. With implementation of a Matlab script, plots of AoA, and angular acceleration as functions of time were created.

The actuation system successfully incorporated a microcontroller, motor driver, stepper motor, bevel gear drive, system trigger, and a battery power supply. The microcontroller in conjunction with the motor driver allowed programming of many different actuation modes; it also allowed variation of actuation speeds and delays. During testing this package was reliable and effective. The battery power supply was sufficient to complete multiple runs without recharging. The stepper motor proved to provide insufficient torque for the operating conditions and was one of the main causes of issues in testing. Fabrication of the model and assembly of these components proves to be a very tedious and time consuming process making it not suited for testing a large numbers of designs. Rather it would only be used for testing of designs that have had sufficient development using other techniques prior to using the actuated control surface approach.

The results of testing initially did not show good correlation with the theoretical predictions. This was due to the control fin having incomplete control as a result of insufficient torque supplied by the stepper motor. This issue led to unexpected tail fin actuation that did not align with the actuation assumed in theoretical calculations. Replacing the theoretical actuation with the observed actuation to modify the theoretical results showed a significant improvement of the data. Some error still remained due to the models control fins exiting the Mach cone and causing unpredictable results. Overall the theoretical calculations were deemed valid and had these errors not been present, correlation of data is expected to be much greater.
So it has been shown that it is possible to install an actuation system into a small-scale model suited to the testing cross section of a hypersonic facility. This system does demonstrate some control within the hypersonic test flow however in this case insufficient motor torque results in errors and test failure. The resulting test data that was captured indicates that had the testing errors of control fin malfunction and Mach cone exiting not been present there would be an acceptable correlation of experimental data with theoretical predictions. As a testing technique it is recommended that further work be undertaken to develop it further. With more development and design refinement it is foreseeable that this technique can be used as a tool in the design of hypersonic vehicles.

6.3 Further Work
It is apparent that at this stage successful testing of a model using actuated control surfaces has not been achieved. However future work on the project could make improvements to the existing work undertaken in this research.

To have a successful manoeuvre the main area that should be focused on is the stepper motor and gear system. Other higher quality stepper motors than those used in this research are available at a higher cost. They can be purchased in a variety of diameters with gearboxes attached. These better quality steppers have increased torque outputs and higher step resolutions enabling much better control over the fins. In addition the bevel gearbox tolerances need to be refined to reduce backlash present, or alternatively the drive system could be redesigned to eliminate the bevel gear drive altogether.

Assuming the issues with the stepper motor are solved it would be desirable to improve the data capture system. The graphical approach is okay as a confirmation of the models position however it is not very accurate and the process of post processing the data is tedious. Using an accelerometer to record angular accelerations and a pressure transducer to record model AoA would provide the data far quicker and with a much greater accuracy. To determine fin position a stepper motor with built in encoder could also be used. Note that if the stepper has sufficient torque it would operate as expected and so this feature may not be required.

Following a successful test using the basic geometry investigated in this research the system could be implemented into other real hypersonic bodies of interest. Each body geometry will present with it unique challenges to mount batteries, microcontrollers and motor and drive systems. Further on from this different types of manoeuvres other than the basic pitching manoeuvre could be tested such as roll and yaw.
Reference List


Buttsworth, D 2014a, *TUSQ_nozzle_exit_conds.m*, version 1, Matlab Script, via email, 27 May 2014

Buttsworth, D 2014b, *wedge_wing.m*, version 1, Matlab Script, via email, 12 August 2014


ENG3003 Engineering Management: Study Book 2 2014, University of Southern Queensland, Toowoomba
Ennis, RJ 2013, ScramSpace hypersonic aerodynamic drag coefficient ($C_D$) determined by free-flight experiments at TUSQ wind-tunnel, Bachelor dissertation, University of Southern Queensland, Toowoomba.


Appendix A – Project Specification

University of Southern Queensland

FACULTY OF ENGINEERING AND SURVEYING

ENG 4111/4112 Research Project

PROJECT SPECIFICATION

FOR: Michael John FOWLER

TOPIC: DEVELOPMENT OF A TETHERED MODEL HOUSING AN ACTUATED TAIL FIN TO DEMONSTRATE A MANOUVER IN THE USQ HYPERSONIC WIND TUNNEL.

SUPERVISOR: Professor David Buttsworth

PROJECT AIM: Develop a model to house a stepper motor system capable of actuating a model rocket tail fin. Using this model demonstrate a simple manoeuvre in the USQ hypersonic wind tunnel during the steady run time.

PROGRAMME: Issue C, 24th February 2014

1. Research methods of experimentally performing manoeuvres in a hypersonic wind tunnel.
2. Determine the model’s expected response to tail fin actuation by means of a semi-analytical method for validation of the results obtained experimentally.
3. Design and build a scaled model capable of housing an on-board tail fin actuating system & accelerometer sensor package suitable for use in the hypersonic tunnel. Model design should also consider the potential for a nose mounted altitude sensor to be incorporated at a later date.
4. Perform a series of model manoeuvre tests in the wind tunnel, using a high-speed camera and on-board accelerometer package to show the results.
5. Analyse the results to determine the effect tail fin actuation has on the models flight profile.
6. Submit a dissertation on the undertaken research.

As time permits:

7. Undertake tests with the model using different amounts of tail fin actuation to demonstrate other manoeuvres.
8. Implement a pressure transducer system in the model to record model attitude and further validate other results.
Figure B.1: Geometric details of considered hypersonic bodies (Ashby, 1961)
Figure B.2: Coefficient data for basic body, model I (Ashby, 1961)
Figure B.3: Experimental & fitted data comparison for body normal force coefficients

Figure B.4: Experimental & fitted data comparison for body axial force coefficients
Figure B.5: Experimental & fitted data comparison for body centre of pressure

Figure B.6: Calculated lift & drag coefficients for an 8.6 degree wedge wing
Appendix C – Matlab Scripts

C.1 hypersonic_manoeuvre_simulation.m

% hypersonic_manoeuvre_simulation
% Created by: Michael Fowler
% Student No: 0061019338
% Created: 05/05/2014
% Last Edited: 09/09/2014

% Simulates a simple pitching manoeuvre of a hypersonic missile type
% vehicle fitted with two dependent control surfaces.

clear all
clc

%% Define Constants

global L As Ap x_pivot x_cp_wing I_pivot p_inf M q_inf beta_req W x_cg
delay beta_dot_deg R k

ds = 25.4*10^-3; % Diameter of cylindrical body, m
As = (ds^2*pi)/4; % Cross-sectional area of body, m^2
chord = 0.050; % Chord length of wing, m
width = 0.025; % Wing width, m
wedge_angle = 8.6; % Wing wedge angle, degrees
Ap = 2*(chord*width); % Plan form area of control surfaces (2 fins), m^2
L = 254*10^-3; % Model overall length, m
x_pivot = 96*10^-3; % Pivot point location from model nose, m
x_cp_wing = 222*10^-3; % C.P of wing from model nose, i.e wing pivot, m
x_cg = 150.5*10^-3; % Centre of gravity of Model from nose, m
m = 0.123; % Model mass, kg
q_inf = 287; % Gravitational constant, m/s^2
W = m*g; % Model weight, N
delay = 0.01; % Hold time for fin position in seconds
beta_dot_deg = 80; % Fin actuation rate in degrees per second
M = 5.84; % Flow measured Mach number
p_0 = 0.92e6; % Flow measured stagnation pressure in Pa
T_0 = 560; % Flow stagnation temperature in K
k = 1.4; % Ratio of specific heats (air)
R = 287; % Gas Constant, J/kg.K (air)

%% Prompt for desired AoA
answer = input('Enter a desired AoA in degrees (between -20 to 20): ', 's');
[alpha_deg, status] = str2num(answer);
while ~status || ~isscalar(alpha_deg) || alpha_deg > 20 || alpha_deg < -20
    answer = input('Invalid angle. Enter again: ', 's');
    [alpha_deg, status] = str2num(answer);
end

alpha = alpha_deg*pi/180; % Convert angle to radians
angle_data = -0.7:0.001:0.7;  % An array of angles, -0.7 to 0.7 radians

%% Prompt for mode
mode = input('Would you like to plot experimental data? (y/n): ', 's');
if isempty(mode)  % give default answer if nothing is provided
    mode = 'y';
end

while ~strcmpi(mode, 'y') && ~strcmpi(mode, 'n')  % check answer is valid
    mode = input('Invalid. Enter ''y'' or ''n'': ', 's');
    if isempty(mode)  % give default answer if nothing is provided
        mode = 'y';
    end
end

if mode == 'y'  % If plotting experimental data, prompt for data filename
    RUN_no = str2func(input('Enter run name in the format RUNXXX: ', 's'));
end

%% Determination of Free Stream Dynamic & Static Pressures
% 'TUSQ_nozzle_exit_conds.m' function by David Buttsworth modified by % Michael Fowler
[rho,U,p_inf] = TUSQ_nozzle_exit_conds(M,p_0,T_0);

q_inf = 0.5*rho*U^2;  % Free stream dynamic pressure, Pa

%% Entering Data & Curve Fitting
% Production of plots to compare fitted functions with experimental data % points

global C_N_body_raw C_A_body_raw x_cp_body_raw C_L_wing_raw C_D_wing_raw

C_N_body_fit = C_N_body_fn(angle_data);

figure(1)
subplot(2,2,1)
plot(C_N_body_raw(:,1).*180/pi,C_N_body_raw(:,2),'x',angle_data.*180/pi,C_N_body_fit)
xlabel('Alpha (degrees)'); ylabel('Coefficient of Normal Force')
title('Experimental & Fitted Data Comparison for Body Normal Force Coefficients')
legend('Experimental Data','Fitted Function','Location','SouthEast');
ylim([0,inf]); xlim([0,30])
grid on

