Modelling and Simulation of Small Scale Fixed-Wing Autonomous Aerial Vehicles

KHAN, Mahmud Safat

Modelling and Simulation of Small Scale Fixed-Wing Autonomous Aerial Vehicles

Mahmud Safat Khan

A thesis submitted in partial fulfilment of the requirements of
Sheffield Hallam University
for the degree of Doctor of Philosophy

October 2021

Candidate Declaration

I hereby declare that:

1. I have not been enrolled for another award of the University, or other academic or professional organisation, whilst undertaking my research degree.

2. None of the material contained in the thesis has been used in any other submission for an academic award.

3. I am aware of and understand the University's policy on plagiarism and certify that this thesis is my own work. The use of all published or other sources of material consulted have been properly and fully acknowledged.

4. The work undertaken towards the thesis has been conducted in accordance with the SHU Principles of Integrity in Research and the SHU Research Ethics Policy.

5. The word count of the thesis is 41519.

Name	*Mahmud Safat Khan*
Date	*October 2021*
Award	*PhD*
Faculty	*Business, Technology and Engineering*
Director(s) of Studies / Supervisor	*Dr Jonathan Potts / Dr Lyuba Alboul*

Acknowledgements

I would like to start by expressing my thanks and gratitude to my parents, Abul Hashem Khan & Farida Yeasmin, who have devoted their lives to making sure every opportunity to do better in life is within my reach. They have dedicated their lives to being my real-life superheroes, shielding me from hardship in life and sacrificing for my sake. Thank you, Abbu and Ammu, for your never-ending love and support. Without all your contribution, my research work wouldn’t have been possible.

Thank you to my wife, Rebekah. Her round the clock dedication to helping me believe in myself, taking care of my mental and physical wellbeing, made it possible for me to keep going through the toughest parts of this journey. She has sacrificed her money, time and wellbeing for years to ensure I remain capable of delivering on my goals. Thank you, Rebekah, for never giving up on me. Thank you for every occasion you help put me back together with your love and support. Without you, I’d not have been able to finish this journey.

I would also like to acknowledge the role my friend, Stoyan Kostadinov, played in ensuring I see myself to the end of this project while not feeling alone. Thank you, Stoyan, for making my time in Sheffield so much better with all the conversations, debates, consultations, advice and so much more. It’s my privilege to have such a selfless and caring person as a friend.

I’d like to express my gratitude to my Director of Studies, Dr Jonathan Potts and my research supervisor, Dr Lyuba Alboul. Thank you for all the help and support you guys provided throughout the years to ensure my success. Thank you for your confidence in me. Thank you for all the personal and professional help and guidance. Thank you for all the hours you’ve put into me and my work, even on weekends and vacations, to help and motivate me to do the best work possible.

Thanks to the Student Support & Wellbeing staff at SHU. Thank you for ensuring that all necessary accommodations have been made such that my health issues do not stop me from progressing through the stages of my research journey.

Thanks also to all the authors, teachers, researchers, software developers, free/open-source community around the world, who have shared your tools and knowledge, making it possible for me to conduct this research.

Lastly, thanks to all my friends and family around the globe. Thank you for believing in me and cheering me on.

Abstract

Although various tools (e.g., Wind tunnels, Professional-grade CFD packages, Industrial Grade Flight Simulators, etc.) for aid in the development of fixed-wing UAV/UASs exists, the associated high costs and skills requirements stunt adequate development work for many students, researchers, and even prospective businesses. The aim of this thesis, therefore, is to present investigations of effective modelling techniques, improvement of simulation fidelity, exploration and demonstration of alternative simulation and modelling approaches. The approach taken in conducting all the relevant work factored in the use of free and openly available tools and code to provide solutions for higher fidelity simulation and modelling without reliance on the traditional expensive methods. MATLAB/Simulink has been utilized to develop a 12 state, 6 degrees of freedom simulation. A scaled down Remote Control model aircraft (FMS SkyTrainer 182) and a widely used commercial off-the-shelf (COTS) simulator (X-Plane) was used to demonstrate alternative modelling techniques. Techniques to improve X-Plane's modelling and simulation fidelity for small scale UAVs are demonstrated. Processes for improving aerodynamic lift-modelling inputs for the MATLAB/Simulink codes are outlined. Application of the free and open source XFLR5 program for supplying better aerodynamic/stability/control data for modelling is demonstrated. The work undertaken in all the aforementioned areas collectively establish an approach to simulation and modelling of small scale fixed-wing UAVs/UASs that improves simulation fidelity without the reliance on expensive facilities or software solutions.

Table of Contents

1 Introduction

From the moment the Wright brothers realised the long-overdue dream of mankind to fly, humanity has come up with countless mutations of the realisation of that dream. Not only did mankind learn how to make a craft take flight like a bird, but we also learned a lot of different ways to do so while simultaneously finding new reasons to do so. In that quest, we ended up designing crafts ranging from helicopters all the way to spacecraft. Besides learning to move ourselves from one point on this planet to another, with that knowledge, we managed to put man on the moon.

Despite it being seemingly counterintuitive, humanity has simultaneously dreamed of severing the bond between people and these flying machines. Humanity was quick to realise a range of advantages of flying machines that didn't require being manned. However, until the recent advances made in the electronics/computer industry, not much could be done about this dream. Moore's law came to the rescue of humanity, and finally the necessary hardware for such automation became available. This started the quest to develop small Unmanned Aerial Vehicles (UAV)/Unmanned Aerial Systems (UAS), leading to the present day where we discovered a wide range of applications ranging from hobbyists flying them recreationally to military and scientific research applications.

As the global market for UAVs keeps growing, the demand for better and more improved systems is growing too. One efficient way to meet that demand is through the optimisation of existing technology to synthesise a new one. While this approach has resulted in wide scale development for multi-rotor UAVs, the same doesn't hold true for fixed-wing designs. Despite being efficient fliers that are capable of outclassing multi-rotor designs in range, endurance and payload capacity, etc., their development and adoption has suffered through a range of complications that require the attention of researchers in the field. While the development cycle of regular fixed-wing aircraft is well studied and understood, much of the methods and tools utilised creates problems with accessibility. That requires the identification of areas where such improvements are possible.

1.1 Context

Despite considerable progress made in the autonomous system development in modern times, there is an obvious asymmetry in the adoption of fixed-wing designs. While multi-rotor designs have been adopted widely and in large numbers by various interest groups (recreational, commercial, scientific community, military, etc.), the same cannot be stated for fixed-wing designs. Even for the academic/research focused demographics, who are in a relatively better position for development work, it is significantly harder to work on fixed-wing designs. The following are some major challenges:

- Problems with Flight Tests.
- Control Systems implementation problem.
- Lack of much needed available data on various Aerodynamic and Stability/Control parameters.
- Inapplicability of existing research methods/procedures to small autonomous platforms.
- High demand on developer skills due to interdisciplinary nature of the systems.
- High demand on financial resources due to the expensive nature of various traditional tools and methods in the Aerospace industry.

The core problem can be better appreciated by considering the difference in the ease of testing of both fixed-wing and multi-rotor designs. In principle, a multi-rotor drone can be flown on a researcher’s desktop if needed. Safety harnesses can be attached to the system as various system configurations are being tested before any outdoor fully autonomous flight tests are considered. The situation for fixed-wing designs is dramatically different. Unlike multi-rotors, fixed-wing designs generally need to be in a constant state of motion while maintaining certain orientations in space to avoid a stall potentially resulting in the destruction of the platform.

The flying characteristics will depend on the aerodynamic behaviour of the aircraft which is not known prior to the test flights and data collection. Even if an attempt is made to use traditional test-flight procedures to obtain this data, the lack of skilled test pilots combined with the hardship of carrying out tests designed for human-controlled aircraft with typical control methods adds to higher risk of potential failure. These problems are exacerbated by the typical servo-actuated control surface movements that these fixed-wings rely on. There may be no practical way of providing all the desired control inputs during the test flight should the control authority be outside of the input range available. The consideration of environmental factors (e.g. wind) which affect the dynamic behaviour of small-scale UAVs much more than full-scale human-piloted aircraft only serves to complicate the process further.

Traditional experimental means and procedures for estimating much needed aerodynamic data (e.g. Wind Tunnels) are both expensive and inaccessible for many. Not only are these resources difficult to access and utilise by prospective researchers in the developed world, they are often unavailable entirely in many parts of the world that are economically less advanced. There are computational fluid dynamics (CFD) simulation methods, as an alternative but this is both resource and skills development intensive.

The development of such aircraft, therefore, requires a highly inter-disciplinary approach. As a result, specialisation in every relevant domain is not a reasonable presupposition. This means that while it is possible to explore relevant literature to develop a research methodology, it may ultimately prove unfeasible for many potential researchers and developers. While the development of more accurate and sophisticated tools and methods serves to aid in the work undertaken by existing researchers, it does little to recruit more minds from a wide range of fields of interests to contribute to the overall development and utilisation of such fixed-wing platforms.

While there exists various means of modelling and simulation methods which can help address these problems greatly, these models and simulations themselves rely on advance knowledge of the previously mentioned aircraft parameters for good fidelity. All these aforementioned challenges require the exploration and development of methodological processes that are better suited for small autonomous fixed-wing aircraft. It creates a demand to find alternatives to the traditional approaches that can address these problems in ways that do not reduce accessibility.

1.2 Aims and Objectives

The primary aim of this research is to explore and develop accessible methods for modelling and simulation of fixed-wing UAVs. The success of this study relies on the following objectives that need to be satisfied:

- Comprehensive literature review on modelling & simulation and analytical/computational methods for obtaining aerodynamic, stability & control parameters of fixed-wing aircraft.
- Utilise common and widely available software and hardware tools for simulation/modelling work with an emphasis on free/open-source solutions.
- Explore and appraise independent simulation and modelling tools.
- Identify the limitations of each tool/method and develop improved solutions to overcome any transformation, formatting or integration issues.
- Construct multiple and independent models/simulations of a target small-scale, fixed-wing airframe.
- Conduct a range of appropriate tests on the models to aid in the creation of aerodynamic and stability/control datasets for the target airframe.
- Integrate the discrete and distinct tools, methods and techniques into a methodology that can be followed to convert readily available small-scale fixed-wing aircraft into high fidelity models/simulations.
- Test fly the physical fixed-wing target airframe to obtain validation data to tune the simulation models.

1.3 Contribution of the work

This work develops a more accessible and systematic way of modelling and simulating small-scale fixed-wing Autonomous Aerial Vehicles. Traditional means of aerospace modelling & simulation are reviewed and the challenges of applying them to such small-scale vehicles are highlighted. Existent alternative solutions are reviewed for their limitations and advantages. A methodical approach, emphasising the use of simpler tools and techniques, free/Open-source software, custom code, common and widely used software platforms, etc., is proposed to address the aims and objectives of this research work.

The challenge of unavailable geometric/inertial modelling data for such readily available small-scale airframes has been addressed with a systematic measurement approach relying on the use of simple and accessible tools and techniques. This research shows that the techniques applied do not have any significant impact on the subsequent important modelling and simulation work. The unavailability of essential low Reynolds number aerodynamic data has been tackled with an approach involving the utilisation of free/open-source software (e.g. XFLR5) and post-processing techniques leading to the creation of useful aerodynamic datasets. The data obtained proved useful for stability/control analysis which enables better control system design. This way of obtaining the data also permits flow visualisations informing performance/safety analysis which replaces the requirement for a wind tunnel testing phase. The systematic process of starting with the measurements of the small-scale airframe and ending with invaluable aerodynamic/stability & control/performance, is documented and demonstrated within this research in detail.

Challenges regarding the lack of integration/documentation of various common simulation platforms are addressed in this research. The application of these tools, for accurate modelling & simulation of the flight dynamics and performance of such small-scale airframes is not well documented in the published literature. This research shows that by following the specific steps taken to address the identified limitations, it is indeed possible to transform a popular simulation platform for full-scale aircraft (e.g. X-Plane) into a useful platform for high fidelity small-scale fixed-wing vehicle development. With further development and integration of freely available parameter estimation code, it is shown that conventional flight tests can be conducted on this high-fidelity model, yielding key stability performance characteristics of the small-scale aircraft prior to any high-risk airframe hardware flight test. Part of this work also led to the validation of freely available software tools for performance prediction of typical power plant and propeller combinations.

The aforementioned analysis and data have been utilised to demonstrate the development of a 12 state 6 degree of freedom MATLAB/Simulink simulation of a target airframe, leveraging freely available templates and their extension. In this research, a systematic process of integrating the aerodynamic and stability/control characteristics of a small-scale fixed-wing aircraft, into a widely used simulation platform (e.g. MATLAB/Simulink) is demonstrated via the use of simple and accessible software packages and associated custom coding. The difficulty of representing the differing aerodynamic behaviour of random airframes, requiring their unique modelling parameters, has been resolved. A method has been developed to allow for aerodynamic data of any target airframe to be easily ported into the developed MATLAB/Simulink environment. The challenges of accurate visualisation of complex airframes and its impact on the simulation architecture and/or cost have been resolved. Alternative techniques have been developed to overcome the challenge of complex airframe visualisation without the use of the existing resource intensive simulation visualisation tools. This has been successfully tested with the developed simulation environment, demonstrating full functionality, and thus eliminating the need for complex/expensive simulation architecture.

The suite of simulation/modelling tools and techniques outlined in this thesis collectively comprise a comprehensive methodology for the modelling & simulation of small-scale fixed-wing Autonomous Aerial Vehicles. Furthermore, this method addresses the challenge of accessible and low-cost solutions to develop simulation/modelling tools that are freely available to be used by researchers operating both professionally and recreationally. At the time of writing, there exists no such published methodology.

1.4 Publication

Khan, M., Alboul, L. and Potts, J., 2020, May. Development of Tools and Methods for Autonomous Fixed-wing UAV Research. In UKRAS20 Conference: “Robots into the real world” Proceedings (pp. 69-70). EPSRC UK-RAS Network.