C_A_body_fit = C_A_body_fn(angle_data);
subplot(2,2,2)
plot(C_A_body_raw(:,1).*180/pi,C_A_body_raw(:,2),'x',angle_data.*180/pi,C_A_body_fit)
xlabel('Alpha (degrees)'); ylabel('Coefficient of Axial Force')
title('Experimental & Fitted Data Comparison for Body Axial Force Coefficients')
legend('Experimental Data','Fitted Function','Location','SouthEast');
ylim([0,inf]); xlim([0,30])
grid on

% Converting center of pressure location into percent of body length
x_cp_body_fit = x_cp_body_fn(angle_data)/L;

subplot(2,2,3)
plot(x_cp_body_raw(:,1).*180/pi,x_cp_body_raw(:,2),'x',angle_data.*180/pi,x_cp_body_fit)
xlabel('Alpha (degrees)'); ylabel('Percent Body Length')
title('Experimental & Fitted Data Comparison for Body Centre of Pressure Location')
legend('Experimental Data','Fitted Function','Location','SouthEast');
xlim([0,30]);
grid on

%% Calculating theoretical wing lift & Drag Coefficients
% 'wedge_wing.m' function by David Buttsworth modified by Michael Fowler

% Calculating wing coefficient data
[C_L_wing_raw, C_D_wing_raw] = wedge_wing(0:0.1:30,wedge_angle);
C_L_wing_raw = [(0:0.1:30)',C_L_wing_raw'];% Wing lift coefficient data
C_D_wing_raw = [(0:0.1:30)',C_D_wing_raw'];% Wing drag coefficient data

% Mirroring Lift matrix to obtain lift values at negative AoA
C_L_wing_raw_neg = fliplr(C_L_wing_raw);
C_L_wing_raw = [-C_L_wing_raw_neg;C_L_wing_raw(2:end,:)];

% Convert angle values from degrees to radians
C_L_wing_raw(:,1) = C_L_wing_raw(:,1).*pi/180;
C_D_wing_raw(:,1) = C_D_wing_raw(:,1).*pi/180;

subplot(2,2,4)
plot(C_L_wing_raw(:,1).*180/pi,C_L_wing_raw(:,2),C_D_wing_raw(:,1).*180/pi,C_D_wing_raw(:,2));
xlabel('Alpha (degrees)'); ylabel('Coefficient of Lift')
title('Calculated Data for Wing Lift & Drag Coefficients')
legend('Calculated Lift Data','Calculated Drag Data','Location','SouthEast');
ylim([0,inf]); xlim([0,30])
grid on

%% Beta Calculations
% Determining required values of tail fin actuation beta for various AoA targets
for i = 1:length(angle_data)
    alpha_range = angle_data(i);
    beta_range = angle_data;
    M_pivot = M_pivot_fn(alpha_range,beta_range);
    [~,inx] = min(abs(M_pivot)); % Take value which makes M_pivot closest to 0
    beta(i) = angle_data(inx); % Beta for given alpha value
end

beta_req = interp1(angle_data,beta,alpha);
fprintf('Beta required in degrees is: %2.3g\n',beta_req*180/pi);

% Plot of Beta required for various AoA
figure(2)
plot(angle_data.*180/pi,beta.*180/pi)
xlabel('Alpha (degrees)'); ylabel('Beta (degrees)');
title('Tail Fin Actuation Beta as a Function of AoA')
legend('AoA to Tail fin Actuation Correlation', 'Location', 'SouthEast');
ylim([-30,30]); xlim([-20,20]);
grid on

%% Angular acceleration Calculations
% ODE solver accesses the 'alphadot.m' function which contains the second order differential equation of angular acceleration arranged in Cauchy Form. Initial conditions are 0 and 0 for angular velocity and angular displacement at t = 0. Time range only needs to be small from 0 - 0.25 s.

t_range = [0,0.25]; % Range of times to be checked, seconds
alpha_int = [0, 0]; % Initial conditions for alpha and alpha_dot
[t,x] = ode45('alphadot',t_range,alpha_int);

alpha_position = x(:,1); % Angular position of model
alphadot_velocity = x(:,2); % Angular velocity of model
beta_position = beta_actuation(t); % Beta relative to model axis
beta_relative = beta_position - alpha_position; % Beta relative to flow

% Determining the acceleration
for i = 1:length(alpha_position)
    M_pivot(i) = M_pivot_fn(alpha_position(i), beta_position(i));
    alphadotdot_acc(i) = M_pivot(i)./I_pivot;
end
%% Output plot provides alpha, and beta as a function of time
if mode == 'n'
    figure(3)
    plot(t, alpha_position.*180/pi, t, beta_position.*180/pi, t, beta_relative.*180/pi);
    xlabel('Time (s)'); ylabel('Angle (degrees)');
    legend('AoA, Alpha', 'Tail fin position, Beta', 'Tail fin position relative to Flow', 'Location', 'NorthEast');
    title(['Body & Fin Angles for a ', num2str(alpha_deg), ' degree AoA Manoeuvre']);
    grid on
end

%% Data comparison for position
if mode == 'y'
    RUN_file = RUN_no();
    RUN_time = RUN_file(:,1);
    RUN_alpha = RUN_file(:,2);
    RUN_beta = RUN_file(:,3);

    figure(3)
    plot(t, alpha_position.*180/pi, t, beta_position*180/pi, RUN_time, RUN_alpha, 'o', RUN_time, RUN_beta, 'x');
    xlabel('Time (s)'); ylabel('Angle (degrees)');
    legend('Expected AoA, Alpha', 'Expected tail fin position, Beta', 'Observed AoA, Alpha', 'Observed tail fin position, Beta', 'Location', 'NorthEast');
    title(['Body & Fin Angles for a ', num2str(alpha_deg), ' degree AoA Manoeuvre, and ', func2str(RUN_no), ' data']);
    grid on
end

%% Angular acceleration as a function of time
if mode == 'n'
    figure(4)
    plot(t, alphadotdot_acc.*180/pi);
    xlabel('Time (s)'); ylabel('Angular Acceleration (degrees/s^2)');
    title(['Angular acceleration for a ', num2str(alpha_deg), ' degree AoA Manoeuvre']);
    grid on
end

%% Data comparison for angular acceleration
if mode == 'y'
    RUN_alpha_poly = polyfit(RUN_time, RUN_alpha, 6);
    RUN_alphadotdot_poly = polyder(polyder(RUN_alpha_poly));
    RUN_alphadotdot = polyval(RUN_alphadotdot_poly, RUN_time);

    figure(4)
    plot(t, alphadotdot_acc.*180/pi, t, alphadotdot_acc.*180/pi, RUN_time, RUN_alphadotdot, 'o');
    xlabel('Time (s)'); ylabel('Angular Acceleration (degrees/s^2)');
legend('Expected angular acceleration', 'Observed angular acceleration', 'Location', 'NorthEast')
title(['Angular acceleration for a', num2str(alpha_deg), ' degree AoA manoeuvre, and', func2str(RUN_no), ' data']);
grid on

end

%% Forces on Model Surfaces

% Wing forces
F_L_wing = interp1(C_L_wing_raw(:,1), C_L_wing_raw(:,2), beta_relative).*q_inf.*Ap;
% Lifting force on wing
F_D_wing = interp1(C_D_wing_raw(:,1), C_D_wing_raw(:,2), abs(beta_relative)).*q_inf.*Ap;

% Body forces
F_N_body = C_N_body_fn(alpha_position).*q_inf.*As;
F_A_body = C_A_body_fn(alpha_position).*q_inf.*As;

% Plot of model surface forces as a function of time
figure(5)
plot(t, F_L_wing, t, F_D_wing, t, F_N_body, t, F_A_body);
legend('Wing lift force', 'Wing drag force', 'Body normal force', 'Body axial force');
xlabel('Time (s)'); ylabel('Force (N)');
title(['Model Aerodynamic Forces for a ', num2str(alpha_deg), ' degree AoA Manoeuvre']);
grid on
C.2 M_pivot.m

function [ M_pivot ] = M_pivot_fn( alpha,beta )
%M_pivot_fn Sums the moments acting about the bodies pivot
% Function sums the moments produced by the models weight, body
% normal
% force, wing drag force, and wing lift force. The result is a
% moment
% that is trying to rotate the body. A positive moment makes the
% body
% pitch nose up, whilst a negative moment makes it pitch nose down.

global As q_inf x_pivot Ap x_cp_wing x_cg W C_L_wing_raw C_D_wing_raw

% Calculating the wing lift and drag data for given position
C_L_wing = interp1(C_L_wing_raw(:,1),C_L_wing_raw(:,2),(beta-alpha));

C_D_wing = interp1(C_D_wing_raw(:,1),C_D_wing_raw(:,2),abs(beta-alpha));

% Summing all moments about the pivot
M_pivot = -W.*(x_pivot-x_cg).*cos(alpha) - C_N_body_fn(alpha)...
.As.*q_inf.*(x_cp_body_fn(alpha)-x_pivot) +
Ap.*q_inf.*(x_cp_wing...
-x_pivot).*(-C_D_wing.*sin(alpha)+C_L_wing.*cos(alpha));
end
function [ beta ] = beta_actuation( t )
%beta_actuation Determines the tail fin position at a given time
% Based on the time during the flow this function determines the tail fin
% position based up the motors actuation speed and time the tail fin is
% held at its fully actuated position.

global beta_req delay beta_dot_deg

beta_dot = abs(beta_dot_deg*pi/180); % Convert actuation rate to radians per second

time_1 = abs(beta_req)/beta_dot; % Time when fin reaches beta required

% Time when fin starts returning to beta = 0

j = 0;

while j < length(t);
  j = j+1;
  if 0<=t(j) && t(j)<time_1
    beta = beta_dot .* t(j); % Fin actuation up
  elseif time_1<=t(j) && t(j)<time_2
    beta = abs(beta_req); % Holding fin position for delay time
  else
    beta = abs(beta_req) - beta_dot .* (t(j) - (time_2)); % Fin actuation return
    if beta <= 0;
      beta = 0;
    end
  end
  beta_store(j) = beta;
end

if beta_req < 0
  beta_store = -beta_store;
end

beta = beta_store;
end
C.4 alphadot.m

function [ xdot ] = alphadot(t,x)
%alphadot Determines the AoA, alpha and angular velocity alpha_dot
% Presents the function for angular acceleration in Cauchy form so that
% it can be solved via a ODE solver.

%% Entering Constants
global I_pivot

%% Calculating M_pivot
% Uses function file M_pivot_fn to determine instant model pitching moments

% Determine Beta
beta = beta_actuation(t);

% Determine resulting moment about pivot
M_pivot = M_pivot_fn(x(1), beta);

%% Assigning values to xdot

xdot(1) = x(2);
xdot(2) = M_pivot./I_pivot - 0.*x(2);

xdot = [xdot(1); xdot(2)];

end
function [rho,U,p_inf] = TUSQ_nozzle_exit_conds(M,P0,T0)
% TUSQ_nozzle_exit_conds determines the freestream flow conditions
% from
% the stagnation conditions measured at the nozzle exit. Original
% script
% is by David Buttsworth modified by Michael Fowler for his
% application.
% global k R

p_inf = isenpres(0,M,k)*P0;    % Free stream static pressure
T_inf = isentemp(0,M,k)*T0;    % Free stream temperature
rho = p_inf/R/T_inf;          % Free stream density
a = sqrt(k*R*T_inf);          % Free stream velocity
U = a*M;

function P2P1=isenpres(M1,M2,k);
% G=k/(k-1);
P2P1=isentemp(M1,M2,k).^G;

function T2T1=isentemp(M1,M2,k);
% K=(k-1)/2;
T2T1=(1+K*M1.^2)./(1+K*M2.^2);
C.6  wedge_wing.m

function [C_L_wing, C_D_wing] = wedge_wing(alpha, wedge_angle)
%wedge_wing Calculates wing lift & drag components by Newtonian flow
% theory
% Accepts an input array of angles for wing angle of attack to flow in
% degrees, and a wedge angle in degrees. It then calculates the
% coefficient of pressure on the wings upper and lower surfaces which can
% then be converted into wing lift and drag components. The original
% function was produced by David Buttsworth, and modified by Michael
% Fowler to provide data for a greater range of angles of attack, and to
% determine the lift and drag components.

%% Entering constants
global M p_inf q_inf k

p2p1_upper = zeros(size(alpha));
p2p1_lower = zeros(size(alpha));

%% Determining pressure ratio on wing surface turning away from flow

for i = 1:length(alpha)
    if alpha(i) < wedge_angle/2
        % flow deflection angle on lower side of wing gets lower with
        wing deflection angle
        omega_lower = wedge_angle/2 - alpha(i);
        f_lower = @(M,theta,k) omega_shock(M,theta,k) - omega_lower;
        theta_lower = fzero(@(theta) f_lower(M,theta,k),20);
        p2p1_lower(i) = pshock(M,theta_lower,k);
        p2p1_lower_interp(i) = p2p1_lower(i);
    else
        p2p1_lower(i) = interp1(alpha(1:length(p2p1_lower_interp)),p2p1_lower_interp,alpha(i),
        'spline');
    end
    if p2p1_lower(i) <= 0.5
        p2p1_lower(i) = 0.5;
    end
end

%% Determining pressure ratio on wing surface turning towards flow

for i = 1:length(alpha)
    % flow deflection angle on upper side of wing gets larger with
    wing deflection angle
    omega_upper = wedge_angle/2 + alpha(i);
    f_upper = @(M,theta,k) omega_shock(M,theta,k) - omega_upper;
    theta_upper = fzero(@(theta) f_upper(M,theta,k),20);
    p2p1_upper(i) = pshock(M,theta_upper,k);
end

%% Converting pressure ratios into lift and drag coefficients
\[
\text{CN\_surface\_lower} = ((p2p1\_lower.*p\_inf-p\_inf)/q\_inf) \; ; \; \% \; \text{Normal force coefficient on lower surface of wing}
\]

\[
\text{CN\_surface\_upper} = ((p2p1\_upper.*p\_inf-p\_inf)/q\_inf) \; ; \; \% \; \text{Normal force coefficient on upper surface of wing}
\]

\[
\text{C\_D\_wing} = -\text{CN\_surface\_lower}.*\sin(\alpha - \text{wedge\_angle}/2) + \text{CN\_surface\_upper}.*\sin(\alpha+\text{wedge\_angle}/2);
\]

\[
\text{C\_L\_wing} = -\text{CN\_surface\_lower}.*\cos(\alpha - \text{wedge\_angle}/2) + \text{CN\_surface\_upper}.*\cos(\alpha+\text{wedge\_angle}/2);
\]

\begin{verbatim}
function p2p1 = pshock(M,theta,k)
    \% shock pressure ratio as function of shock angle
    \% theta=theta*pi/180;
    if theta > pi/2,
        theta=pi/2;
    end;
    p2p1=1+2*k/(k+1)*(M.^2*(\sin(theta))^2-1);
\end{verbatim}

\begin{verbatim}
function omega = omega\_shock(M,theta,k)
    \% shock flow turning angle as function of shock angle
    \% t=theta*pi/180;
    if t > pi/2,
        t=pi/2;
    end;
    omega=atan(2/tan(t)*(M.^2*(\sin(t))^2-1)/(M.^2*(k+cos(2*t))+2));
    if omega < 1e-10,
        omega=0;
    end;
    omega = omega*180/pi;
\end{verbatim}
C.7 C_A_body_fn.m

function [ C_A_body ] = C_A_body_fn( alpha )
%C_A_BODY_fn Contains the raw data for axial force coefficients on the body
%    Accepts an input angle in radians and outputs the corresponding body
%    axial force coefficient.

global C_A_body_raw

alpha = abs(alpha);

% Raw C_N data for body (angles in degrees)
C_A_body_raw = [0,0.0944; 1,0.0833; 2,0.0888; 4, 0.0944; 6,0.1;
             8,0.1055;... 
            10,0.1167; 12,0.125; 14,0.1333; 16,0.15; 18,0.1639; 20,0.1694;...
            22,0.1833; 24,0.2167; 26,0.225; 28,0.2278; 30,0.2361];

% Convert angle values from degrees to radians
C_A_body_raw(:,1) = C_A_body_raw(:,1).*pi/180;

% Interpolate desired value using suitable function fit
C_A_body =
    interp1(C_A_body_raw(:,1),C_A_body_raw(:,2),alpha,'spline');

end
C.8  C_N_body_fn.m

function [ C_N_body ] = C_N_body_fn(alpha)
%C_N_BODY_fn Contains raw data for normal force coefficients on the body
%   Accepts an input angle in radians and outputs the corresponding body
%   normal force coefficient

global C_N_body_raw

% Raw C_N data for body
C_N_body_raw = [0,0; 1,0.056; 2,0.111; 4,0.167; 6,0.333; 8,0.556; 10,0.778; ...
    12,1.00; 14,1.278; 16,1.556; 18,1.889; 20,2.222; 22,2.556; 24,2.944; ...
    26,3.333; 28,3.722; 30,4.222];

C_N_body_raw_neg = fliplr(C_N_body_raw');

C_N_body_raw = [-C_N_body_raw_neg';C_N_body_raw(2:end,:)];

% Convert angle values from degrees to radians
C_N_body_raw(:,1) = C_N_body_raw(:,1).*pi/180;

% Interpolate desired value using suitable function fit
C_N_body = interp1(C_N_body_raw(:,1),C_N_body_raw(:,2),alpha,'pchip');

end
C.9  x_cp_body_fn.m

function [ x_cp_body ] = x_cp_body_fn(alpha)
%x_cp_body_fn Contains raw data for centre of pressure of the body
%   Accepts an input angle in radians and outputs the corresponding
%   body centre of pressure as distance from the model nose.

% Entering constants
global L x_cp_body_raw

% Considering only the absolute AoA
alpha = abs(alpha);

% x_cp_body_raw values are based on percent body length from model nose
x_cp_body_raw = [0,0.4944; 1,0.20271; 2,0.40000; 4,0.38333; 6,0.45417;
                8,0.47083; 10,0.47917; 12,0.48889; 14,0.49583; 16,0.50278;
                18,0.50972;...
                20,0.51389; 22,0.51944; 24,0.52083; 26,0.52500; 28,0.52917;
                30,0.53194];

% Convert angle values from degrees to radians
x_cp_body_raw(:,1) = x_cp_body_raw(:,1).*pi/180;

% Interpolate desired value using suitable function fit
x_cp_body = interp1(x_cp_body_raw(:,1),x_cp_body_raw(:,2),alpha,'pchip').*L;

end
C.10 RUN283.m

function [ RUN283_data ] = RUN283( )

% RUN283 Contains the observed data from RUN283
%   Data from run 283 is placed in this function file to provide
%   simple
%   access via the main script.