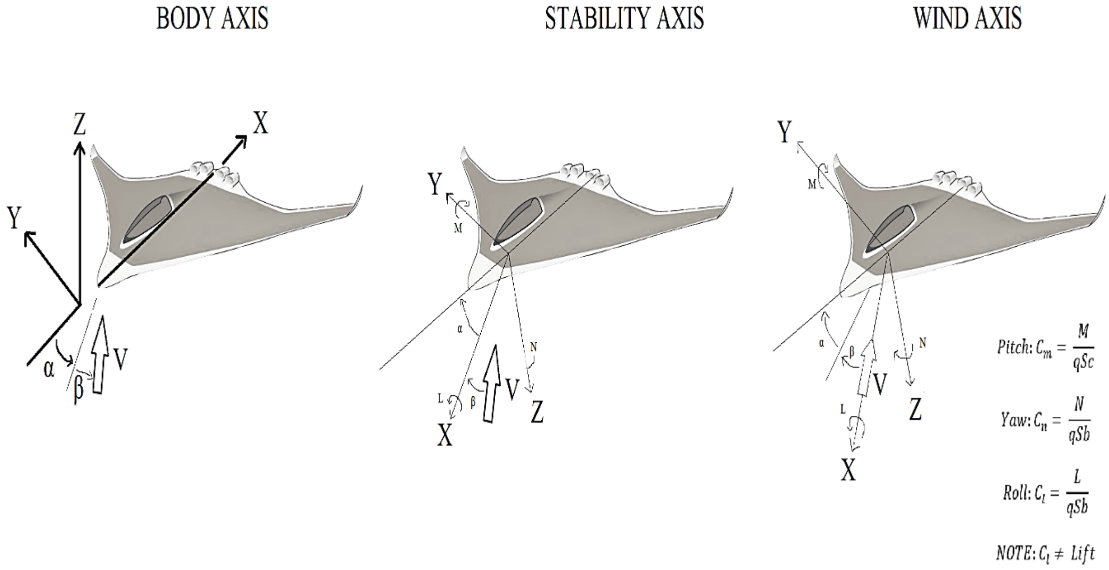
2 Literature Review

2.1 Fixed-Wing UAV Dynamic Modelling

In principle, the stability and control of fixed-wing UAV/UAS platforms are no different than that of the manned aircrafts. The difference primarily lies in the way the principles are implemented. Considering the fact that a UAS platform may even be fully autonomous, without any human guidance, the stability and control issues then become perhaps the most important aspect of the UAS. Any progress in this matter necessitates grasping the dynamic behaviour of the UAS platform. In the following subsections, known and effective methods of derivation and implementation of such models from the literature are explored.

2.1.1 Coordinate Systems

The position and orientation of the aircraft needs to be represented within the simulation unambiguously. Any attempt at such localisation and description of state highlights the importance of reference frames and methods of conversion between two different frames.



BODY AXIS
STABILITY AXIS
WIND AXIS
Z
X
Y
α
β
V
Y
M
α
N
L
β
V
X
Z
Y
M
α
β
N
L
V
Z
X
Pitch: $C_m = \frac{M}{qSc}$
Yaw: $C_n = \frac{N}{qSb}$
Roll: $C_l = \frac{L}{qSb}$
NOTE: $C_l \neq Lift$

Figure 2.1 Three traditional axis systems (based on Raymer et al. [1][2])

Although there are no issues with using custom coordinates systems, the traditionally used systems, such as in Figure 2.1, have proven to be useful in many different contexts. Virtually all mathematical tools presented throughout the literature that's at our disposal will end up referring to one of these traditional systems. There are three most common axis-systems used for aircraft analysis [1]:

Types	Details
Body Axis System	The body axis system is perhaps the most intuitive one. The X axis runs along the fuselage while the Z axis points upwards. Lift being generated perpendicular to the wing direction results in inducing problems with this system [1]. That's due to the lift/drag direction variation with the angle of attack of the aircraft.
Wind Axis System	This system tries to resolve the shortcoming of the Body Axis System by orienting the X axis with the direction of the relative wind. One inherent problem of the system is that yawing makes the airplane unsymmetrical about the X-Z plane [1].
Stability Axis System	This system is a compromise between the former two where the symmetry is preserved by not making the X axis offset to the yaw angle despite the same alignment as the Wind Axis System.

Table 2-1 Comparison of traditional axis-systems

The calculations for the aerodynamic forces and moments acting on the aircraft are done relative to a frame of reference. The motion of the aircraft itself, based on Newtonian equations, are captured relative to a frame of reference. Any on board sensors such as GPS, Airspeed Sensors, etc. operate within their own reference frames. Any higher level control inputs (flight course, loiter, etc.) will presuppose its own reference frame. Due to the types of shortcomings pointed out in the above table, no one coordinate frame is used as the standard basis for all calculations. Instead, whenever information is to be shared between modules of the simulation with different reference frames, matrix rotation and translation is utilised to convert from one frame to the other. This approach resolves the issues arising from inconsistent reference frames while exploiting MATLAB/Simulink’s ease with matrix manipulation.

2.1.2 Rotation Matrices

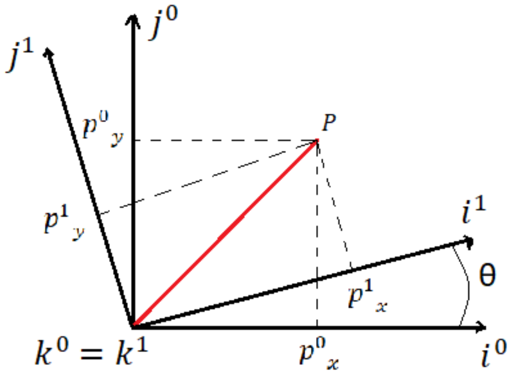


Figure 2.2 Rotation of frames based on [3][4]

The two different frames (Frame 1 and Frame 2) depicted in Figure 2.2 are considered for the derivation of rotation matrices [3][4]. Frame 1 is defined by i^0, j^0 and k^0 and Frame 2 is defined by i^1, j^1 and k^1. The point P, which is a vector depicted by the red line, can be represented in either frames as depicted in Figure 2.2. The relationship between these unit basis vectors (i^0, j^0,k^0 , i^1, j^1 and k^1) can be represented through rotation matrices. For example, we may derive the value of p^1 as follows [3]

$$p^1 = R_0{}^1 p^0 \qquad \text{[Eq. 2.1]}$$

In the above equation (Eq. 2.1), R_0^1 stands for the following rotation matrix that takes us from Frame 1 to Frame 2:

$$R_0^1 \triangleq \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{[Eq. 2.2]}$$

The right-hand rotation about the y and x axis are represented in a similar fashion [3] as follows:

$$R_0^1 \triangleq \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \qquad \text{[Eq. 2.3]}$$

$$R_0^1 \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \qquad \text{[Eq. 2.4]}$$

These righ-hand rotation (of θ) matrices (Eq. 2.3-2.4) are utilised to keep the vector P constant while shifting to a new coordinate frame. Rotation matrices can also be utilised to rotate vectors within a fixed frame of reference.

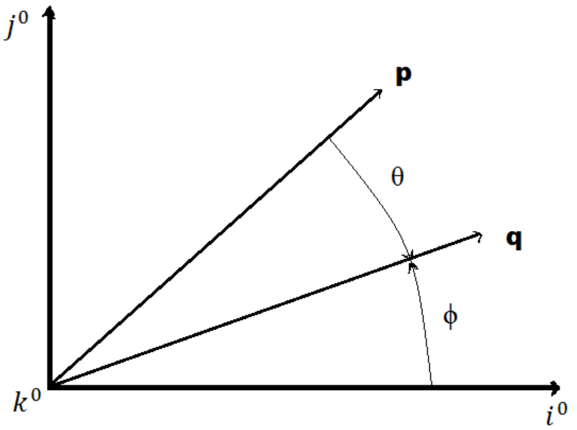


Figure 2.3 Rotation of vectors within a reference frame based on [3]

Similar determinations can be achieved via the left-hand rotation of a vector about an axis. For example, in Figure 2.3, the vector p can be rotated about the k^0 axis within the frame. The components of vectors p and q can be defined as follows[3][4]:

$$p = \begin{pmatrix} p\cos(\theta+\phi) \\ p\sin(\theta+\phi) \\ 0 \end{pmatrix} \quad \text{[Eq. 2.5]}$$

$$q = \begin{pmatrix} q\cos\phi \\ q\sin\phi \\ 0 \end{pmatrix} \quad \text{[Eq. 2.6]}$$

Based on the trigonometric identities [3], the vector p is expressed as follows:

$$p = \begin{pmatrix} p\cos\theta\cos\phi - p\sin\theta\sin\phi \\ p\sin\theta\cos\phi + p\cos\theta\sin\phi \\ 0 \end{pmatrix} \quad \text{[Eq. 2.7]}$$

The vector p (Eq. 2.7) can be obtained via a left-hand rotation of the vector q about the k^0 axis. This implies that it can express the relationship between the components of these two vectors by using the same rotation matrix (R_0^1) used to go from Frame 1 to Frame 2. But due to a rotation around the k^0 axis in the reverse direction, the transpose of the rotation matrix (Eq. 2.8) is used instead [3].

$$\left(R_0{}^1\right)^T \triangleq \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{[Eq. 2.8]}$$

This expresses the relationship between p and q as shown in Equation 2.9.

$$p = \left(R_0{}^1\right)^T q \quad \text{[Eq. 2.9]}$$

Therefore, from the vector p through a rotation of θ to the vector q or from the vector q to the vector p through a rotation of θ by the use of R_0^1 and $(R_0^1)^T$. The utility of it is captured within the simulation in the form of bi-directional rotation of the aircraft about every axis. Operationally, it can be expected to have misalighnments between the aircraft's frame, the direction of incoming wind and onboard sensors. All of that can have further misalignment of their respective frames with any map used for guidance and navigation. The complexity arising from that can be resolved via this technique of matrix rotation[3][4][5][6]. Such use of rotation matrix in the literature by Beard et al. allow for a good solution to the problems of reference frames encountered in the development of appropriate flight simulation modelling of fixed-wing platforms as detailed in the subsequent sections.

2.1.3 Reference Frames used for simulations

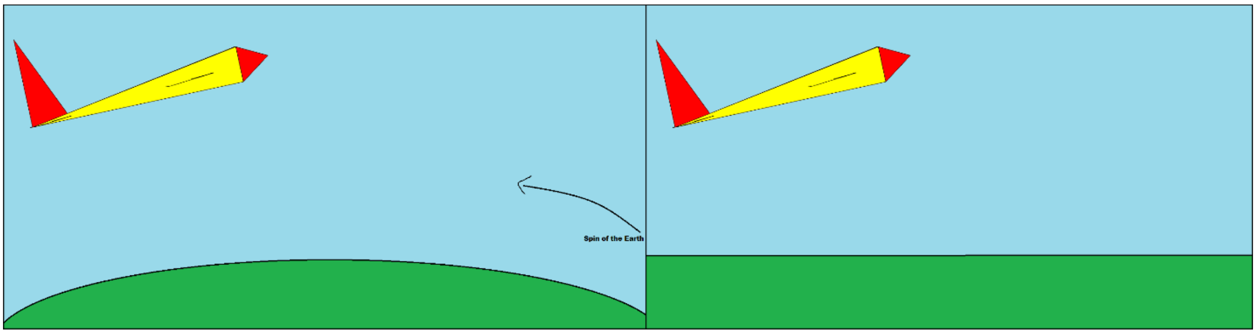


Figure 2.4 Depiction of curved earth and spinning earth (left) and flat and stationary earth (right)

For the sake of simplicity, certain assumptions are made before defining the Reference Frames used in the simulation[3]. Two such noteworthy assumptions are:

- Flat Earth: Assumption of zero curvature of the surface of the Earth over the operational range of the simulated aircraft.
- Stationary Earth: Assumption that the Earth itself is not spinning.

On top of reducing mathematical modelling complexity and computational resources required for the simulation, this frees up resources to be allocated to other factors that impact the overal accuracy and precision of any guidance, navigation and control task. In practice, engineering limitations can render many highly accurate mathematical models not only useless but a waste of resources. For example, proper modelling of the effect of wind [7]and accurate estimation of states that cannot be measured directly (eg. side-slip)[8][9] is going to impact the fidelity of the simulation to a disproportionately higher degree than the curvature of the Earth or it's spin. Considering engineering limitations, errors in measurement, errors in estimations and stochastic wind-related factors[10][11][12], it seems unjustified to devote resources to modelling factors whose impact on the overall mission is going to be relatively insignifacnt and buried within the noise of the other mentioned uncertainties.

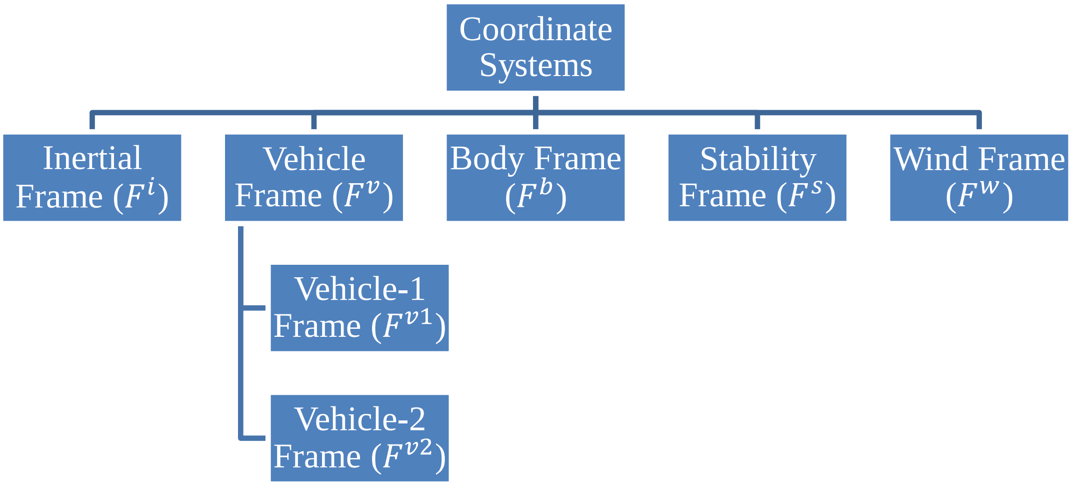


Figure 2.5 Tree depicting coordinate systems and their relationships

As illustrated in the above picture, a range of different coordinate systems have been considered from the literature [2][3][4][5] and utilised for the construction of the simulation. Via matrix rotation and translations, it is possible to derive and/or deduce one frame from another by following simple matrix operations (rotation & translation). This allows for easier handling of the operations within MATLAB/Simulink. Once the necessary rotation matrices are defined, they can be used as functions anywhere within the simulation.

2.1.3.1 Inertial Frame

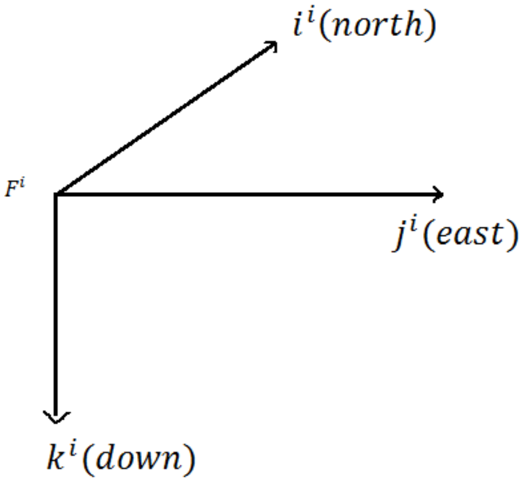


Figure 2.6 Inertial frame based on [3]

This is the coordinate system originating on a fixed point on earth with i^i alighned to the north and j^i aligned to the east directions. The k^i axis points vertically down into the theoratic center of the earth (down). The Inertial Frame (F^i) is the most intuitive and generally found as the most widely used coordinate system due to it's simplicity[3][4][5].

2.1.3.2 Vehicle Frame

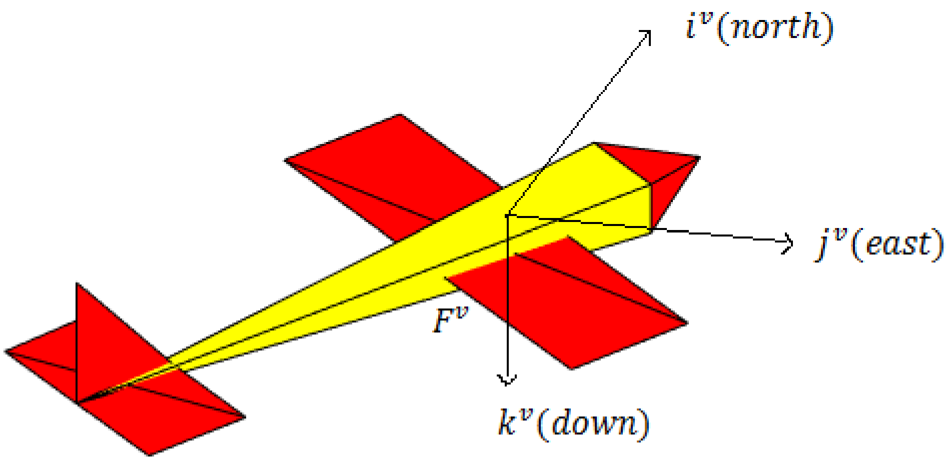


Figure 2.7 Vehicle Frame based on [3]

The Vehicle Frame (F^v) can be derived from the Inertial Frame(F^i) simply by placing the origin of the frame at the center of mass/center of gravity of the aircraft[[3][4][5]. However, for possible misalignment, it can be rotated to have the axis of the aircraft align with the axis of this frame [3]. Which is shown in the next two frames.

2.1.3.3 Vehicle-1 Frame

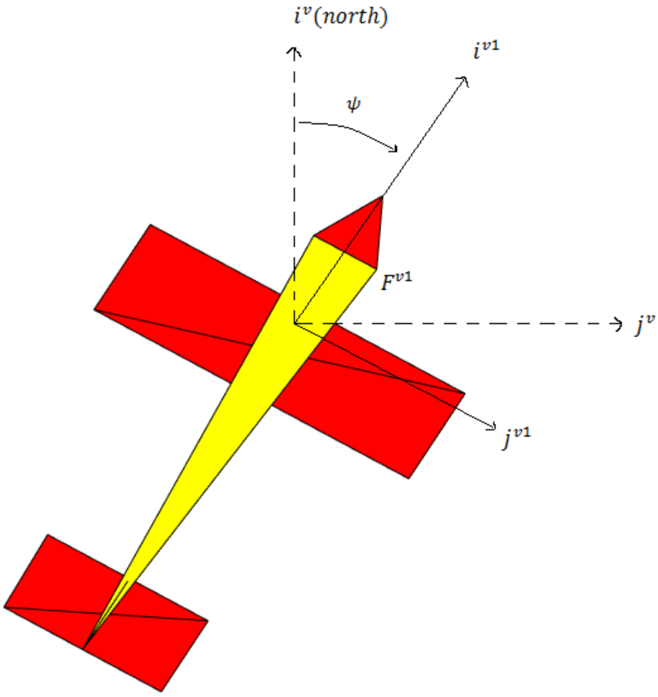


Figure 2.8 Vehicle-1 Frame based on [3]

Vehicle-1 Frame (F^{v1}) is deduced from the Vehicle Frame (F^v) by rotating the i^vaxis until it coincides with the heading (ψ) of the aircraft such that i^{v1}points out the nose of the aircraft while j^{v1} points out the starboard wing [3][4][5]. Since this rotation is about the vertical axis, k^{v1} remains unchanged from k^v except for it's label. This transformation is a right hand rotation and is done via the use of the same rotation matrix (Eq. 2.10) shown as follows:

$$R_v^{v1} = Righthand\ rotation\ from\ Vehicle\ to\ Vehicle - 1\ Frame \qquad \text{[Eq. 2.10]}$$

The notation format here are from Beard & McLain (2012) and use the subscript($v1$) to indicate the desired frame to rotate and the superscript(v) indicates the frame to which should be rotated [3]. Which in this case is a right hand rotation from the Vehicle Frame to the Vehicle-1 Frame. The magnitude of the rotation is the quantity ψ. As such the rotation matrix (Eq. 2.11) can be easily derived to be as follows.

$$R_v^{v1}(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{[Eq. 2.11]}$$

2.1.3.4 Vehicle-2 Frame

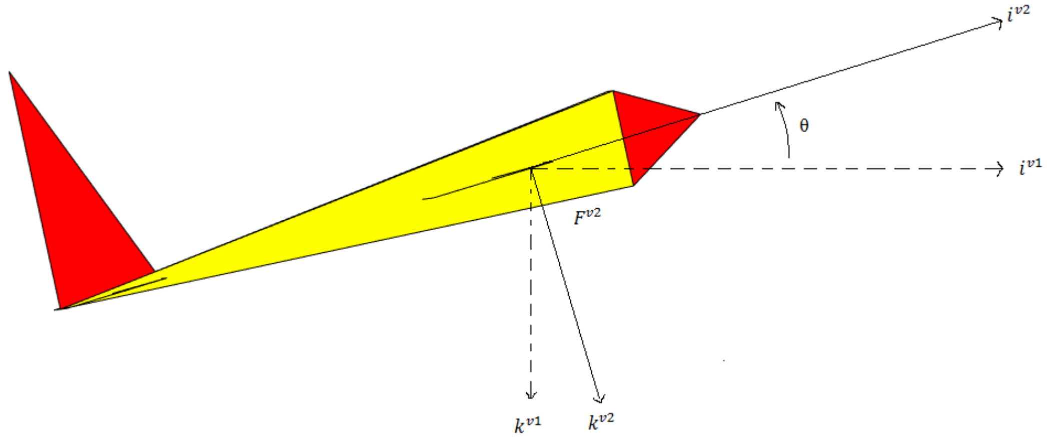


Figure 2.9 Vehicle-2 Frame based on [3]

The Vehicle-2 Frame (F^{v2}) is derived similarly from the Vehicle-1 Frame(F^{v1}) by rotating F^{v1} by the pitch angle (θ) [3]. This rotation takes place about the j^{v1}axis such that j^{v2} still coincides with j^{v1}after the rotation and continues to point out the starboard wing as before [3][4][5]. For this case, a similar approach as before and use the rotation matrix in Equation 2.12.

$$R_{v1}^{v2}(\psi) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \qquad \text{[Eq. 2.12]}$$

2.1.3.5 Body Frame

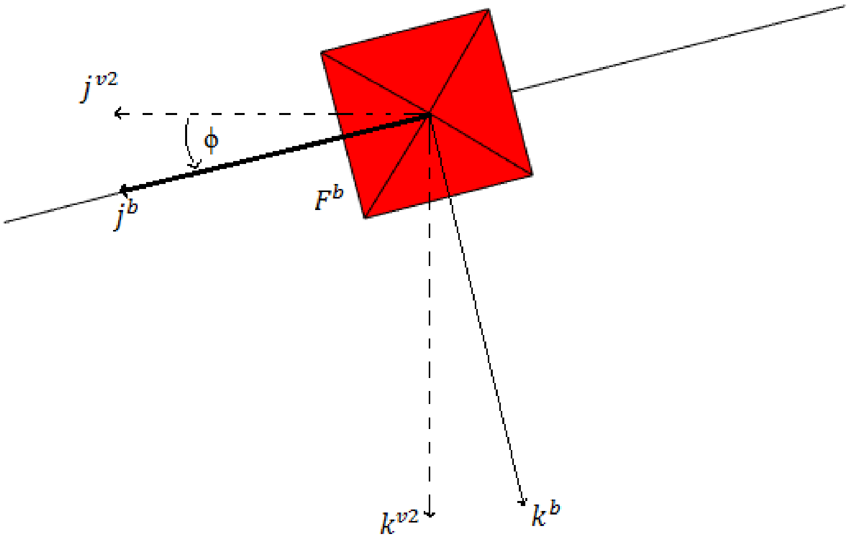


Figure 2.10 Body Frame based on [3]

The Body Frame (F^b) is derived from the Vehicle-2 frame (F^{v2}) via a right hand rotation through the roll angle(ϕ) [3]. The rotation makes j^b pass through the starboard wing and k^b pendicular to that (pointing down) while maintaining the origin at the center of mass/center of gravity. The rotation matrix (Eq. 2.13) for this is as follows [[3][4][5]].

$$R_{v2}^{b}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \qquad \text{[Eq. 2.13]}$$

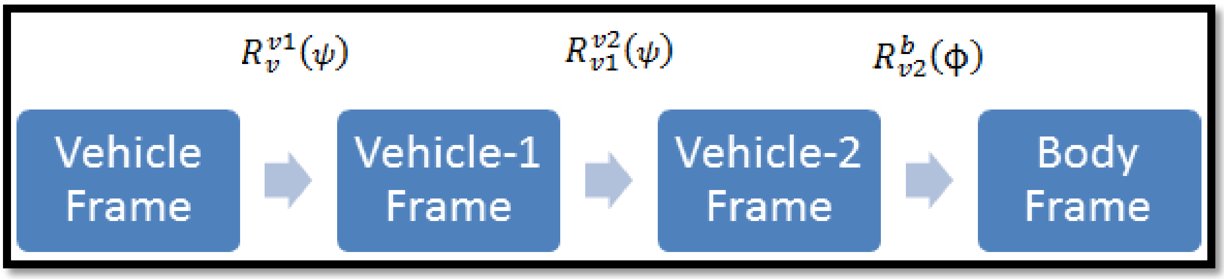


Figure 2.11 Transformation of Vehicle to Body frames through Rotation Matrices

One of the strengths of this breakdown of frame to frame transition is that it explains the transition from $F^v$to F^bin a more intelligible manner. This is because the sequence of rotations involved to go from $F^v$to F^b include the rotations described by the matrices used to go from $F^v$to F^{v1} and $F^{v1}$to F^{v2}. Following the general convention [4][5][6], therefore the frames may be transformed from $F^v$to F^b [3] through the following rotation matrix in Equation 2.14.

$$R_{v}^{b}(\phi,\theta,\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[Eq. 2.14]

After the matrix multiplication, this yields the same rotation matrix as found in the literature [3][4][5][6] and can be expressed as shown in Equation 2.15.

$$R_{v}^{b}(\phi,\theta,\psi) = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}$$

[Eq. 2.15]

In the Methodology and Results sections, the underlying mathematical models and simulation details for the aerodynamic forces and resulting moments are shared. Those calculations are carried out on the Stability Frame (F^s).

2.1.3.6 Stability Frame:

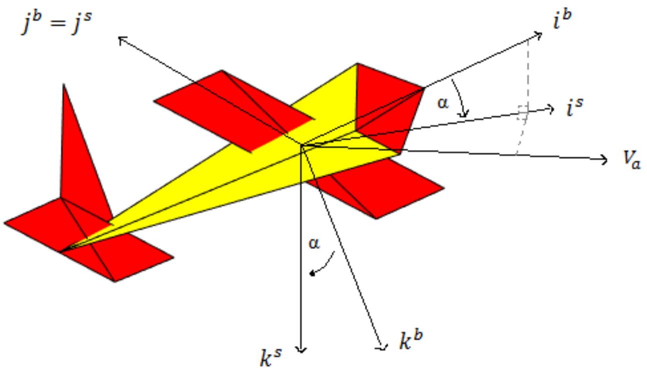


Figure 2.12 Stability Frame based on [3]

The Stability Frame (F^s) is derived from the Body Frame (F^b) by rotating about the j^b axis[3]. Instead of using a right hand rotation, a left hand rotation is utilised to accomade the need for the angle of attack(α) to be defined as a positive rotation about j^b/ j^s from this new frame[3][4][5]. As stated earlier, this frame is important as the alignment of the airspeed vector (V_a) and i^saxis allows for the aerodynamic calculations to be carried out. Therefore the left-handed rotation matrix is utilised [3] and write down the rotation matrix in Equation 2.16.

$$R_b^s(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \qquad \text{[Eq. 2.16]}$$

2.1.3.7 Wind Frame:

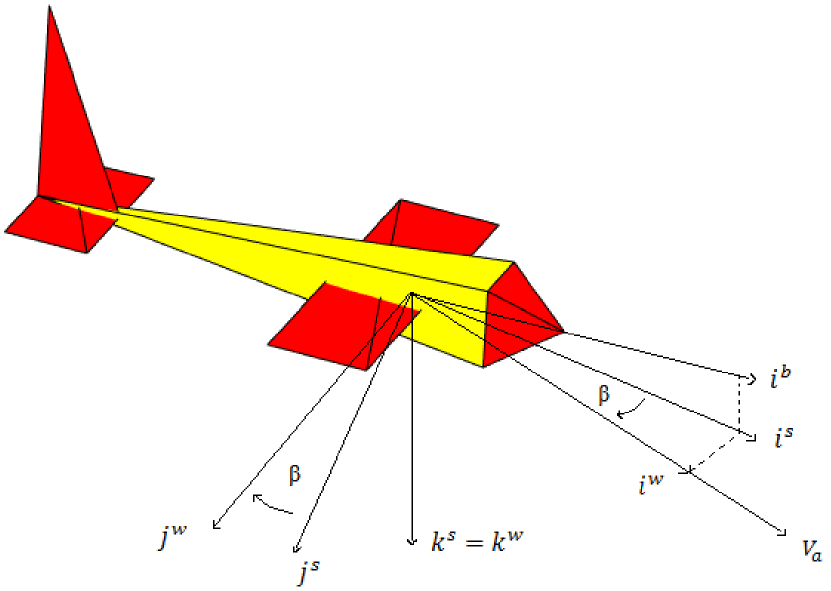


Figure 2.13 Wind Frame based on [3]