% Providing array of times with steps of 2/1500 frames per second
time = [0:2/1500:0.039]; % Time increments are in two frames per data point

% Angle of attack data (nose up +ve)
alpha = [0 -0.5 -1 -1 -1 -2 -2 -2 -1.5 -1 -0.5 0 1 1.5 2.5 3 4 ...
        4.5 5.5 6.5 7 8 9 10 10 10 10.5];

% Control surface angle of attack (nose down +ve)
beta = [0 0.5 1 2 4 5 5.5 6.5 5.5 7.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5 9.5 9 ... 
       8.5 8.5 8.5 8.5 8.5 5.5 4.5 3.5 1.5 0.5 0.5];

% Compiling data
RUN283_data = [time',alpha',beta'];

end
C.11 RUN286.m

function [ RUN286_data ] = RUN286( )
%RUN286 Contains the observed data from RUN286
% Data from run 286 is placed in this function file to provide simple
% access via the main script.
% Providing array of times with steps of 4/3000 frames per second
% time = [0:4/3000:0.089]; % Time increments are in four frames per data point
% Angle of attack data (nose up +ve)
alpha = [0 0 0 0 0 0 0 0 0 0 0 0.5 0.5 0.5 1 1.5 2 2 2.5 3.0 3.5 3.5 ...
        4 4 4 3.5 3.5 3.5 3.5 3 2 1.5 1 0 -0.5 -1 -1.5 -2 -2.5 -2.5 -3 -
        3.5 ...
        -4 -4 -4.5 -4 -3.5 -3 -2.5 -1.5 -0.5 1 2.5 4 5 6.5 7.5 9 10.5 12 ...
        14 15.5 16 17.5 19.5 18];
% Control surface angle of attack relative to body (nose down +ve)
beta = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -0.5 -1 -1.5 -
       1.5 ... -1.5 -1 -0.5 -0.5 -0.5 -0.5 0 0.5 0.5 1 2 2.5 3 4 4.5 4.5 4.5 5 5 ...
       5.5 5.5 5.5 6.5 6.5 6.5 6.5 7 6.5 7 7 17 7 7 7 7 7 15.5 5 4 2];
% Compiling data
RUN286_data = [time',alpha',beta'];
end
Appendix D – Results of Theoretical Analysis

Figure D.1: Expected body & fin angles for a 5 degree AoA manoeuvre
Figure D.2: Expected angular acceleration for a 5 degree AoA manoeuvre

Figure D.3: Expected body & control surface forces for a 5 degree AoA manoeuvre
Figure D.4: Expected body & fin angles for a 15 degree AoA manoeuvre

Figure D.5: Expected angular acceleration for a 15 degree AoA manoeuvre
Figure D.6: Expected body & control surface forces for a 15 degree AoA manoeuvre
Appendix E – Risk Assessment

E.1 Appendix Outline
Throughout the manufacture, assembly, and testing of the model there will be various risks that arise. To prevent potential damage to equipment or injury to persons an appropriate identification of potential risks must be made. Following this must be an assessment of these risks and where applicable appropriate measures are to be implemented to reduce the risk level. The procedure that will be outlined here is taken from the ENG3003 Engineering Management: Study Book 2 (2014) it consists of 4 steps; Step 1: Hazard Identification, Step 2: Risk Assessment, Step 3: Risk Control, and Step 4: Monitor & Review.

E.2 Potential Risks
The first step in the risk assessment is to identify the different possible hazards. During the manufacture & assembly of the model, these include:

- Soldering components
- Handling materials with sharp edges or burrs
- Electrical hazards from circuitry power supplies
- Operation of hand held grinder for cutting components
- Moving tail fins present a pinch hazard.

Similarly there are hazards encountered during the setup and operation of the hypersonic facility and associated equipment, these include:

- Pressurised Air Cylinders connected to the hypersonic tunnel
- Noise due to high pressure air & pumping equipment
- Trip Hazards over equipment cabling
- Closing test section and other moving parts present a pinch hazard.

E.3 Risk Assessment & Management
This section will cover the remaining steps in the risk assessment process which include, Step 2: Risk assessment, Step 3: Risk Control, and Step 4: monitor and review.

In performing a risk assessment a risk assessment matrix will be used. Table E.1 shows an adapted risk assessment matrix provided in the ENG3003 Engineering Management: Study book 2 (2014). For each of the hazards presented previously the consequences involved will be rated either based on the injury they could cause or the damage they may cost with the current control measures in place. The likelihood of this event occurring is then considered and based upon these two factors a risk level is identified. The results of this assessment are then recorded in Table E.2. In situations where the risk level
identified is higher than a low risk activity further control measures will need to be considered.

As identified in Table E.2 the current risk control procedures in place are sufficient to provide a low risk level. This means that for all considered tasks no additional precautions or controls are required. However as part of step 4 the current procedures should be continually monitored and any changes to the hazards should be accounted for by a review of this risk assessment.

**Table E.1: Risk assessment matrix**

<table>
<thead>
<tr>
<th>Probability</th>
<th>Consequence 1 - Insignificant No Injury 0-$5K</th>
<th>Consequence 2 - Minor First Aid $5K-$50K</th>
<th>Consequence 3 - Moderate Med Treatment $50K-$100K</th>
<th>Consequence 4 - Major Serious Injuries $100K-$250K</th>
<th>Consequence 5 - Catastrophic Death &gt; $250K</th>
</tr>
</thead>
<tbody>
<tr>
<td>A - Almost Certain 1 in 2</td>
<td>M</td>
<td>H</td>
<td>E</td>
<td>E</td>
<td>E</td>
</tr>
<tr>
<td>B - Likely 1 in 100</td>
<td>M</td>
<td>H</td>
<td>H</td>
<td>E</td>
<td>E</td>
</tr>
<tr>
<td>C - Possible 1 in 1000</td>
<td>L</td>
<td>M</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>D - Unlikely 1 in 10 000</td>
<td>L</td>
<td>L</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>E - Rare 1 in 1 000 000</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
</tr>
</tbody>
</table>

**Recommended Action Guide**

- **E** = Extreme Risk – Task **MUST NOT** proceed
- **H** = High Risk – Special Procedures Required
- **M** = Moderate Risk – Risk Management Plan/Work Method Statement Required
- **L** = Low Risk – Use Routine Procedures
<table>
<thead>
<tr>
<th><strong>Risk</strong></th>
<th><strong>Consequence</strong></th>
<th><strong>Risk Level</strong></th>
<th><strong>Current Control</strong></th>
<th><strong>Additional Control Required</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Soldering Components</td>
<td>Potential for personal burns or fire hazard</td>
<td>D2 – Low Risk</td>
<td>Rest soldering iron on stand when not in use Do not leave unattended when hot Use clamp to hold components while soldering</td>
<td>Low Risk no further Control required</td>
</tr>
<tr>
<td>Material Sharp Edges &amp; Burrs</td>
<td>When cutting shafts to size they will leave burrs which could cause small cuts</td>
<td>D1 – Low Risk</td>
<td>After cutting parts clean sharp edges with a file Wear gloves when handling sharp materials</td>
<td>Low Risk no further control required</td>
</tr>
<tr>
<td>Electrical Hazards from Power Supplies</td>
<td>When prototyping model electronics power sources present an electric shock hazard</td>
<td>E1 – Low Risk</td>
<td>Voltage and current of supplies is sufficiently low to not cause injury Circuits are only worked on with power disconnected</td>
<td>Low Risk no further control required</td>
</tr>
<tr>
<td>Operating Hand held Grinder</td>
<td>When cutting metal components the use of a grinder presents an injury and fire hazard</td>
<td>D2 – Low Risk</td>
<td>Gloves and safety glasses are worn when operating Grinder is operated in a clear environment with fire hazards removed</td>
<td>Low Risk no further control required</td>
</tr>
<tr>
<td>Model Tail fin Operation</td>
<td>Potential to cause a pinch hazard</td>
<td>E1 – Low Risk</td>
<td>Motor torque is sufficiently low that it will not cause personal injury</td>
<td>Low Risk no further control required</td>
</tr>
<tr>
<td>Pressurised Air Cylinders which operate the tunnel</td>
<td>Fittings could blow off or air could be discharged through pressure reliefs without warning</td>
<td>D1 – Low Risk</td>
<td>Wear eye protecting when operating the facility Ensure fittings/pipe is in good condition before operation</td>
<td>Low Risk no further control required</td>
</tr>
<tr>
<td>Noise</td>
<td>High velocity air and air pumps will be operating causing large amounts of noise causing hearing damage</td>
<td>D1 – Low Risk</td>
<td>Hearing protection is worn if the noise level is potentially going to be above a damaging amount</td>
<td>Low Risk no further control required</td>
</tr>
<tr>
<td>Trip Hazards over cabling connecting equipment</td>
<td>Could cause personal injury if someone falls or could pull equipment down and cause damage</td>
<td>D2 – Low Risk</td>
<td>Become familiar with facility before operating equipment Be aware whilst walking Remove all trip hazards where possible</td>
<td>Low Risk no further control required</td>
</tr>
<tr>
<td>Closing the tunnel test section &amp; moving parts</td>
<td>Pinch hazard when closing the test section flanges or from moving parts</td>
<td>D2 – Low Risk</td>
<td>Keep body away from all potential pinch points Follow instruction of experienced technicians</td>
<td>Low Risk no further control required</td>
</tr>
</tbody>
</table>
Appendix F – Actuation System Development

F.1 Appendix Outline
This appendix documents the stages of development in the tail fin actuation system. The development has been split into 3 stages with each progressing towards the final item for implementation into the model. The final code used is also included at the end of this Appendix.

F.2 Stage 1

F.2.1 Setup
Stage 1 was a very basic setup with its main purpose being to expand on the author's limited prior knowledge of Arduino microcontroller based systems. This system used an Arduino UNO R3 microcontroller, a DFRduino L298P Dual 2 amp motor driver shield, and a 12 VDC bipolar stepper motor with a 7.5-degree full-stepping angle.