In later chapters, the modelling of the side-slip and the aircraft's dynamic responses relative to the oncoming airflow are discussed. This is an important consideration and was modelled to factor in and capture the sideways travel or translation of the airframe as it attempts to move forward. The angular displacement of the heading (side-slip angle) there is referred to as β. This necessitates the transition from the Stability Frame (F^s) to the Wind Frame(F^w) and it can be achieved with ease via a right hand rotation about the vertical $k^s = k^w$ axis by the amount β [3][4][5]. Therefore the rotation matrix is as shown in Equation 2.17.

$$R_s^w(\beta) = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{[Eq. 2.17]}$$

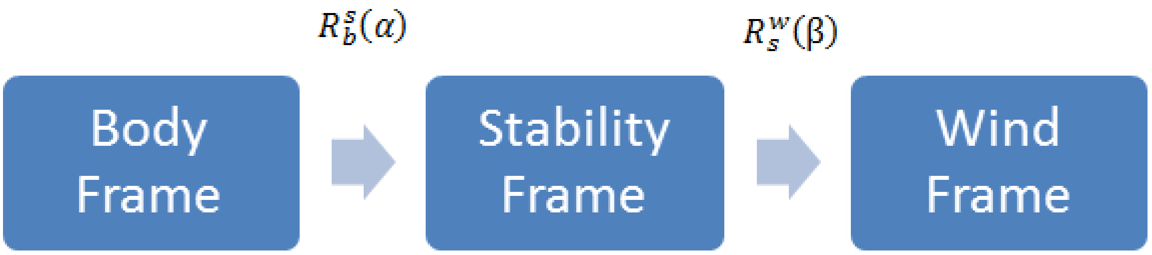


Figure 2.14 Transformation of Body to Wind Frame through Rotation Matrices

Similar to the the transitional process of Vehicle Frame(F^v) to Body Frame (F^b), the Stability Frame(F^s) in this case can be used as a step to the conversion of the Body

Frame(F^b) into the Wind Frame(F^w). That once again yields the same rotation matrix as within the literature [3][4][5] (Eq. 2.18) for such conversions.

$$R_b^w(\alpha,\beta) = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \qquad \text{[Eq. 2.18]}$$

After the matrix multiplication, the rotation matrix can be written as Equation 2.19 [3].

$$R_b^w(\alpha,\beta) = \begin{bmatrix} \cos\beta\cos\alpha & \sin\beta & \cos\beta\sin\alpha \\ -\sin\beta\cos\alpha & \cos\beta & -\sin\beta\sin\alpha \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \qquad \text{[Eq. 2.19]}$$

This not only allows for a frame conversion from the Body Frame to the Wind Frame, the same matrix and use of it's transpose, $(R_b^w)^T$ can be used to convert frames the other way around if needed.

2.2 Aerodynamic Characteristics

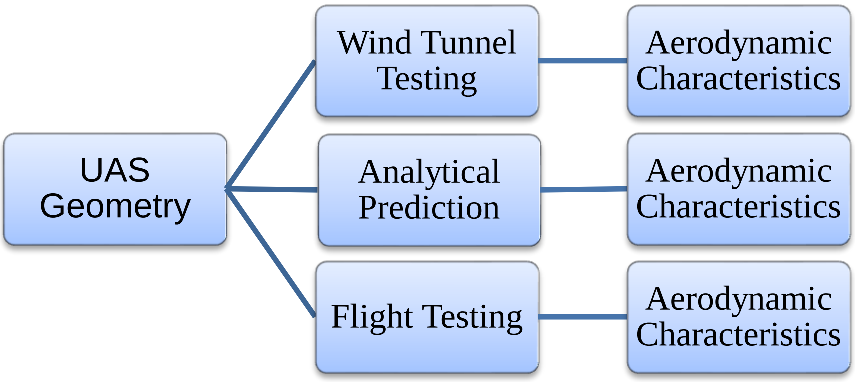

UAS Geometry
Wind Tunnel Testing
Analytical Prediction
Flight Testing
Aerodynamic Characteristics
Aerodynamic Characteristics
Aerodynamic Characteristics

Figure 2.15 Three approaches to the derivation of aerodynamic characteristics

It is possible to deduce the parameters needed for the aerodynamic characteristics of the UAS platform through multiple methods [13][14]. Methods such as Wind Tunnel Testing and Flight Testing of the actual UAS platform can provide valuable data at the cost of a lot of technical expertise, resources and time [15][16][17]. As such, for this research, the Analytical Prediction route was explored for convenience. Software packages like the USAF Digital DATCOM , XFLR5 and X-Plane enable, at the very least, some working estimation for these parameters [18][19]. Once the stability and control derivatives are obtained, a more accurate MATLAB/Simulink model can be created.

2.3 MATLAB vehicle system modelling

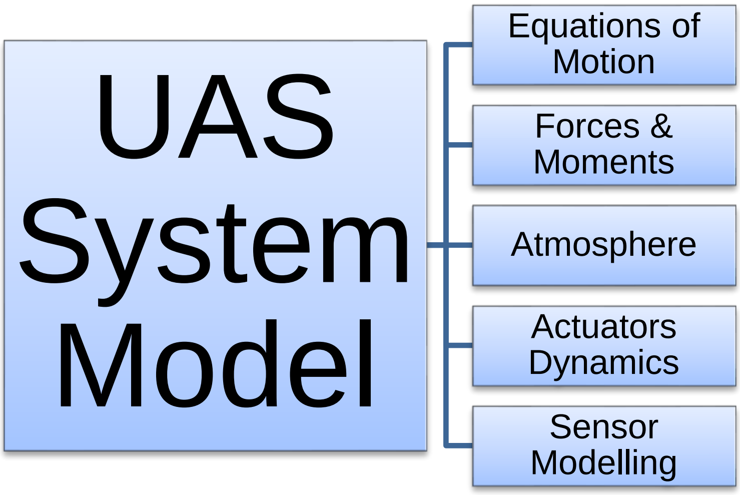


Figure 2.16 Components of usual vehicle system modelling

The identified modules of the overall system design can be easily realised through the use of the Aerospace Blockset™ blocks in MATLAB/Simulink [20].

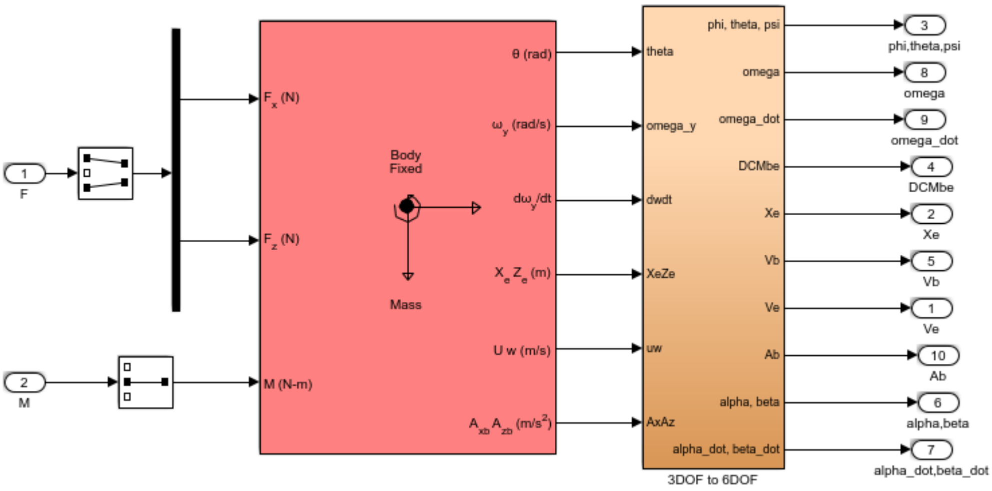


Figure 2.17 3 degree-of-freedom Euler block [20]

A block such as the one portrayed above can be easily utilised to analyse the longitudinal stability characteristics of the UAS as it allows us to implement the equations of motions of the UAS platform. As long as the aircraft is stable, there is no reason why it cannot be extended to implement a 6 DOF (6 Degrees of Freedom) system. These blocks however require that Digital DATCOM or some other means of supplying the stability and control derivative are used [20][21].

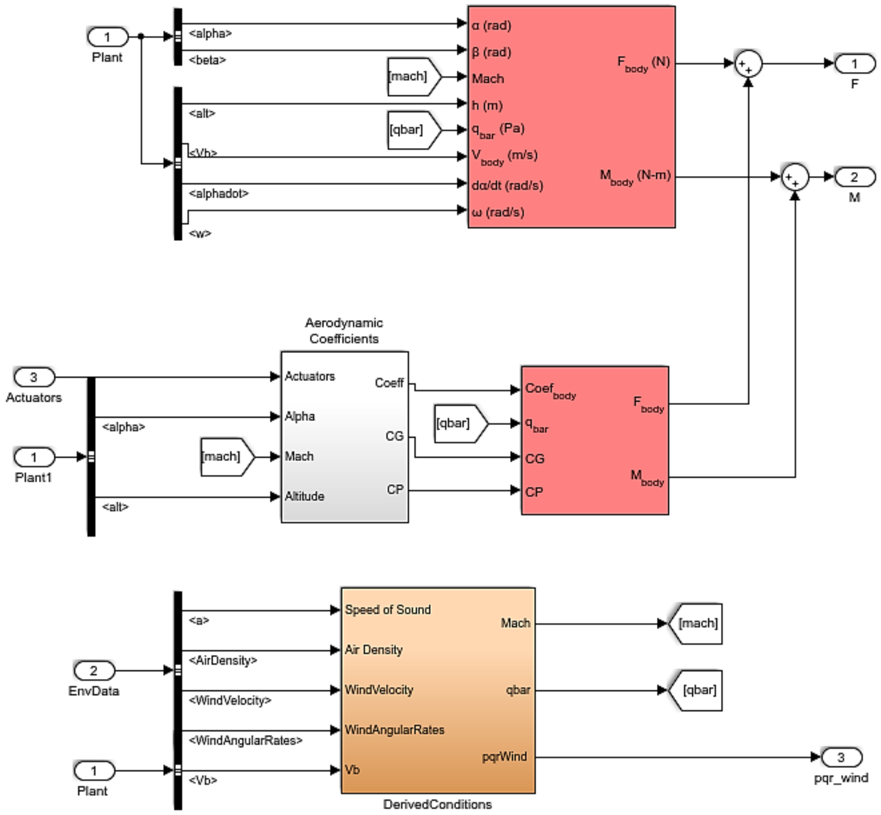


Figure 2.18 Integration of aerodynamic block and other blocks [20]

Digital DATCOM block integrated with Aerodynamic Forces and Moments blocks [20] In a very similar manner, Simulink allows us to model the other aforementioned modules of the system, such as the actuators, sensors and atmosphere as depicted in the following image.

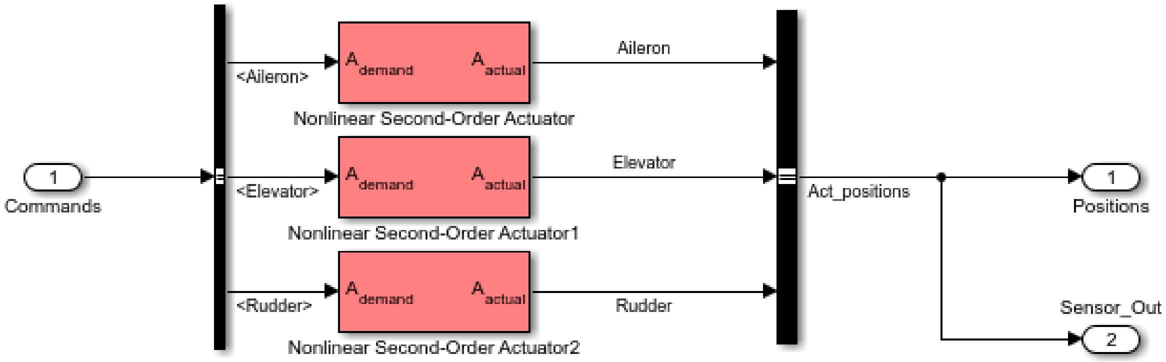


Figure 2.19 Depiction of actuator modelling implemented using Aerospace Blockset [20]

Simulink block integration for actuators, environment and sensor modelling [20] While these functionalities of MATLAB/Simulink are widely used, they come at a premium and are not as accessible to those under financial constraint. However, the standard package does include the ability to model similar blocks with similar functionality by the use of MATLAB function blocks and S-Function blocks [22][23]. It is possible to create these blocks and run them both compiled or interpreted. The S-function blocks also have promising potential benefits for integration with existing code written in the C language [24]. The downside to this approach is that it requires that the researcher provide the underlying mathematical models and setup all other block related configurations themselves. This highlights the need for proper mathematical modelling discussed in the next section.

2.4 Mathematical Modelling

While application of the simulation tools (described in the Methods and Results sections) for the constructions of the various models of the FMS 182 (target airframe) requires a mixed approach, it is important to understand the underlying mathematical modelling and their theoretic foundation. Accurate dynamic modelling of a UAS platforms demands appropriate insight into the structural and aerodynamic characteristics of the platform. There are three popular approaches to such modelling [25] and they are:

- First-principle modelling
- Grey-box modelling
- Black-box modelling

Due to the reason that black-box modelling doesn't provide enough insight into the system, only the former two modelling techniques are considered. The first-principle, aka the white-box, modelling system takes into consideration the inertial, aerodynamic, and structural properties [25]. The most popular systematic approaches known are the Lagrangian and Newton-Euler methods [26][27]of achieving this. However, this approach can be very detailed and therefore time-consuming. One drawback of the reliance on this approach alone is that validation can become very difficult as certain aerodynamic parameters may not be easily obtained with accuracy [26][27]. This problem can be addressed through Grey-box modelling where we exploit actual experimental data to derive accurate parameters for the specific UAS platform (FMS 182) [25]. This type of modelling has gained popularity in controller design approaches due to its accuracy [28].

However, due to practical considerations, it isn’t always feasible to start with actual flight test experimentation and as such the research approach may rely first on the First-principle modelling and only improve on that with Grey-box modelling once appropriate levels of simulation tests and actual flight tests are completed. Until that stage is reached, tools discussed earlier, such as X-Plane, remain the substitute for any such flight tests.

2.5 Various modelling considerations

The equations of motion for the UAS platform can be arrived at from the Newtonian laws of motion. The second law implies that the resultant forces on the platform can be equated to the derivative of its momentum [2]:

$$\sum F = \frac{d}{dt}(mV) \qquad \text{[Eq. 2.20]}$$

Where F is force, m is mass and v is speed (Eq. 2.20). Similarly the resultant moment on the UAS platform can be stated as the time derivative of the angular momentum [2]:

$$\sum M = \frac{dH}{dt} \qquad \text{[Eq. 2.21]}$$

Where M is the moment and H is the angular momentum (Eq. 2.21). At this point, if certain assumptions about things like:

- Symmetry Plane of the UAS
- Constant mass of the UAS
- Rigidity of the UAS platform
- Inertial reference plane

can be made, we may derive the important rotation matrix equation (Eq. 2.22) which helps us transform between axis systems while providing us with the body angular velocities of the UAS[29].

$$R = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & sin\phi cos\theta \\ 0 & -sin\phi & cos\phi cos\theta \end{bmatrix} \qquad \text{[Eq. 2.22]}$$

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = R \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \qquad \text{[Eq. 2.23]}$$

This is done in the same spirit as detailed in the section describing the Coordinate Frames. R is the transformation matrix while ϕ, θ and ψ are the Euler angles (Eq. 2.23). Derivations of the force and moment equations based on these principles are presented in the following sections. It is important to note that due to the reasoning provided in the section describing the Coordinate Systems used, it is important to get the axis systems and any translation between the systems right. Not to mention that depending on the number of degrees of freedom, the representation of these equations may change.

Once the determination of the parameters are made, methodological approaches can be exploited to determine the State Space models for the UAS [30][31][32] leading to the potential development of autonomous guidance, navigation and control systems for the fixed-wing airplane.

2.5.1 Longitudinal State Space Model

The Roskam method may be used [33] to generate the perturbation equations under the assumption of straight and level flight ($\dot{\theta} = q \quad \ddot{\theta} = \dot{q}$) with the following set of equations (Eq. 2.24) as:

$$\begin{gathered}
\dot{u} = -g\theta\cos\,\Theta_1 + X_u u + X_{T_u} u + X_\alpha \alpha + X_{\delta_e}\delta_e \\
U_1\dot{\alpha} - U_1\dot{\theta} = -g\theta\sin\Theta_1 + Z_u u + Z_\alpha \alpha + Z_{\dot{\alpha}}\dot{\alpha} + Z_q\dot{\theta} + Z_{\delta_e}\delta_e \\
\ddot{\theta} = M_u u + M_{T_u} u + M_\alpha \alpha + M_{T\alpha}\alpha + M_{\dot{\alpha}}\dot{\alpha} + M_q\dot{\theta} + M_{\delta_e}\delta_e
\end{gathered}$$

[Eq. 2.24]

This can be re-written in the State Space format as shown in Equation 2.25.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (U_1 - Z_{\dot{\alpha}}) & 0 & 0 \\ 0 & -M_{\dot{\alpha}} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u + X_{T_u} & X_\alpha & 0 & -g\cos\,\Theta_1 \\ Z_u & Z_\alpha & U_1 - Z_q & -g\sin\Theta_1 \\ M_u + M_{T_u} & M_\alpha + M_{T\alpha} & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_{\delta_e} \\ Z_{\delta_e} \\ M_{\delta_e} \\ 0 \end{bmatrix} [\delta_e]$$

[Eq. 2.25]

As long as the dimensional derivatives for our UAS platform (FMS 182) are known, they can be entered into the above equations and this should permit further utilisation of these models for the future development work for the autonomous control.

2.5.2 Lateral State Space Model

In a similar fashion to the determination of the State Space format for the longitudinal case, the Roskam method [33] can be followed where the dynamics can be represented with the following set of equations (Eq. 2.26) as shown below for steady and level flight ($p = \dot{\phi}\,;\quad \dot{p} = \ddot{\phi}\,;\quad r = \dot{\psi};\quad \dot{r} = \ddot{\psi}$):

$$
\begin{gathered}
U_1\dot{\beta} + U_1\dot{\psi} = g\phi\cos\,\Theta_1 + Y_\beta\beta + Y_p\dot{\phi} + Y_r\dot{\psi} + Y_{\delta_\alpha}\delta_\alpha + Y_{\delta_r}\delta_r \\
\ddot{\phi} - \bar{A}_1\ddot{\psi} = L_\beta\beta + L_p\dot{\phi} + L_r\dot{\psi} + L_{\delta_\alpha}\delta_\alpha + L_{\delta_r}\delta_r \\
\ddot{\psi} - \bar{B}_1\ddot{\phi} = N_\beta\beta + N_{T_\beta}\beta + N_p\dot{\phi} + N_r\dot{\psi} + N_{\delta_\alpha}\delta_\alpha + N_{\delta_r}\delta_r \\
\bar{A}_1 = \frac{I_{xz}}{I_{xx}} \quad and \quad \bar{B}_1 = \frac{I_{xz}}{I_{zz}}
\end{gathered}
$$

[Eq. 2.26]

Just as before, this may be transformed to the State Space format (Eq. 2.27).

$$
\begin{bmatrix} 1 & 0 & 0 & \bar{A}_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_1 & 0 \\ -\bar{B}_1 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} \dot{p} \\ \dot{\phi} \\ \dot{\beta} \\ \dot{r} \end{bmatrix}
= \begin{bmatrix} L_p & 0 & L_\beta & L_r \\ 1 & 0 & 0 & 0 \\ Y_p & g\cos\,\Theta_1 & Y_\beta & (Y_r - U_1) \\ N_p & 0 & N_\beta + N_{T_\beta} & N_r \end{bmatrix}
\begin{bmatrix} p \\ \phi \\ \beta \\ r \end{bmatrix}
+ \begin{bmatrix} L_{\delta_\alpha} & L_{\delta_r} \\ 0 & 0 \\ Y_{\delta_\alpha} & Y_{\delta_r} \\ N_{\delta_\alpha} & N_{\delta_r} \end{bmatrix}
\begin{bmatrix} \delta_\alpha \\ \delta_r \end{bmatrix}
$$

[Eq. 2.27]

As required for the longitudinal case, if substitution of the dimensional derivatives into the above equation is done, the lateral State Space model for the target fixed-wing UAS (FMS 182) can be obtained. However, there are some key factors of consideration which leads to producing some drawbacks for the State Space method. They are as follows:

- Such State Space modelling is simplified linear modelling.
- The coupling effects between the longitudinal and lateral dynamics are ignored.
- Rotation of the earth is ignored due to the use of the Earth-Fixed Body-Fixed coordinate system [43].
- The linearization may force an assumption of constant thrust.
- If full state outputs are not available, simulation of any navigation mode may become impossible.

2.6 Parameter Estimation

Parameter Identification/Parameter Estimation involves the methods used for the estimation of stability/control derivatives and behaviours from flight data [25][27][34]. Various different methods ranging from Pallister's study of lateral stability in the 60's to Klien's contributions [35][36][37][38] to the field led to the more systematic approaches widely adopted today. Further developed approaches emerged from the works of Baek [39] who successfully derived the parameters of a Jetstream in the 90s. While the methods typically used were developed for processing data from actual test flights of fixed-wing full-scale aircraft, there is nothing in principle that would stop its use for the small scale fixed-wing UAV like the FMS 182 (the target airframe).

Many of these available approaches have resulted in establishing a field of study of its own. The complications and complexity involved results in hardship in successfully implementing them. A simpler approach is considered by Mullen [34] to determine mathematical models from flight data. The two following parametric models (Eq. 2.28-2.29) for analysis of longitudinal dynamic modes of fixed-wing aircraft can be obtained from Mullen [34].

$$\begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} z_w & z_q \\ m_w & m_q \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} z_\eta \\ m_\eta \end{bmatrix} \eta \qquad \text{[Eq. 2.28]}$$

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_u & x_w & x_q & x_u \\ z_u & z_w & z_q & z_\theta \\ m_u & m_w & m_q & m_\theta \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} x_\eta \\ z_\eta \\ m_\eta \\ m_\theta \end{bmatrix} \eta \qquad \text{[Eq. 2.29]}$$

The first model is for the reduced form of longitudinal state space model of the short period mode of the aircraft, while the larger model underneath is the full order form [34]. The terms u, w, q and θ are forward velocity, vertical velocity, pitch rate and pitch angle. The dot accent implies the time derivative of these parameters. η is the elevator deflection. The remaining parameters are all stability derivatives in dimensional form. While the approach is simpler, basic understanding of flight dynamics is presupposed as the process of starting with flight data and ending with the above equations populated with estimations has steps that if not performed properly will render the results unreliable. Adding to that is the fact that ultimately these methods are analytical approximations and will have their own degree of uncertainties involved.

Although the example MATLAB implementations of this simplified version are written in old MATLAB format and were not meant to readily accept the data from modern external simulation platforms (eg. X-Plane), the MATLAB implementation is promising for potential development to make it work for the purpose of this research.

2.7 Mathematical models for MATLAB/Simulink implementation

The approach to modelling all the Forces and Moments acting on the aircraft discussed earlier is too simplistic for producing simulations with reasonable degree of fidelity. Luckily, there exists sufficient amount of derivations of the equations discussed earlier in the forms that are more usable in MATLAB/Simulink modelling. The following derivation of useful mathematical models are standard published sets of equations and formulations used for the kinematics and dynamics modelling of fixed-wing aircraft [2][3][4][30][33][50][51][52][53][54][55][56][57]. This derivation based on the literature is presented in the following subsections relating to the mathematical modelling.

2.7.1 Aerodynamic Forces and Moments

The forces and moments acting on the aerofoil are based on the dynamic pressure ($\frac{1}{2}\rho {V_a}^2$) acting on the aerofoil. These forces and moments are measured at the aerodynamic center of the aerofoil.

$$F_{lift} = \frac{1}{2}\rho {V_a}^2 S C_L \qquad \text{[Eq. 2.30]}$$

$$F_{drag} = \frac{1}{2}\rho {V_a}^2 S C_D \qquad \text{[Eq. 2.31]}$$

$$m = \frac{1}{2}\rho {V_a}^2 S c C_m \qquad \text{[Eq. 2.32]}$$

The derivation process [3] starts from the general expressions for aerodynamic lift, drag and moment (Eq. 2.30-2.32). S represents the platform area of the wing/aerofoil, $C_L/C_D/C_m$ are non-dimensional aerodynamic coefficients and c represents the mean chord of the aerofoil [3]. These forces and moments are then broken down to their longitudinal and lateral components and modelled appropriately.

2.7.2 Longitudinal Forces and Moments

The dynamic behaviour and motion of the UAV in the pitch plane is a factor of the longitudinal component of the aerodynamic forces & moments acting on it. Functionally, the angle of attack (α), pitch rate (q) and the elevator deflection (δ_e) goes on to have the greatest impact on the dynamics of the UAV on this plane. Even though these equations are non-linear in nature, we are able to arrive at a linear approximation under the presupposition of maintaining the angle of attack (α) small enough such that the airflow over the aerofoil remains attached and is laminar [2-4]. The general first-order Taylor series approximation of these equations can therefore be written as [3]:

$$F_{lift} = \frac{1}{2}\rho {V_a}^2 S\left[C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e\right] \quad \text{[Eq. 2.33]}$$

$$F_{drag} = \frac{1}{2}\rho {V_a}^2 S\left[C_{D_0} + C_{D_\alpha}\alpha + C_{D_q}\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e\right] \quad \text{[Eq. 2.34]}$$

$$m = \frac{1}{2}\rho {V_a}^2 Sc\left[C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e\right] \quad \text{[Eq. 2.35]}$$

The C_{L_0}, $C_{D_0}$and C_{m_0} are merely the respective lift, drag and pitching moment coefficients for the condition where $\alpha = q = \delta_e = 0$. The factor $\frac{c}{2V_a}$ is introduced into the equation to simply make the q term dimensionless [3]. In this Taylor series approximation representation (Eq. 2.33-2.35), within the literature, the terms dealing with α and q are classed as the stability derivatives and the ones involving δ_e are referred to as the control derivatives. The fixed-wing in use as our target platform excludes extreme manoeuvres or acrobatic dynamics and as such the flight envelope only includes steady and stable flight mechanics. As such, within the relevant flight envelop, this linearization approach with the assumption of low angle of attack can be argued to capture the dynamics of the UAV appropriately. Therefore almost all complications of modelling from unsteady airflow behaviour can be ignored except for the stall characteristics [3].

2.7.3 Lateral Forces and moments

$$f_y = \frac{1}{2}\rho {V_a}^2 S C_Y(\beta, p, r, \delta_a, \delta_r) \quad \text{[Eq. 2.36]}$$

$$l = \frac{1}{2}\rho {V_a}^2 Sb C_l(\beta, p, r, \delta_a, \delta_r) \quad \text{[Eq. 2.37]}$$

$$n = \frac{1}{2}\rho {V_a}^2 SbC_n(\beta, p, r, \delta_a, \delta_r) \qquad \text{[Eq. 2.38]}$$

The aerodynamic forces and moments acting on the UAV on the lateral plane are responsible for both rotational and translational motions along that plane [2-4][57]. The sideslip (β), roll rate (p), yaw rate (r), aileron deflections (δ_a and δ_r) are the primary and most significant contributors [3] to these forces and moments. In Equations 2.36-2.38, f_y represents the lateral aerodynamic forces acting on the UAV [3]. The moments arising from roll is represented with l and the n represents the moments arising from the yaw. In the same manner as before, the equations are linearized [3]. This results in the type of formulation presented later in the methods section.

2.7.4 Propulsive Forces & Moments

$$F_{x_p} = S_{prop}C_{prop}\left[P_{downstream} - P_{upstream}\right] \qquad \text{[Eq. 2.39]}$$

Beard et al [3]. utilise the following derivation of the formulation for modelling of the forces and moments caused by the power plant. In order to simplify the modelling process of the thrust generated by the propeller and the associated torque, the following are assumed:

- The trust produced by the propellers acts only along the body axis of the UAV.
- The propeller is perfectly efficient.