The setup of this system is very simple as the motor driver is designed to press onto the Arduino microcontroller. Each coil of the stepper motor then connects to a motor terminal. Motor 1 terminals (the blue and black wires coming from the controller) control the A coil and the motor 2 terminals (the yellow and orange wires) control the B coil (See Figure F.1). Finally the orange and yellow leads on top of the motor driver are connected to the voltage in and ground pins respectively. These are then connected to a variable voltage power supply which is set to 12 VDC.

F.2.2 Arduino Script File
The following script is very basic as its main purpose was to educate the author in the programing language and in microcontrollers in general. Because it is using a motor driver as opposed to a stepper driver the inputs available are a direction and on/off for each motor. The way a stepper motor works is to
energise the A coil by switching it ‘on’, to advance one step the A coil is switched ‘off’ and the B coil switched ‘on’. To advance another step the B coil is switch ‘off’ and the A coil is switched ‘on’ but this time the current direction is reversed. Finally the A coil is switched ‘off’ and the B coil is switched ‘on’ with its current reversed. This pattern is then repeated until the desired number of steps is reached.

The script file makes use of this principle by treating the motor driver on/off function as the on/off function for each coil, and the motors direction change function as the normal/reverse direction of current in the coil. It also implements a simple step counter, which allows the motor to be stopped after a desired number of steps. The code is seen below;

```cpp
// Arduino UNO stepper using motor shield_rev 0
int motorPin1 = 5; // on/off pin motor 1
int motorPin2 = 4; // direction pin motor 1
int motorPin3 = 6; // on/off pin motor 2
int motorPin4 = 7; // direction pin motor 2
int delayTime = 500; // Delay between steps (smaller faster rotation)
int Distance = 0;
int STOP = 20; // Sets the number of steps to take before stopping

void setup() {
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(motorPin3, OUTPUT);
  pinMode(motorPin4, OUTPUT);
}

void loop() {
  digitalWrite(motorPin1, HIGH); // A coil on
  digitalWrite(motorPin2, LOW); // A coil normal direction
  digitalWrite(motorPin3, LOW); // B coil off
  digitalWrite(motorPin4, LOW); // B coil normal direction
  delay(delayTime);
  Distance = Distance + 1; // Advance step counter

  if (Distance == STOP) // Hold current step if step limit reached
  {
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);

    while(1) {} //}
  }
}
```
digitalWrite(motorPin1, LOW); // A coil off
digitalWrite(motorPin2, LOW); // A coil normal direction
digitalWrite(motorPin3, HIGH); // B coil on
digitalWrite(motorPin4, LOW); // B coil normal direction

delay(delayTime);

Distance = Distance + 1; // Advance step counter

if (Distance == STOP)
{
  digitalWrite(motorPin1, LOW);
  digitalWrite(motorPin2, LOW);
  digitalWrite(motorPin3, HIGH);
  digitalWrite(motorPin4, LOW);

  while(1) {} }
}
digitalWrite(motorPin1, HIGH); // A coil on
digitalWrite(motorPin2, HIGH); // A coil reverse direction
digitalWrite(motorPin3, LOW); // B coil off
digitalWrite(motorPin4, HIGH); // B coil reverse direction
delay(delayTime);

Distance = Distance + 1; // Advance step counter

if (Distance == STOP)
{
  digitalWrite(motorPin1, HIGH);
  digitalWrite(motorPin2, HIGH);
  digitalWrite(motorPin3, LOW);
  digitalWrite(motorPin4, HIGH);

  while(1) {} }
}
digitalWrite(motorPin1, LOW); // A coil off
digitalWrite(motorPin2, HIGH); // A coil reverse direction
digitalWrite(motorPin3, HIGH); // B coil on
digitalWrite(motorPin4, HIGH); // B coil reverse direction
delay(delayTime);

Distance = Distance + 1; // Advance step counter

if (Distance == STOP)
{
  digitalWrite(motorPin1, LOW);
  digitalWrite(motorPin2, HIGH);
  digitalWrite(motorPin3, HIGH);
```c
digitalWrite(motorPin4, HIGH);

while(1) {} 
```

### F.2.3 Results
- It was possible to move the stepper a prescribed number of steps and have it hold position whilst providing a holding torque
- Current draw for the stepper motor was about 1.8 amps at 12V

### F.2.4 Findings
- It will not be possible to provide a holding torque for the full test setup duration (approximately 1 hour) as the current draw will likely be too great even for the smaller stepper seen in stage 2
- The motor had to be disconnected when uploading programs to the Arduino microcontroller otherwise it tried to operate from USB power which would likely cause damage to board regulators.

### F.3 Stage 2

#### F.3.1 Setup
This setup made much further progress which started implementing most of the components that were to be used in the final model. This used the Arduino UNO R3 microcontroller, Pololu DRV8834 stepper motor driver, a 15mm diameter bipolar stepper motor, and a 14mm micro switch.

Because this setup is considerably more complex Figure F.2 shows an electrical schematic of how everything was connected.

![Figure F.2: Stage 2 electrical schematic](image-url)
Figure F.3 also shows the actual testing setup where the Arduino board can be seen to the left, the stepper driver is mounted on the breadboard along with the micro switch, and to the right is the 15mm stepper motor. Also mounted on the breadboard in the lower left is a 220 micro Farad capacitor, its original application was to smooth incoming power so that if the motor suddenly starts it does not draw excessive power and cause the Arduino board to reset. It was later found to not be required.

![Figure F.3: Stage 2 setup](image)

This system makes use of the configurable pins; M0, M1, STEP, and DIR on the stepper driver. M0 and M1 are used to configure micro stepping, and depending on the input to these pins different step modes can be selected. The STEP pin commands a step to occur and the DIR pin sets the direction of the steps. The SLP pin turns the stepper driver on and off, this is connected to the 5V regulated supply from the Arduino board to provide a HIGH signal to keep it on. The stepper then connects to the A1 and A2 pins for coil A, and B1 and B2 pins for coil B. Also connected to the microcontroller is a switch when open the signal on this pin is HIGH, when depressed it puts the pin to ground and provides a LOW signal.

### F.3.2 Arduino Script File

This script file is considerably more complex than the previous one and incorporates the use of a micro switch to trigger the fin actuation and to select a different fin actuation to run depending on the number of times the switch is depressed.

Initially it waits for the switch to be pressed and upon release it runs the first fin actuation mode. This advances a counter and the program returns to the beginning. Again it waits for the switch to be pressed and released however it now runs the second fin actuation mode, again it returns to the beginning of the program. Pressing the switch a third time runs the final fin actuation mode at which point the program returns to the beginning ready to run the first mode again.

For development purposes the modes programed are 2 rotations forward and 2 rotations backward for mode 1, half a rotation forward and half a rotation
backward for mode 2, and a quarter rotation forward and a quarter rotation backward for mode 3. For the actual model these will be varied to suit the required movements. The following is the code used;

```c
// Full stepper control with modes for DRV8834_rev 0
const int stepPin = 4;  // Step pin
const int dirPin = 3;   // Direction pin
const int M1Pin = 5;   // Micro step configure pin
const int M0Pin = 6;   // Micro step configure pin
const int switchPin = 2; // Switch pin
const int modes = 3;
const int STOP = 40;
const int timegap = 5;  // Time gap between steps

void setup() {
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(M1Pin, OUTPUT);
  pinMode(M0Pin, OUTPUT);  // Set to INPUT and M0Pin LOW for high-impedance
  pinMode(switchPin, INPUT);
  digitalWrite(switchPin, HIGH);
  digitalWrite(M1Pin, LOW);
  digitalWrite(M0Pin, LOW);  // Set to LOW and M0Pin to INPUT for high-impedance
  digitalWrite(stepPin, LOW);
  digitalWrite(dirPin, LOW);
}

void loop() {

  int buttonstate = digitalRead(switchPin);

  for (int j=0; j < modes; j++) { // Allows simple configuration of motor without programmer

    while (digitalRead(switchPin) == HIGH) { // Wait for switch depression
      delay(1);
    }

    while (digitalRead(switchPin) == LOW); { // Wait for switch release
      delay(1);
    }

    // Mode 1 Coding, 2 rotations both directions
    if (buttonstate == HIGH && j == 0) {
      digitalWrite(dirPin, LOW);
    }
  }
```
for (int i=0; i < STOP; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}

delay(1000);

digitalWrite(dirPin, HIGH);

for (int i = 0; i < STOP; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}

// Mode 2 Coding, half rotations both directions
if (buttonstate == HIGH && j == 1) {

digitalWrite(dirPin, LOW);
for (int i=0; i < 10; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}

delay(1000);

digitalWrite(dirPin, HIGH);

for (int i = 0; i < 10; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}

// Mode 3 Coding, quarter rotations both directions
if (buttonstate == HIGH && j == 2) {

digitalWrite(dirPin, LOW);

for (int i=0; i < 5; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
}


delay(timegap);
}
delay(1000);
digitalWrite(dirPin, HIGH);

for (int i = 0; i < 5; i++) {

digitalWrite(stepPin, HIGH);
delay(timegap);
digitalWrite(stepPin, LOW);
delay(timegap);
}
delay(1);
} }

F.3.3 Results
• It was possible to utilise stepper motor micro stepping including ½, ¼, 1/8, 1/16 and 1/32 step sizes
• Reaction times for the motor after release of the switch were almost instant however confirmation via high speed footage would be needed for actual measurement as times of interest are in ms
• Current draw of the stepper was between 200 – 300 mA at 8 VDC after adjustment of the stepper board current limiter

F.3.4 Findings
• It was found that the 15mm stepper motor has 20 steps at 18 degrees per step
• Micro stepping allows the motor to be positioned at up to 1/32 of a full step (nominally 0.56 degrees for this motor) and can be actively held there
• The micro steps are not perfect as they are not all exactly the same number of degrees. Some are larger/smaller than the theoretical amount, this is because the micro steps are created through current variation of the A and B coils in the motor, which is not perfect.
• Although micro steps were not perfect they were consistent, this suggests a trial and error approach can be used to set actuation angles
• When current to the motor is cut off it centres itself at a half step position (between the A and B coil), this needs to be considered to set the zero degree fin position
• Current draw is still too high to have the stepper driver enabled for the full test duration, this suggests the stepper board needs to be switched off until required
• The program has to completely run through for the next mode to be ready which is tedious
F.4  Stage 3

F.4.1  Setup
Stage 3 implements a system ready for installation into the final model. It uses the Pololu A-Star 32U4 microcontroller, Pololu DRV8834 stepper driver, 15mm diameter bi-polar stepper motor, 14mm micro switch, and 2 CR2 lithium ion rechargeable batteries with a nominal voltage of 3.6 V and 600 mAh capacity.