In Equation 2.39, F_{x_p} represents the thrust produced by the propeller acting along the body axis. S_{prop} is the area that the propeller sweeps out and C_{prop} is an aerodynamic coefficient of the propeller and they are both constants. These values need to be measured or estimated by analytical means. This translates to the determination of the thrust force by the difference in air pressure ahead of and behind the propeller ($P_{downstream} - P_{upstream}$). From Bernoulli's equations [3][58]:

$$P_{upstream} = P_0 + \frac{1}{2}\rho {V_a}^2 \qquad \text{[Eq. 2.40]}$$

$$P_{downstream} = P_0 + \frac{1}{2}\rho {V_{exit}}^2 \qquad \text{[Eq. 2.41]}$$

The V_{exit} term (Eq. 2.40-2.41) stands for the velocity of the air right after the propeller has acted on it. It can be further simplified (Eq. 2.42) by ignoring the complications of any transient response and therefore arriving at the following linear expression [3].

$$V_{exit} = k_{motor}\delta_t \qquad \text{[Eq. 2.42]}$$

2.7.5 Gravity

$$f_g^v = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \qquad \text{[Eq. 2.43]}$$

$$f_g^b = R_v^b f_g^v \qquad \text{[Eq. 2.44]}$$

Modelling of gravitational forces has the simplest form [3] where previously stated rotation matrices are utilised to convert from one frame to another(Eq. 2.43-2.44) while retaining the basic description as the product of mass and gravitational acceleration on earth.

2.7.6 Moments modelling

$$\begin{bmatrix} l \\ m \\ n \end{bmatrix} = moments_{aerodynamic} + moments_{propeller} + moments_{gravity} \qquad \text{[Eq. 2.45]}$$

As done for the forces, the same approach to sum together the combined moments acting on the UAV is taken (Eq. 2.45-2.46). Therefore, we can express the total moments as follows [3].

$$\begin{bmatrix} l \\ m \\ n \end{bmatrix} = \frac{1}{2}\rho {V_a}^2 S \begin{bmatrix} b\left[C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_r}\delta_r\right] \\ c\left[C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e\right] \\ b\left[C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a + C_{n_r}\delta_r\right] \end{bmatrix} + \begin{bmatrix} -k_{T_p}(k_\Omega\delta_t)^2 \\ 0 \\ 0 \end{bmatrix}$$

[Eq. 2.46]

2.7.7 Dynamics and Kinematics

$$\frac{d}{dt}\begin{bmatrix} p_n \\ p_e \\ p_d \end{bmatrix} = R_b^v \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$ [Eq. 2.47]

Via the use of rotation matrices, position values and the velocity vectors of the airframe are related (Eq. 2.47). The formulation then becomes as shown in Equation 2.48 [3].

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}^T \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

[Eq. 2.48]

The same type of formulation can be used for the roll (p), pitch (q), r (yaw) rates and their derivatives. This is shown in more detail in the section describing the Simulink modelling methods. More details on these individual derivations are available from the source literature [2-4][30][33][51-58]. There are variations and different formulations. Since the templates from R. W. Beard and T. W. McLain were used for the Simulink work, the derivations and formulations presented by them are highlighted for pertinence. The formulations they present are also highly useful for transcribing to code, which is a necessity for modelling and simulation work.

2.8 Aerodynamics, Stability and Control Parameters

Modelling process of a fixed-wing aircraft relies on accurate information on various parameters about the aircraft relating to its aerodynamics and stability characteristics in general. This is evident from the large number of such parameters the previously mentioned sections on mathematical modelling and Aerospace Blockset (Simulink) [20] demand from the researcher.

There are various methods of System Identification which can greatly aid in this process. The two general types are the parametric and non-parametric types [59]. Since this research focuses on model building, the parametric type is of great utility. This works by using actual flight test data alongside a known model of the system. There are various different types of such identification techniques which specialise on linear/non-linear systems, time/frequency domains, all the way from white to black box models [60][61][62]. The Parameter Estimation/Identification discussed earlier is a subset of this field. Although these existing techniques are effective and useful, it isn’t always feasible to rely on them for various reasons. As stated previously, these methods rely on actual measured flight data [59-62] which may not be available. Small scale fixed-wing aircraft like the target airframe used for this research operate in Reynolds number regimes where the dynamics can be very different [63][64]. This highlights the importance of obtaining aerodynamic data for the small-scale platform which is therefore guaranteed to behave differently at the expected low Reynolds number ranges of below around 500k [64]. The inertial properties can also significantly impact the ability to test the aircraft itself and provide the expected inputs.

The general pattern of application of these methods follows the following theme [59][63][66]:

- Prepare Aircraft for flight test
- Conduct Flight Test under specific constraints
- Provide very specific sets of inputs (to excite specific dynamic modes)
- Record Flight Data
- Process Flight Data
- Iteratively Improve the model

This of course translates to certain limitations of special importance to this research. One of the limitations is that the researcher must have the aircraft already been modelled (e.g. Simulink/X-Plane/etc.). Alternatively, it presupposes access to fixed-wing platforms that already are flight-ready and have proper control systems in place both to safely fly and conduct the series of flight tests necessary for System Identification. This strongly implies and argues for the necessity of preliminary models. One way to mitigate these problems is via the use of computational methods and tools discussed in the next subsections. It should also be noted here that in theory it is possible to extract many of these parameters from wind tunnel testing. That however is an expensive and resource intensive method [66] which may not even be available to most researchers and prospective researchers around the world.

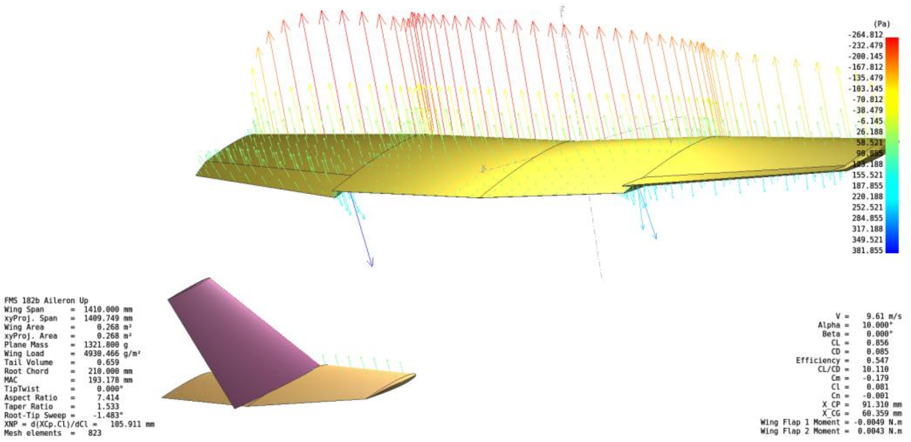
2.8.1 CFD

Computational Fluid Dynamics exploits modern computer systems and their processing power to numerically analyse fluid flow problems based on simulation of the fluid and interacting mediums [67]. There are a range of industrial-grade CFD software packages, such as Star CCM/Ansys/SolidWorks, etc., capable of handling Aerodynamic simulations [68][69][70], across a range of test conditions (airspeed, density, Reynolds number, etc.). Tools like them can be exploited to get the aerodynamic parameters for improved modelling. However, they suffer from a range of practical issues such as:

- Traditional CFD packages are likely to be expensive.
- Their successful operations presuppose a great degree of knowledge of Fluid Dynamics.
- This is a specialization field and most researchers may lack the required skills or number of years required to attain them.

Considering the fact that the test results obtained from these computational means will eventually need to be juxtaposed against the results obtained from flight tests, alternative and simpler CFD code may prove to be more cost effective and less resource intensive. The XFoil code developed by Mark Drela for MIT is an example of such a tool where 2D panel codes are used alongside boundary layer codes for aerofoil analysis [71]. The open-source software package XFLR5 [72] implements the XFoil and allows for aerodynamic and stability/control analysis based on 3D panel method, VLM and Lifting Line Theory [73][74].

2.8.1.1 XFLR5



```
FMS 182b Aileron Up
Wing Span        =  1410.000 mm
xyProj. Span     =  1409.749 mm
Wing Area        =     0.268 m²
xyProj. Area     =     0.268 m²
Plane Mass       =  1321.800 g
Wing Load        =  4930.466 g/m²
Tail Volume      =     0.659
Root Chord       =   210.000 mm
MAC              =   193.178 mm
TipTwist         =     0.000°
Aspect Ratio     =     7.414
Taper Ratio      =     1.533
Root-Tip Sweep =    -1.483°
XNP = d(XCp.Cl)/dCl =   105.911 mm
Mesh elements  =   823
```

```
                    V =    9.61 m/s
                Alpha =   10.000°
                 Beta =    0.000°
                   CL =    0.856
                   CD =    0.085
           Efficiency =    0.547
                CL/CD =   10.110
                   Cm =   -0.179
                   Cl =    0.081
                   Cn =   -0.001
                 X_CP =   91.310 mm
                 X_CG =   60.359 mm
 Wing Flap 1 Moment = -0.0049 N.m
 Wing Flap 2 Moment =  0.0043 N.m
```

Figure 2.20 An aircraft performance test visualisation from XFLR5

XFLR5’s potential flow method use allows for faster computation time and it can run on most average modern computers. It utilises the vicious boundary layer models from XFoil along with its panel solvers [75]. As Shafer et al [75] concludes and confirms, the method used in XFLR5 is capable of similar solutions to that of the more sophisticated approaches in CFD for low angle of attacks. Considering that the target airframe is meant to operate at steady level flight situations throughout most of the flight envelop, this makes it the most cost effective approach to estimating the necessary parameters.

XFLR5 is capable of exploiting the XFOIL code for aerofoil analysis and even predicting stability/control derivatives (crucial for the modelling) [75][76]. The benefit from such useful features is however limited by the effort put into creating a large number of reasonably accurate models necessary for such applications. Another potential problem source is the lack of fuselage modelling. While it is possible to model the fuselage of the plane for aerodynamic analysis, the solver method needs the wings and fuselage to be separated to avoid erroneous outputs. This means that adding the fuselage to the model reduces the accuracy of the simulations.

The data generated by XFLR5 are however both useful for the type of MATLAB/Simulink model discussed earlier and the widely available commercial simulators like the ones discussed in the next subsection.

2.8.2 Flight Simulators

The most widely used commercial flight simulators for research application are X-Plane [77] and FlightGear [78]. FlightGear relies on the users to provide the models for visualisation while X-Plane simulates its own Flight Models. Therefore, modelling in FlightGear may be used as an external visualizer for a MATLAB/Simulink simulation but cannot be considered as an independent simulation by itself. As such, the utilities of X-Plane for research application are discussed in Chapter 2.8.2.2.

2.8.2.1 Comparison of Simulation Solutions

Accessibility of any given simulation platform is likely to impact its utilisation by prospective researchers and interested parties. In the light of that, comparison of various different solutions is presented in Table 2-2. The comparison is based on the following criteria:

- Cost of the hardware
- Cost of the software
- Level of expertise needed
- Simulation Fidelity

	Free Software	**MATLAB**	**MATLAB+**	**X-Plane**	**FlightGear**	**Industrial**
Hardware cost	L	M	M	M	M	H
Software Cost	L	M	M-H	L-M	L	H
Expertise	H	M	M	L-M	M	M-H
Fidelity	L-H	M-H	M-H	M-H	L-H	H

Table 2-2 Comparison of the most popular flight simulator platforms for research (L =low, M = medium, H= high)

In the above table, 6 different means of simulations are compared. Columns 2-7 represent the 6 plausible solutions. Column 2 (Free software) represents a setup where every module of the simulation platform is coded from scratch in a programming language (e.g. C++). Column 3 represents the utilisation of MATLAB/Simulink basic package while column 4 (MATLAB+) represents the same package with premium add-ons [20]. Columns 5 and 6 represent the widely used simulation solutions discussed earlier while the final column (Industrial) represents the use of professional/industrial simulation solutions such as the Merlin [86].

Due to the variability in the absolute costs of these solutions, they are compared on a relative scale ranging from low to high. As Table 2-2 shows, there is no straight forward answer to the problem of platform choice. But general patterns do aid in the process of elimination based on the most important criteria. For example, the lowest cost is for the free software based custom coded simulation solution and the highest cost is for the Industrial solutions. The high cost of such industrial simulation solutions already rules it out as a preferred solution. While the custom coded free software based on is desirable in terms of cost, the expertise (high skills) required for its development makes it inaccessible for a wide range of researchers/interested parties. In the light of that, the Table 2-3 is presented to compare the remaining solutions.

	MATLAB/ Simulink	MATLAB/Simulink (with premium add-ons such as Aerospace Blockset)	X-Plane	FlightGear
RAM	8 GB	8GB	16-24 GB RAM or more	6-8 GB
Processor	Any Intel or AMD x86-64 processor with four logical cores and AVX2 instruction set support	Any Intel or AMD x86-64 processor with four logical cores and AVX2 instruction set support	Intel Core i5 6600K at 3.5 Ghz or faster	A quad core processor with ~ 2 GHz each, 64 bit architecture
Graphics	No specific graphics card is required.	No specific graphics card is required.	A DirectX 12-capable video card from NVIDIA, AMD or Intel with at least 4 GB VRAM (GeForce GTX 1070 or better or similar from AMD)	1024-2048MB of dedicated DDR3+ (DDR5 preferred) VRAM (i.e. 512 Mb VRAM minimum)
Cost (relative)	Medium	High	Low	Low

Table 2-3 A comparison of the more accessible simulation platforms based on [20][77][78]

While FlightGear is the lowest cost platform, it still requires external flight models driving its simulation. X-Plane offers its independent flight model generation and is relatively inexpensive. While MATLAB/Simulink offers additional premium packages that can speed up the development process, it ends up increasing the software cost for the end user. The basic MATLAB/Simulink package however offers similar functionality as long as the end user writes custom code for it. As such two independent simulation solutions can be constructed by using basic MATLAB/Simulink and X-Plane.

2.8.2.2 X-Plane:

X-Plane is a flight simulation software package that utilises the Blade Element Theory for predicting aircraft behaviour based on the geometry and aerofoil data [77]. It was originally designed with the intention of simulating flights of commercial/regular sized aircraft. As a result, the simulation comes pre-packaged with a collection of good models of such planes. What made X-Plane so popular for research application is that it allows the users to go beyond simulating just the default stock of aircraft.