Again the setup is shown through an electrical schematic in Figure F.4 below. The features of this circuit are essentially the same as shown in stage 2 the main differences being the replacement of the Arduino microcontroller with the much more compact Pololu A-Star microcontroller, and the use of 2 CR2 Lithium ion rechargeable batteries as the power source. These are also the batteries, which will be used, in the finished model.

![Figure F.4: Stage 3 electrical schematic](image)

This setup is shown in the top of Figure F.5 where the Pololu microcontroller and stepper driver are soldered together via short pieces of breadboard wire. Together they form a compact unit with only 4 outgoing wires to operate the stepper motor and 3 incoming wires, 2 for the power supply and 1 to receive the signal for the trigger switch. Obviously once implemented into the model the breadboard and alligator leads are replaced by more permanent connections.
F.4.2 Arduino Script File

This script file provides further adjustments to the previous one used but utilises the same structure. The most noticeable changes is the inclusion of a additional control to set the DRV8834 to sleep and wake up, a feature which will significantly reduce battery consumption when the motor is not running. Also added is a feature to prevent the program starting until the switch is pressed, this will stop the board unintentionally running whilst being programmed via USB.

The mode can now be selected without running through the entire program as was required previously. Now the switch is simply depressed the number of times depending on the mode required. To confirm that the correct program was selected it runs a test showing the actuation mode, following this it waits for the switch to be depressed and released. This will simulate the model being loaded into the tunnel (switch depressed) and subsequently being released at flow onset (switch released). The code is included below.

```cpp
// Full stepper control with modes for DRV8834_rev A
const int stepPin = 3; // Step pin
const int dirPin = 2; // Direction pin
const int M1Pin = 5; // Micro step configure pin
const int M0Pin = 6; // Micro step configure pin
const int slpPin = 4; // Sleep on/off pin
const int switchPin = 7; // Trigger switch pin
const int modes = 3;
const int STOP = 40;
const int timegap = 5; // Time gap between steps
```
void setup() {
    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);
    pinMode(M1Pin, OUTPUT);
    pinMode(M0Pin, OUTPUT); // Set to INPUT and M0 pin LOW for high-impedance
    pinMode(switchPin, INPUT);
    pinMode(slpPin, OUTPUT);
    digitalWrite(switchPin, HIGH);
    digitalWrite(M1Pin, LOW);
    digitalWrite(M0Pin, LOW); // Set to LOW and M0Pin to INPUT for high-impedance
    digitalWrite(stepPin, LOW);
    digitalWrite(dirPin, LOW);
    digitalWrite(slpPin, LOW);
}

void loop() {
    int finish = 0;
    int j = 0;

    while (digitalRead(switchPin) == HIGH) { // Prevents program running when USB connected
        delay(1);
    }
    delay(1000);

    for (int k=0; k < 5000; k++) { // 5 sec delay to program mode
        if (digitalRead(switchPin) == LOW) {
            j = j + 1;
            while(digitalRead(switchPin) == LOW) {
                delay(1);
            }
        }
        delay(1);
    }

    while(finish == 0 && j < modes) {
        // Mode 1 Coding
        if (j == 0) {
            // Run test of mode 1
            digitalWrite(slpPin,HIGH);
            digitalWrite(dirPin, LOW);

            for (int i=0; i < STOP; i++) {

digitalWrite(stepPin, HIGH);
delay(timegap);
digitalWrite(stepPin, LOW);
delay(timegap);
}
delay(1000);
digitalWrite(dirPin, HIGH);

for (int i = 0; i < STOP; i++) {
    digitalWrite(stepPin, HIGH);
delay(timegap);
    digitalWrite(stepPin, LOW);
delay(timegap);
}
digitalWrite(slpPin, LOW);

// Actual run of mode 1
while (digitalRead(switchPin) == HIGH) { // Wait for switch to be depressed
delay(1);
}

while (digitalRead(switchPin) == LOW) { // Wait for switch release
    delay(1);
}
digitalWrite(slpPin, HIGH);
digitalWrite(dirPin, LOW);

for (int i = 0; i < STOP; i++) {
    digitalWrite(stepPin, HIGH);
delay(timegap);
    digitalWrite(stepPin, LOW);
delay(timegap);
}
delay(1000);
digitalWrite(dirPin, HIGH);

for (int i = 0; i < STOP; i++) {
    digitalWrite(stepPin, HIGH);
delay(timegap);
    digitalWrite(stepPin, LOW);
delay(timegap);
}
digitalWrite(slpPin, LOW);
finish = 1;
}
// Mode 2 Coding
if (j == 1) {
    // Run test of mode 2
    digitalWrite(slpPin,HIGH);
    digitalWrite(dirPin, LOW);
    
    for (int i=0; i < 10; i++) {
        digitalWrite(stepPin, HIGH);
        delay(timegap);
        digitalWrite(stepPin, LOW);
        delay(timegap);
    }
    digitalWrite(slpPin,LOW);
    digitalWrite(dirPin, HIGH);
    for (int i = 0; i < 10; i++) {
        digitalWrite(stepPin, HIGH);
        delay(timegap);
        digitalWrite(stepPin, LOW);
        delay(timegap);
    }
    digitalWrite(slpPin,HIGH);
    digitalWrite(dirPin, LOW);
    
    // Actual run of mode 2
    while(digitalRead(switchPin) ==HIGH) {// Wait for switch to be depressed
        delay(1);
    }
    while(digitalRead(switchPin) ==LOW) {// Wait for switch release
        (flow start)
        delay(1);
    }
    digitalWrite(slpPin,HIGH);
    digitalWrite(dirPin, LOW);
    for (int i=0; i < 10; i++) {
        digitalWrite(stepPin, HIGH);
        delay(timegap);
        digitalWrite(stepPin, LOW);
        delay(timegap);
    }
    delay(1000);
digitalWrite(dirPin, HIGH);

for (int i = 0; i < 10; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}
digitalWrite(slpPin, LOW);
finish = 1;
}

// Mode 3 Coding
if (j == 2) {
    // Run test of mode 3
    digitalWrite(slpPin, HIGH);
    digitalWrite(dirPin, LOW);

    for (int i = 0; i < 5; i++) {
        digitalWrite(stepPin, HIGH);
        delay(timegap);
        digitalWrite(stepPin, LOW);
        delay(timegap);
    }
delay(1000);

digitalWrite(dirPin, HIGH);

for (int i = 0; i < 5; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}
digitalWrite(slpPin, LOW);
// Actual run of mode 3
while (digitalRead(switchPin) == HIGH) {// Wait for switch to be depressed
    delay(1);
}
while (digitalRead(switchPin) == LOW) {// Wait for switch release
    delay(1);
}
digitalWrite(slpPin, HIGH);
digitalWrite(dirPin, LOW);

for (int i = 0; i < 5; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}
```cpp
digitalWrite(stepPin, HIGH);
delay(timegap);
digitalWrite(stepPin, LOW);
delay(timegap);
}
delay(1000);
digitalWrite(dirPin, HIGH);
for (int i = 0; i < 5; i++) {
  digitalWrite(stepPin, HIGH);
delay(timegap);
digitalWrite(stepPin, LOW);
delay(timegap);
}
digitalWrite(slpPin, LOW);
finish = 1;
}
delay(1);
}
}

F.4.3 Results
- The microcontroller and stepper driver package is quite compact and
  occupies a space of about 25mm long, 15mm wide, and 12mm deep
- Setting the stepper driver into sleep mode reduces current draw from
  about 250 mA down to 35 mA
- Waking the stepper driver from sleep mode appears to occur almost
  instantly
- The battery pack supplies a full charge voltage of 8.2 V and theoretically
  this should drop to 5.5 V at maximum discharge
- Sudden start-up of the motor does not drain excessive battery power and
  cause the boards to lose power and reset

F.4.4 Findings
- Overall the system operated very effectively and consistently
- Demonstration of the selected manoeuvre before setting the model up for
  flow triggering works well and will be useful during experimental setup
- The trigger, batteries, microcontroller & stepper package, and the
  stepper motor are sufficiently compact to fit easily inside a 20mm inner
  diameter tube as required for the model selected

F.5 Final Code Used in all Runs
The code provided below was used for all runs with modifications only made to
the number of steps and step directions in the different mode profiles. It is very
similar to the code presented in Section F.4. The main addition is the use of the
```
controller LED, which flashes after a successful actuation, this helps confirm where in the program the microcontroller is.

```c
const int stepPin = 3; // Step pin
const int dirPin = 2; // Direction pin
const int M1Pin = 5; // Micro step configure pin
const int M0Pin = 6; // Micro step configure pin
const int slpPin = 4; // Sleep on/off pin
const int switchPin = 7; // Trigger switch pin
const int modes = 3;
const int timegap = 5; // Time gap between steps
int LED = 13;

void setup() {
    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);
    pinMode(M1Pin, OUTPUT);
    pinMode(M0Pin, OUTPUT); // Set to INPUT and M0 pin LOW for high-impedance
    pinMode(switchPin, INPUT);
    pinMode(slpPin, OUTPUT);
    pinMode(LED, OUTPUT);
    digitalWrite(switchPin, HIGH);
    digitalWrite(M1Pin, LOW);
    digitalWrite(M0Pin, LOW); // Set to LOW and M0Pin to INPUT for high-impedance
    digitalWrite(stepPin, LOW);
    digitalWrite(dirPin, LOW);
    digitalWrite(slpPin, LOW);
    digitalWrite(LED, LOW);
}

void loop() {
    int finish = 0;
    int j = 0;
    while (digitalRead(switchPin) == HIGH) { // Prevents program running when USB connected
        delay(1);
    }
    delay(1000);
    for (int k=0; k < 5000; k++) { // 5 sec delay to program mode
        if (digitalRead(switchPin) == LOW) {
            j = j + 1;
            while(digitalRead(switchPin) == LOW) {
                delay(1);
            }
        }
    }
```
while (finish == 0 && j < modes) {