As developer tools, X-Plane offers the applications Plane Maker and Airfoil Maker [79]. Plane Maker is used to define the aircraft geometric specifications along with various system specifications. Airfoil Maker is used to generate aerodynamic data based on specific aerofoil shape for specific Reynolds number regimes. While the default collection of aerofoil files (.afl format) cover a lot of the standard NACA aerofoils [80] used in aviation, it lacks appropriate Reynolds number ranges. But in principle, any aerofoil file can be interpolated with custom data to make up for this lack of lower Reynolds number aerofoil files. The simulator is also capable of exporting flight data to text files or sends them over a network if needed. It can use the networking features to both send and receive data.

These features are what led to its extensive use in the research community for applications such as Multi-UAV simulations [81], Autopilot simulations and Controller test platform [82], Software-In-The-Loop simulations [83], Performance and Stability testing [84], etc. Over the years, it's proven useful and reliable enough that NASA developed a set of software tools to interact with the simulator via various different programming platforms [85]. Despite all the great utility, there are various challenges while modelling in X-Plane. Below are some of the important ones considered.

- While the Plane Maker application allows for custom designs, its user interface and various limitations were developed with full scale aircraft in mind. As such, the smaller the target aircraft becomes, the harder it becomes to model it in X-Plane.
- The lack of low Reynolds number aerofoils means that accurate modelling cannot be expected from X-Plane unless a way is found to create custom aerofoil files.
- The standard set of data that is shown in the data export window for flight data recording does not have any means of exporting all data. It lists only the ones it does while X-Plane's memory holds additional useful parameters labelled as "dataref" [79]. Without finding means to read these "dataref" at the correct sampling frequency, the type of parameter estimation work described earlier becomes unreliable.
- The output from "dataref" is limited to the simulation frame rate. This can create problems with sampling (e.g. If high frequency data is required). This is also a problem if the data must be evenly spaced as no two frames require exactly the same time to render in X-Plane.

It is not uncommon to find publications on small autonomous flying platforms that utilised X-Plane for demonstration of the developed systems. However, many of these publications focus on the system element and ignore the modelling fidelity issues of the small UAVs. Which is appropriate for proof of the concepts being explored but does little to standardise a reliable way to appropriately capture the flight dynamics of the small UAV.

However, compared to the high-end and expensive flight simulators such as Merlin [86], X-Plane still allows the greatest amount of flexibility and development opportunity to the highest number of potential researchers at the lowest price. All the aforementioned benefits make X-Plane the ideal candidate for the development of an independent simulation (besides MATLAB/Simulink) as long as the mentioned challenges are addressed adequately.

2.9 References

[1] D. Raymer (2012), Aircraft Design: A Conceptual Approach (5th Edition), pg 602.

[2] R. C. Nelson, Flight Stability and Automatic Control, 2nd Edition, Singapore, McGraw-Hill, 1998, ch.3-5.

[3] R. W. Beard and T. W. McLain. Small Unmanned Aircraft: Theory and Practice. Princeton University Press, USA 2012.

[4] Rotation Matrix Overview: M.W. Spong and M. Vidyasagar, Robot Dynamics and Control. New York: John Wiley & Sons, Inc., 1989

[5] Rotation: B.L. Stevens and F.L. Lewis, Aircraft Control and Simulation. Hoboken, NJ: John Wiley & Sons, Inc., 2nd edition, 2003.

[6] D.T. Greenwood. Principles of Dynamics. Englewood Cliffs, NJ: Prentice Hall, 2nd ed., 1988.

[7] Ceccarelli, N., Enright, J. J., Frazzoli, E., Rasmussen, S. J., & Schumacher, C. J. (2007, July). Micro UAV path planning for reconnaissance in wind. In 2007 American Control Conference (pp. 5310-5315). IEEE.

[8] Hajiyev, C., Soken, H.E. and Vural, S.Y., 2015. *State estimation and control for low-cost unmanned aerial vehicles*. Cham, Switzerland: Springer.

[9] Marantos, P., Koveos, Y., & Kyriakopoulos, K. J. (2015). UAV state estimation using adaptive complementary filters. *IEEE Transactions on Control Systems Technology*, *24*(4), 1214-1226.

[10] Jategaonkar, R. V., & Plaetschke, E. (1989). Algorithms for aircraft parameter estimation accounting for process and measurement noise. *Journal of Aircraft*, *26*(4), 360-372.

[11] Mao, G., Drake, S., & Anderson, B. D. (2007, February). Design of an extended kalman filter for uav localization. In *2007 Information, Decision and Control* (pp. 224-229). IEEE.

[12] De Marina, H. G., Pereda, F. J., Giron-Sierra, J. M., & Espinosa, F. (2011). UAV attitude estimation using unscented Kalman filter and TRIAD. *IEEE Transactions on Industrial Electronics, 59*(11), 4465-4474.

[13] Murphy, P., Klein, V., Frink, N., & Vicroy, D. (2011, August). System identification applied to dynamic CFD simulation and wind tunnel data. In *AIAA Atmospheric Flight Mechanics Conference* (p. 6522).

[14] ILIFF, K., & MAINE, R. (1982, August). NASA Dryden's experience in parameter estimation and its uses in flight test. In *9th Atmospheric Flight Mechanics Conference* (p. 1373).

[15] Yong, T. H., & Dol, S. S. (2015, March). Design and development of low-cost wind tunnel for educational purpose. In *IOP Conference Series: Materials Science and Engineering* (Vol. 78, No. 1, p. 012039). IOP Publishing.

[16] Johnson, E. N., & Fontaine, S. (2001, August). Use of flight simulation to complement flight testing of low-cost UAVs. In *AIAA Modeling and Simulation Technologies Conference*.

[17] Kimberlin, R. D. (2003). *Flight testing of fixed wing aircraft*. Aiaa.

[18] Segui, M., Kuitche, M., & Botez, R. M. (2017). Longitudinal aerodynamic coefficients of hydra technologies UAS-S4 from geometrical data. In *AIAA Modeling and Simulation Technologies Conference* (p. 0579).

[19] Holsten, J., Ostermann, T., & Moormann, D. (2011). Design and wind tunnel tests of a tiltwing UAV. *CEAS aeronautical journal*, *2*(1), 69-79.

[20] Aerospace Blockset for Matlab, https://uk.mathworks.com/products/aeroblks.html, retrieved: 02/10/2017

[21] Triputra, F. R., Trilaksono, B. R., Sasongko, R. A., & Dahsyat, M. (2012, September). Longitudinal dynamic system modeling of a fixed-wing UAV towards autonomous flight control system development: A case study of BPPT wulung UAV platform. In *2012 International Conference on System Engineering and Technology (ICSET)* (pp. 1-6). IEEE.

[22] S-Function blocks, MathWorks, https://www.mathworks.com/help/simulink/sfg/what-is-an-s-function.html. Retrieved 02/05/2020

[23] Custom MATLAB Function block, Mathworks, https://www.mathworks.com/help/simulink/ug/creating-an-example-model-that-uses-a-matlab-function-block.html, retrieved 02/05/2020

[24] C Code S-Function, Mathworks, https://www.mathworks.com/help/simulink/create-cc-s-functions.html, retrieved 02/05/2020

[25] L. Ljung, *System Identification Toolbox User's Guide*. The Math Works, Inc, 2003.

[26] H. Shim, *Hierarchical Flight Control System Synthesis for Rotorcraft-based Unmanned Aerial Vehicles*. PhD thesis, University of California, Berkeley, 2000.

[27] B. Mettler, *Identification Modeling and Characteristics of Miniature Rotorcraft*. Kluwer Academic Publishers, 2003.

[28] G. Cai, B. M. Chen, and T. Lee, *Unmanned Rototcraft Systems*. Springer, 2011.

[29] J. H. Blakelock, Automatic Control of Aircraft and Missiles, 2nd ed., USA: John Wiley & Sons, 1991, ch. 1,3.

[30] Stevens, B. L., and Lewis, F. L., Aircraft Control And Simulation, 2nd edition, John Wiley & Sons, Hoboken, NJ, 2003.

[31] Phillips, W. F., Mechanics of Flight, John Wiley & Sons, Hoboken, NJ, 2004.

[32] Pamadi, B. N., Performance, Stability, Dynamics, and Control of Airplanes, 2nd edition, AIAA Education Series, Reston, VA, 2004.

[33] Roskam, J., *Aircraft Flight Dynamics and Automatic Flight Controls (Part I)*, DAR Corporation, Lawrence, KS, 2003.

[34] G. J. Mullen, Aircraft Parameter Identification Using Matlab, Volume 11 of Cranfield College of Aeronautics report, Cranfield University, 2000.

[35] Pallister K. Investigation of Lateral Stability Characteristics of Aeroplanes with Bluff Upswept Rear Fuselages. College of Aeronautics DCAe thesis, 1967.

[36] Martin, C.A. Further Investigation of the Lateral Stability Characteristics of an Aircraft with a Bluff Upswept Rear Fuselage. College of Aeronautics DCAe thesis,

1968.

[37] Klein, V. Evaluation of the Basic Performance Characteristics of an Instrumentation System. Cranfield Institute of Technology, College of Aeronautics Report, Aero. 21, 1973.

[38] Klen, V. Parameter Identification Applied to Aircraft. Cranfield Institute of Technology, College of Aeronautics, Phd thesis, 1973.

[49] Baek, Y-H. An Experimental Review of Some Aircraft Parameter Identification Techniques. Cranfield University, College of Aeronautics, PhD thesis, 1998.

[50] M.V. Cook, Flight Dynamics Principles. New York: John Wiley & Sons, 1997

[51] T. R. Yechout, S. L. Morris, D. E. Bossert, and W. F. Hallgren, Introduction to Aircraft Flight Mechanics . AIAA Education Series, American Institute of Aeronautics and Astronautics, 2003.

[52] H. Goldstein, Classical Mechanics . Cambridge, MA: Addison-Wesley, 1951.

[53] A. V . Rao, Dynamics of Particles and Rigid Bodies: A Systematic Approach . Cambridge: Cambridge University Press, 2006.

[54] J. B. Marion, Classical Dynamics of Particles and Systems . New York: Academic Press, 2nd edition, 1970.

[55] W. E. Wiesel, Spaceflight Dynamics . New York: McGraw Hill, 2nd ed., 1997.

[56] J. R. Wertz, ed., Spacecraft Attitude Determination and Control . Dordrecht, Neth.: Kluwer Academic Publishers, 1978.

[57] R. F. Stengel, Flight Dynamics . Princeton, NJ: Princeton University Press, 2004.

[58] Batchelor, G.K. (2000). An Introduction to Fluid Dynamics. Cambridge: Cambridge University Press. ISBN 978-0-521-66396-0.

[59] M. B. Tischler and R. K. Remple, Aircraft and Rotorcraft System Identification: Engineering Methods with Flight Test Examples. Virginia: AIAA, Inc, 2012

[60] K. J. Keesman, System Identification: An Introduction. London: Springer-Verlag, 2011.

[61] R. Isermann and M. M ¨unchhof, Identification of Dynamic Systems: An Introduction with Applications. Springer, 2010.

[62] L. Ljung, System Identification-Theory for User, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.

[63] S. Morris and M. Holden, “Design of micro air vehicles and flight test validation,” Proceedings of the Fixed, Flapping and Rotary Wing Vehicles at Very Low Reynolds numbers Conference, pp. 153–176, 2000.

[64] Shyy, W., Lian, Y., Tang, J., Viieru, D., & Liu, H. (2008). Aerodynamics of low Reynolds number flyers.

[65] Vural, Murat, and L. M. Nicolai. "Estimating R/C model aerodynamics and performance." *Illinois Institute of Technology* (2009).

[66] N. Hoffer, C. Coopmans, A. Jensen, and Y. Chen, "Small low-cost unmanned aerial vehicle system identification: A survey and categorization" , in Unmanned Aircraft Systems (ICUAS), 2013 International Conference on, 2013, pp. 897-904.

[67] Lomax, H., Pulliam, T. H., Zingg, D. W., Pulliam, T. H., & Zingg, D. W. (2001). *Fundamentals of computational fluid dynamics* (Vol. 246). Berlin: Springer.

[68] Simcenter Star CCM, https://www.plm.automation.siemens.com/global/en/products/simcenter/STAR-CCM.html, retrieved 10/07/2021

[69] Ansys, https://www.ansys.com/en-gb/products/fluids retrieved 10/07/2021

[70] SolidWorks Flow Simulation, https://www.solidworks.com/product/solidworks-flow-simulation, retrieved 10/07/2021

[71] D. Mark, XFoil, http://web.mit.edu/drela/Public/web/xfoil/, retrieved 23/05/2020

[72] XFLR5, http://www.xflr5.tech, retrieved 23/05/2020

[73] Katz, A. Plotkin, Low-Speed Aerodynamics, 2nd ed., Cambridge University Press, Cambridge, 2001.

[74] G.D. McBrain, ch 11 (Lifting line theory), Theory of Life: Introductory Computational Aerodynamics in MATLAB/OCTAVE, John Wiley & Sons, Ltd, 2012

[75] T. Shafer, S. Vilken, N.M.Favaregh, C.H. Zeune, N. Williams, J. Dansie, Comparison of Computational Approaches for Rapid Aerodynamic Assesment of Small UAVs, AIAA 2014-0039, Session: Low Speed Low Reynolds number Aerodynamics 1, 2014

[76] Results vs Predictions, XFLR5, http://www.xflr5.tech/docs/Results_vs_Prediction.pdf, retrieved 23/05/2020

[77] X-Plane Simulator, https://www.x-plane.com/, last accessed 17/05/2021

[78] FlightGear Simulator, https://www.flightgear.org/, last accessed 17/05/2021

[79] Developers, X-Plane, https://developer.x-plane.com/docs/aircraft/, retrieved 17/05/2021

[80] Allen, Bob. "NACA Airfoils". nasa.gov. NASA. Retrieved 13 Feb 2020.

[81] Garcia, R., & Barnes, L. (2009). Multi-UAV simulator utilizing X-Plane. In *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009* (pp. 393-406). Springer, Dordrecht.

[82] Ribeiro, L. R., & Oliveira, N. M. F. (2010, October). UAV autopilot controllers test platform using Matlab/Simulink and X-Plane. In *2010 IEEE Frontiers in Education Conference (FIE)* (pp. S2H-1). IEEE.

[83] Bittar, A., Figuereido, H. V., Guimaraes, P. A., & Mendes, A. C. (2014, May). Guidance software-in-the-loop simulation using x-plane and simulink for uavs. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 993-1002). IEEE.