    // Mode 1 Coding - Setup for 12.6 degrees of fin actuation
    if (j == 0) {
        // Run test of mode 1
        digitalWrite(slpPin, HIGH);
        digitalWrite(dirPin, HIGH);

        for (int i=0; i < 7; i++) {
            digitalWrite(stepPin, HIGH);
            delay(timegap);
            digitalWrite(stepPin, LOW);
            delay(timegap);
        }
        delay(1000);
    }

    digitalWrite(dirPin, LOW);
    for (int i = 0; i < 7; i++) {
        digitalWrite(stepPin, HIGH);
        delay(timegap);
        digitalWrite(stepPin, LOW);
        delay(timegap);
    }
    digitalWrite(slpPin, LOW);

    // Actual run of mode 1
    while (digitalRead(switchPin) == HIGH) { // Wait for switch to be depressed
        delay(1);
    }

    while (digitalRead(switchPin) == LOW) { // Wait for switch release (flow start)
        delay(1);
    }

    digitalWrite(slpPin, HIGH);
    digitalWrite(dirPin, HIGH);

    for (int i=0; i < 7; i++) {
        digitalWrite(stepPin, HIGH);
        delay(timegap);
        digitalWrite(stepPin, LOW);
        delay(timegap);
    }

}

delay(10);

digitalWrite(dirPin, LOW);

for (int i = 0; i < 7; i++) {
  digitalWrite(stepPin, HIGH);
  delay(timegap);
  digitalWrite(stepPin, LOW);
  delay(timegap);
}

delay(200);

digitalWrite(slpPin,LOW);

while (digitalRead(switchPin) ==HIGH) {
  digitalWrite(LED,HIGH);
  delay(300);
  digitalWrite(LED,LOW);
  delay(300);
  delay(500);
  finish = 1;
}

// Mode 2 Coding - Setup for 5.4 degrees of fin actuation
if (j == 1) {
  // Run test of mode 2
  digitalWrite(slpPin,HIGH);
  digitalWrite(dirPin, HIGH);

  for (int i=0; i < 3; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
  }

  delay(1000);

  digitalWrite(dirPin, LOW);

  for (int i = 0; i < 3; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
  }
}
digitalWrite(slpPin, LOW);

// Actual run of mode 2
while(digitalRead(switchPin) == HIGH) {  // Wait for switch to be depressed
    delay(1);
}

while(digitalRead(switchPin) == LOW) {  // Wait for switch release
    delay(1);
}

digitalWrite(slpPin, HIGH);
digitalWrite(dirPin, HIGH);

for (int i=0; i < 3; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}

delay(10);
digitalWrite(dirPin, LOW);

for (int i = 0; i < 3; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}

delay(200);
digitalWrite(slpPin,LOW);

while (digitalRead(switchPin) == HIGH) {
    digitalWrite(LED,HIGH);
    delay(300);
    digitalWrite(LED,LOW);
    delay(300);
}

delay(500);

finish = 1;
}

// Mode 3 Coding - Setup for 90 degrees of actuation
if (j == 2) {

// Run test of mode 3
digitalWrite(slpPin,HIGH);
digitalWrite(dirPin, HIGH);

for (int i=0; i < 50; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}

delay(1000);
digitalWrite(dirPin, LOW);

for (int i = 0; i < 50; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}

digitalWrite(slpPin,LOW);
// Actual run of mode 3
while(digitalRead(switchPin) ==HIGH) { // Wait for switch to be depressed
    delay(1);
}

while(digitalRead(switchPin) ==LOW) { // Wait for switch release (flow start)
    delay(1);
}

digitalWrite(slpPin,HIGH);
digitalWrite(dirPin, HIGH);

for (int i=0; i < 50; i++) {
    digitalWrite(stepPin, HIGH);
    delay(timegap);
    digitalWrite(stepPin, LOW);
    delay(timegap);
}

delay(1000);
digitalWrite(dirPin, LOW);

for (int i = 0; i < 50; i++) {
    digitalWrite(stepPin, HIGH);
delay(timegap);
digitalWrite(stepPin, LOW);
delay(timegap);
}
delay(200);
digitalWrite(slpPin, LOW);
while (digitalRead(switchPin) == HIGH) {
digitalWrite(LED, HIGH);
delay(300);
digitalWrite(LED, LOW);
delay(300);

delay(500);

finish = 1;
}
delay(1);
}
Appendix G – Model Drawings

G.1 Appendix Outline
The drawings contained within this Appendix are of all revisions of each model component and assembly drawings. A description of each is provided in Table G.1. Table G.2 outlines which part revisions were used for each run.

Table G.1: Drawing numbers and descriptions

<table>
<thead>
<tr>
<th>Drawing Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENG4111-001</td>
<td>Exploded Assembly</td>
</tr>
<tr>
<td>ENG4111-002</td>
<td>Assembly Detail</td>
</tr>
<tr>
<td>ENG4111-010</td>
<td>Nose Section Detail</td>
</tr>
<tr>
<td>ENG4111-011</td>
<td>Body Section Detail</td>
</tr>
<tr>
<td>ENG4111-012</td>
<td>Control Fin Detail</td>
</tr>
<tr>
<td>ENG4111-013</td>
<td>Body End Cap Detail</td>
</tr>
<tr>
<td>ENG4111-100</td>
<td>Support Stand Layout</td>
</tr>
<tr>
<td>ENG4111-101</td>
<td>Support Stand Detail</td>
</tr>
</tbody>
</table>

Table G.2: Drawing revisions used for each run

<table>
<thead>
<tr>
<th>Drawing Number</th>
<th>Run Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RUN283</td>
</tr>
<tr>
<td>ENG4111-001</td>
<td>Rev 0</td>
</tr>
<tr>
<td>ENG4111-002</td>
<td>Rev 0</td>
</tr>
<tr>
<td>ENG4111-010</td>
<td>Rev 0</td>
</tr>
<tr>
<td>ENG4111-011</td>
<td>Rev 0</td>
</tr>
<tr>
<td>ENG4111-012</td>
<td>Rev 0</td>
</tr>
<tr>
<td>ENG4111-013</td>
<td>Rev 0</td>
</tr>
<tr>
<td>ENG4111-100</td>
<td>Rev 0</td>
</tr>
<tr>
<td>ENG4111-101</td>
<td>Rev 0</td>
</tr>
<tr>
<td>ITEM</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>NOSE CONE</td>
</tr>
<tr>
<td>2</td>
<td>BRASS TUBE 2.3OD X 1.5ID X 23</td>
</tr>
<tr>
<td>3</td>
<td>MICROSWITCH</td>
</tr>
<tr>
<td>4</td>
<td>CR2 RECHARGEABLE BATTERY</td>
</tr>
<tr>
<td>5</td>
<td>MICROCONTROLLER &amp; STEPPER DRIVER</td>
</tr>
<tr>
<td>6</td>
<td>REAR BODY SECTION</td>
</tr>
<tr>
<td>7</td>
<td>CONTROL SURFACE PIN</td>
</tr>
<tr>
<td>8</td>
<td>16 TOOTH BEVEL GEAR</td>
</tr>
<tr>
<td>9</td>
<td>SHAFT 2X60</td>
</tr>
<tr>
<td>10</td>
<td>15MM STEPPER MOTOR</td>
</tr>
<tr>
<td>11</td>
<td>1:10 REDUCTION GEARBOX</td>
</tr>
<tr>
<td>12</td>
<td>END CAP &amp; MOTOR HOUSING</td>
</tr>
<tr>
<td>13</td>
<td>STEEL PIN 1X6</td>
</tr>
<tr>
<td>14</td>
<td>STEEL PIN 1X25.4</td>
</tr>
<tr>
<td>15</td>
<td>STEEL PIN 1X21</td>
</tr>
</tbody>
</table>
BAR PREVENTS SLIDING ROD FROM PULLING THROUGH

ROD CAN SLIDE BACK INSIDE SLEEVE

DRAG DISC SITS PERPENDICULAR TO FLOW & IS ATTACHED TO SLIDING ROD

COTTON STRING TIES TO BAR

STAND

SLOTS IN STAND ALLOW FOR ADJUSTMENT OF MODEL INITIAL AOA

FIXED SLEEVE
Appendix H – Raw Experimental Data

H.1 Appendix Outline
There is far too many images from the raw high-speed camera to be included here. Instead the data which is determined via the post processing analysis of the high-speed camera footage is presented in its raw tabulated form. In the case of an image showing interesting or notable results it is included in the body of the report.

H.2 RUN283 Data

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Frame</th>
<th>Body AoA, Alpha (degrees)</th>
<th>Fin Angle to Body, Beta (degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>215</td>
<td>0 (reference)</td>
<td>0 (reference)</td>
</tr>
<tr>
<td>1.33</td>
<td>217</td>
<td>-0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>2.67</td>
<td>219</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>221</td>
<td>-1</td>
<td>2</td>
</tr>
<tr>
<td>5.33</td>
<td>223</td>
<td>-1</td>
<td>4</td>
</tr>
<tr>
<td>6.67</td>
<td>225</td>
<td>-1</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>227</td>
<td>-2</td>
<td>5.5</td>
</tr>
<tr>
<td>9.33</td>
<td>229</td>
<td>-2</td>
<td>6.5</td>
</tr>
<tr>
<td>10.67</td>
<td>231</td>
<td>-2</td>
<td>5.5</td>
</tr>
<tr>
<td>12</td>
<td>233</td>
<td>-2</td>
<td>7.5</td>
</tr>
<tr>
<td>13.33</td>
<td>235</td>
<td>-2</td>
<td>9.5</td>
</tr>
<tr>
<td>14.67</td>
<td>237</td>
<td>-1.5</td>
<td>9.5</td>
</tr>
<tr>
<td>16</td>
<td>239</td>
<td>-1</td>
<td>9.5</td>
</tr>
<tr>
<td>17.33</td>
<td>241</td>
<td>-0.5</td>
<td>9.5</td>
</tr>
<tr>
<td>18.67</td>
<td>243</td>
<td>0</td>
<td>9.5</td>
</tr>
<tr>
<td>20</td>
<td>245</td>
<td>1</td>
<td>9.5</td>
</tr>
<tr>
<td>21.33</td>
<td>247</td>
<td>1.5</td>
<td>9.5</td>
</tr>
<tr>
<td>22.67</td>
<td>249</td>
<td>2.5</td>
<td>9</td>
</tr>
<tr>
<td>24</td>
<td>251</td>
<td>3</td>
<td>8.5</td>
</tr>
<tr>
<td>25.33</td>
<td>253</td>
<td>4</td>
<td>8.5</td>
</tr>
<tr>
<td>26.67</td>
<td>255</td>
<td>4.5</td>
<td>8.5</td>
</tr>
<tr>
<td>28</td>
<td>257</td>
<td>5.5</td>
<td>8.5</td>
</tr>
<tr>
<td>29.33</td>
<td>259</td>
<td>6.5</td>
<td>8.5</td>
</tr>
<tr>
<td>30.67</td>
<td>261</td>
<td>7</td>
<td>8.5</td>
</tr>
<tr>
<td>32</td>
<td>263</td>
<td>8</td>
<td>5.5</td>
</tr>
<tr>
<td>33.33</td>
<td>265</td>
<td>9</td>
<td>4.5</td>
</tr>
<tr>
<td>34.67</td>
<td>267</td>
<td>10</td>
<td>3.5</td>
</tr>
<tr>
<td>36</td>
<td>269</td>
<td>10</td>
<td>1.5</td>
</tr>
<tr>
<td>37.33</td>
<td>271</td>
<td>10</td>
<td>0.5</td>
</tr>
<tr>
<td>38.67</td>
<td>273</td>
<td>10.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>
## H.3 RUN286 Data

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Frame</th>
<th>Body AoA, Alpha (degrees)</th>
<th>Fin Angle to Body, Beta (degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>386</td>
<td>0 (reference)</td>
<td>0 (reference)</td>
</tr>
<tr>
<td>1.33</td>
<td>390</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.67</td>
<td>394</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>398</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5.33</td>
<td>402</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6.67</td>
<td>406</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>410</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9.33</td>
<td>414</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10.67</td>
<td>418</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>422</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13.33</td>
<td>426</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14.67</td>
<td>430</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>434</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>17.33</td>
<td>438</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>18.67</td>
<td>442</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>446</td>
<td>1.5</td>
<td>0</td>
</tr>
<tr>
<td>21.33</td>
<td>450</td>
<td>2.0</td>
<td>0</td>
</tr>
<tr>
<td>22.67</td>
<td>454</td>
<td>2.0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>458</td>
<td>2.0</td>
<td>0</td>
</tr>
<tr>
<td>25.33</td>
<td>462</td>
<td>2.5</td>
<td>0</td>
</tr>
<tr>
<td>26.67</td>
<td>466</td>
<td>3.0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>470</td>
<td>3.5</td>
<td>0</td>
</tr>
<tr>
<td>29.33</td>
<td>474</td>
<td>3.5</td>
<td>0</td>
</tr>
<tr>
<td>30.67</td>
<td>478</td>
<td>4.0</td>
<td>-0.5</td>
</tr>
<tr>
<td>32</td>
<td>482</td>
<td>4.0</td>
<td>-1.0</td>
</tr>
<tr>
<td>33.33</td>
<td>486</td>
<td>4.0</td>
<td>-1.5</td>
</tr>
<tr>
<td>34.67</td>
<td>490</td>
<td>3.5</td>
<td>-1.5</td>
</tr>
<tr>
<td>36</td>
<td>494</td>
<td>3.5</td>
<td>-1.5</td>
</tr>
<tr>
<td>37.33</td>
<td>498</td>
<td>3.5</td>
<td>-1.0</td>
</tr>
<tr>
<td>38.67</td>
<td>502</td>
<td>3.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>40</td>
<td>506</td>
<td>3.0</td>
<td>-0.5</td>
</tr>
<tr>
<td>41.33</td>
<td>510</td>
<td>2.0</td>
<td>-0.5</td>
</tr>
<tr>
<td>42.67</td>
<td>514</td>
<td>1.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>44</td>
<td>518</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>45.33</td>
<td>522</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>46.67</td>
<td>526</td>
<td>-0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>48</td>
<td>530</td>
<td>-1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>49.33</td>
<td>534</td>
<td>-1.5</td>
<td>2.0</td>
</tr>
<tr>
<td>50.67</td>
<td>538</td>
<td>-2.0</td>
<td>2.5</td>
</tr>
<tr>
<td>52</td>
<td>542</td>
<td>-2.5</td>
<td>3.0</td>
</tr>
<tr>
<td>53.33</td>
<td>546</td>
<td>-2.5</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>54.67</td>
<td>550</td>
<td>-3.0</td>
<td>4.5</td>
</tr>
<tr>
<td>56</td>
<td>554</td>
<td>-3.5</td>
<td>4.5</td>
</tr>
<tr>
<td>57.33</td>
<td>558</td>
<td>-4.0</td>
<td>4.5</td>
</tr>
<tr>
<td>58.67</td>
<td>562</td>
<td>-4.0</td>
<td>5.0</td>
</tr>
<tr>
<td>60</td>
<td>566</td>
<td>-4.5</td>
<td>5.0</td>
</tr>
<tr>
<td>61.33</td>
<td>570</td>
<td>-4.0</td>
<td>5.5</td>
</tr>
<tr>
<td>62.67</td>
<td>574</td>
<td>-3.5</td>
<td>5.5</td>
</tr>
<tr>
<td>64</td>
<td>578</td>
<td>-3.0</td>
<td>5.5</td>
</tr>
<tr>
<td>65.33</td>
<td>582</td>
<td>-2.5</td>
<td>6.5</td>
</tr>
<tr>
<td>66.67</td>
<td>586</td>
<td>-1.5</td>
<td>6.5</td>
</tr>
<tr>
<td>68</td>
<td>590</td>
<td>-0.5</td>
<td>6.5</td>
</tr>
<tr>
<td>69.33</td>
<td>594</td>
<td>1.0</td>
<td>6.5</td>
</tr>
<tr>
<td>70.67</td>
<td>598</td>
<td>2.5</td>
<td>7.0</td>
</tr>
<tr>
<td>72</td>
<td>602</td>
<td>4.0</td>
<td>6.5</td>
</tr>
<tr>
<td>73.33</td>
<td>606</td>
<td>5.0</td>
<td>7.0</td>
</tr>
<tr>
<td>74.67</td>
<td>610</td>
<td>6.5</td>
<td>7.0</td>
</tr>
<tr>
<td>76</td>
<td>614</td>
<td>7.5</td>
<td>7.0</td>
</tr>
<tr>
<td>77.33</td>
<td>618</td>
<td>9.0</td>
<td>7.0</td>
</tr>
<tr>
<td>78.67</td>
<td>622</td>
<td>10.5</td>
<td>7.0</td>
</tr>
<tr>
<td>80</td>
<td>626</td>
<td>12.0</td>
<td>7.0</td>
</tr>
<tr>
<td>81.33</td>
<td>630</td>
<td>14.0</td>
<td>7.0</td>
</tr>
<tr>
<td>82.67</td>
<td>634</td>
<td>15.5</td>
<td>7.0</td>
</tr>
<tr>
<td>84</td>
<td>638</td>
<td>16.0</td>
<td>5.5</td>
</tr>
<tr>
<td>85.33</td>
<td>642</td>
<td>17.5</td>
<td>5.0</td>
</tr>
<tr>
<td>86.67</td>
<td>646</td>
<td>19.5</td>
<td>4.0</td>
</tr>
<tr>
<td>88</td>
<td>650</td>
<td>18.0</td>
<td>2.0</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>137</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>