[84] Thong, C. W. (2010). Modeling aircraft performance and stability on x-plane. *Australian Defence Force Academy, Univ. of New South Wales, Canberra, Australia*.

[85] X-Plane Communication Toolbox, NASA, https://software.nasa.gov/software/ARC-17185-1, retrieved 17/05/2021

[86] Merlin Flight Simulator, http://www.merlinsim.com/, retrieved 17/05/2021

3 Methodology

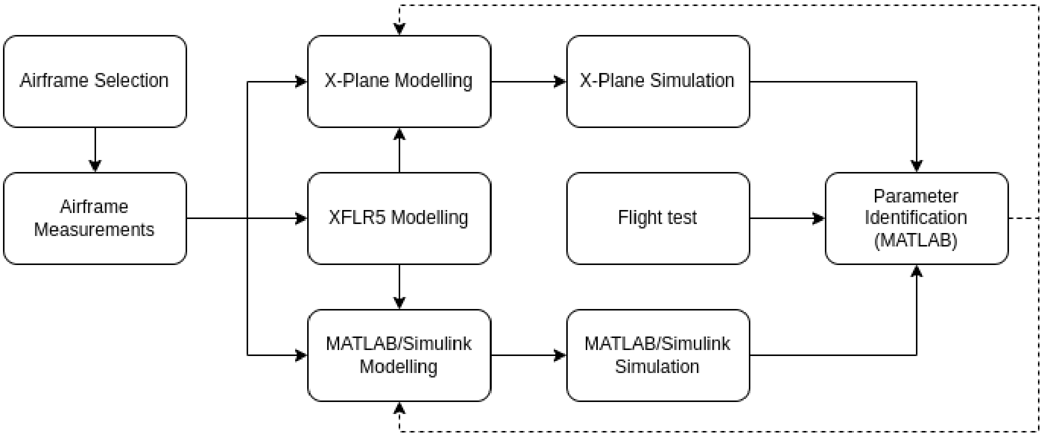


Figure 3.1 An overview of the proposed methodology

The methodology undertaken for this work follows a specific workflow that is detailed in the following subsections. The above block diagram (Figure 3.1) is an illustration of the proposed workflow undertaken for this research. The process starts with determination of an appropriate airframe and relevant key measurements (Chapter 3.1). This then aids in the creation of all the XFLR5 models detailed in Chapter 3.2, X-Plane models detailed in Chapter 3.3 and MATLAB/Simulink models detailed in Chapter 3.4. In the subsequent parts of the process, simulations are setup based on these models and flight tests are proposed. Based on the Parameter Estimation principles discussed in the literature review (Chapter 2.6), a method of extracting useful parameters from these flight tests is proposed and utilised (Chapter 3.3.14 & 4.2.5-4.2.7). As the dashed lines in the diagram show, these outputs can then be utilised to improve the models. This is a simplified overview of the overall methodology and the details are to be found in the subsequent chapters. The outputs, utilities and advantages/limitations of these methods are detailed in Chapters 4 and 5.

3.1.1 Target Airframe

For the purpose of this investigation, various different fixed-wing platforms were considered. For all the undertaken modelling and simulation work and proposed flight tests, a target fixed-wing platform needed to be acquired. The following are some of the important considerations made in that selection process.

- The aircraft configuration be Fixed Wing

- The aircraft must be capable of carrying any on-board autopilot system along with the sensors (ex. GPS, IMU, Sonar, Camera, etc.)
- The aircraft must be light and small enough to be easily transported or be worked on during various phases of the project.
- The design configuration of the aircraft be as stable of a design as possible. The study could have been done using unstable designs too. However, doing that may result in unnecessary system complexity and/or increased risks in real-world flying tests.
- Reasonably priced
- Accessible Data Sheets

The mathematical tools discussed in the chapters involving the dynamic modelling of the UAS along with modelling and simulation tools considered made the finalisation of the UAS platform as the obvious first thing to do for the initialising the modelling and simulation work. The FMS SkyTrainer 182 [1] was selected as the best candidate.

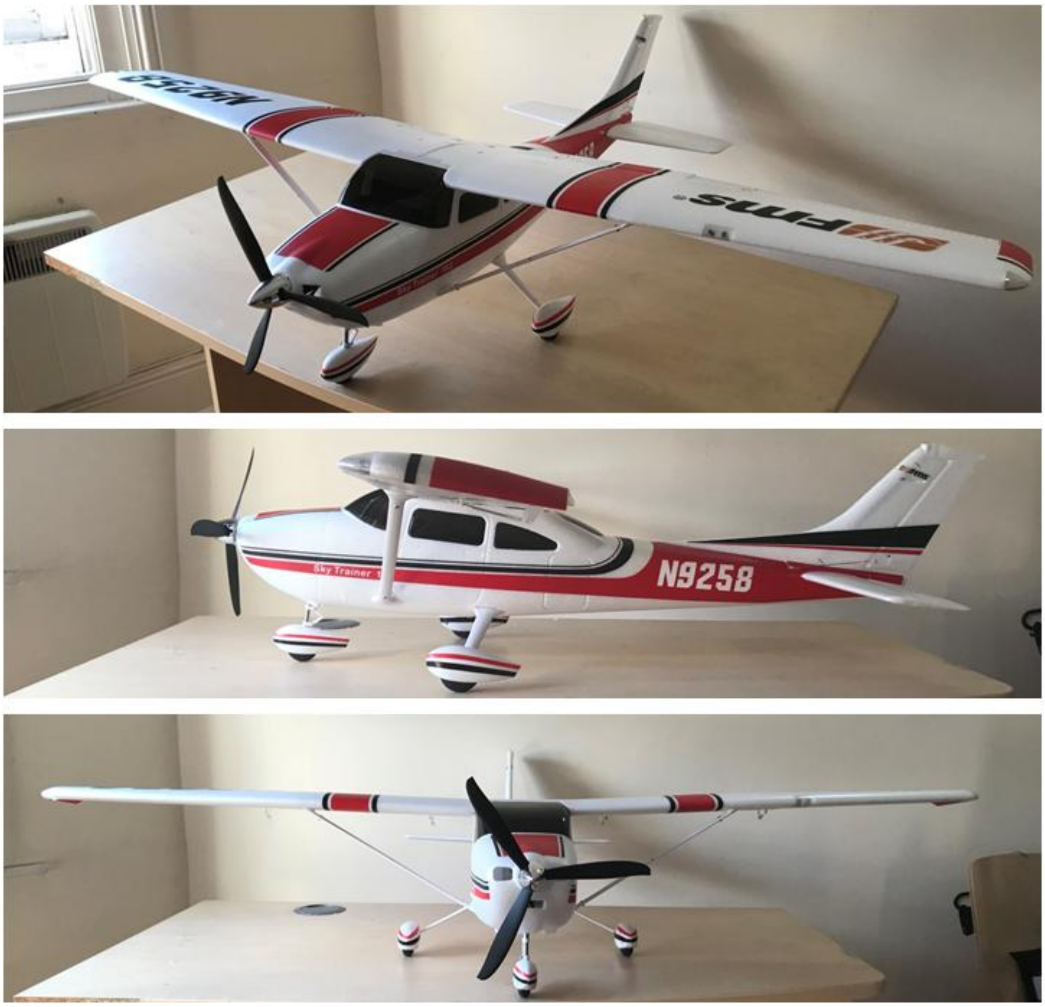


Figure 3.2 FMS 182 (aircraft used for this research)

This target airframe is based on the Cessna 182 [2] and is meant for RC Model flying. The following table [1] shows the specifications of this model plane.

Wingspan	1410mm / 55.5in
Length	1100mm / 43.3in
Flying Weight	1520g
Power System	3536-KV850 Brushless Motor
Speed Control	40A ESC with XT60 Connector
Propeller / EDF	11x6 Three Blade Propeller
Servos	9g x 6
Landing Gear	Fixed main gear with steerable nose
Required Battery	3 Cell 11.1V 2200 to 2600mAh 25C LiPo with XT60 Connector
Required Radio	5 Channel
Rudder	Yes
Flaps	Yes
Ailerons	Yes
Lights	Yes
Hinge Type	Foam
Material	EPO Foam

Table 3-1 Available data on the FMS 182 [1]

Given that the manufacturers (FMS in this case) and the vendors retailing such units are targeting RC flying enthusiasts who simply wish to fly the aircraft recreationally, the data made available on the model is inadequate for modelling and simulation purpose. Even the basic specifications presented above didn't remain the same following certain modifications and inclusions made to the fuselage of the aircraft.

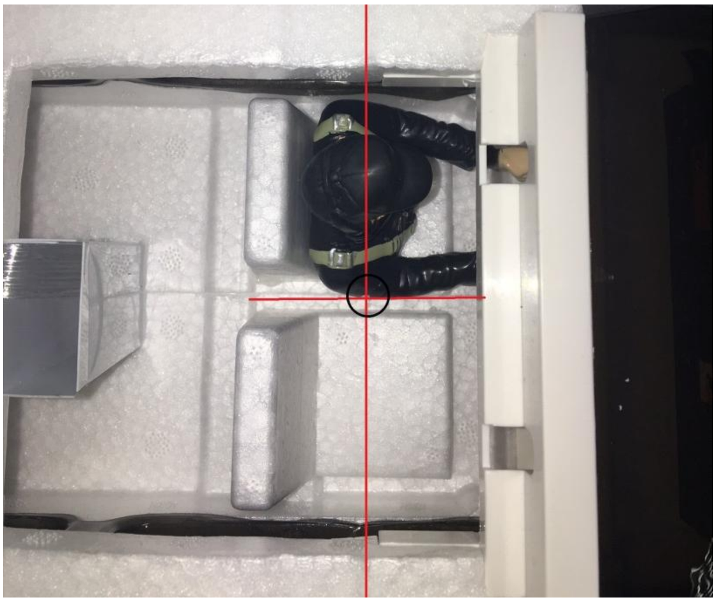


Figure 3.3 The red lines in the picture intersect at the recommended C.G. position

There were model seats and a pilot inside the cockpit where the recommended C.G. (60mm from leading edge of the wing) is meant to be. While being appropriate for model flyers, this was determined to be extra weight and as such considered for removal.

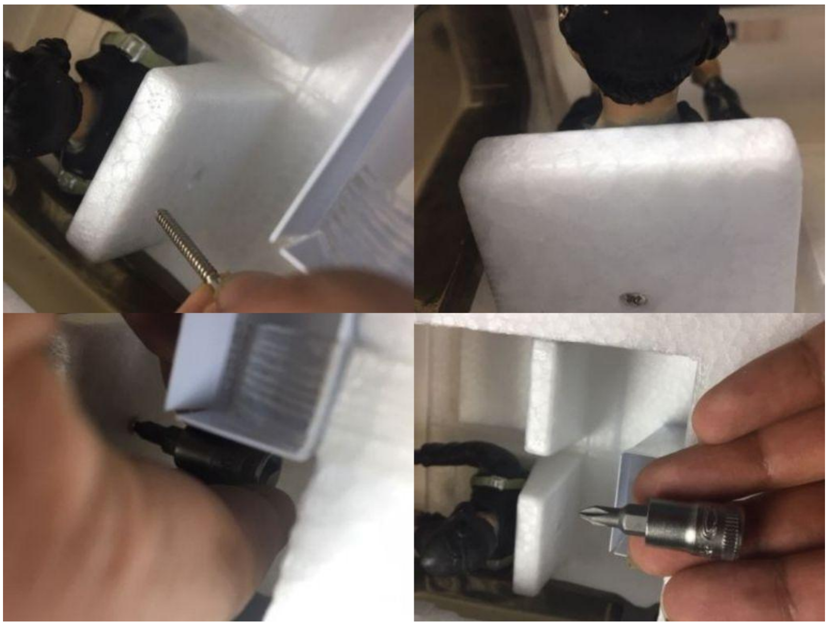


Figure 3.4 Removal of the dummy pilot from the cockpit

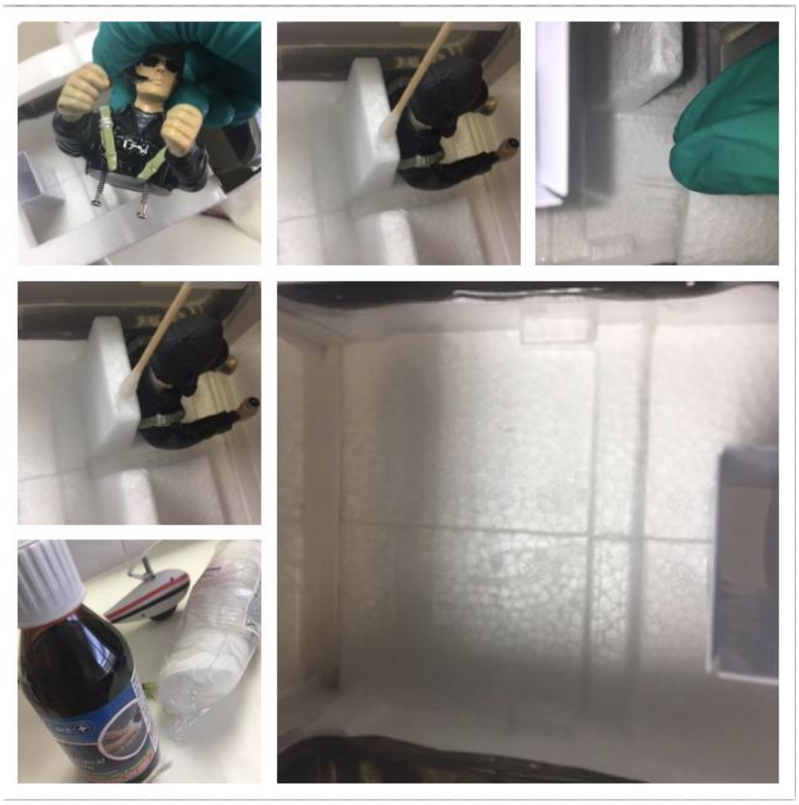


Figure 3.5 Use of methyl salicylate for removal process

There were also concerns about any extra space needed for Flight Controllers, peripherals, on-board sensors, batteries etc. Placing them at a distance from the CG line shown in the image creates undesired moments about the CG. It is therefore advantageous to not only get rid of extra mass, but also create usable space in that region. These installations were removed carefully via the use of a screwdriver, Methyl Salicylate and cotton buds. The surgical spirit (Methyl Salicylate) was used to gradually soften the Hot-melt adhesive by repeatedly applying it on the glued edges with a cotton bud. The process takes a while but eventually all unnecessary components were removed by it.

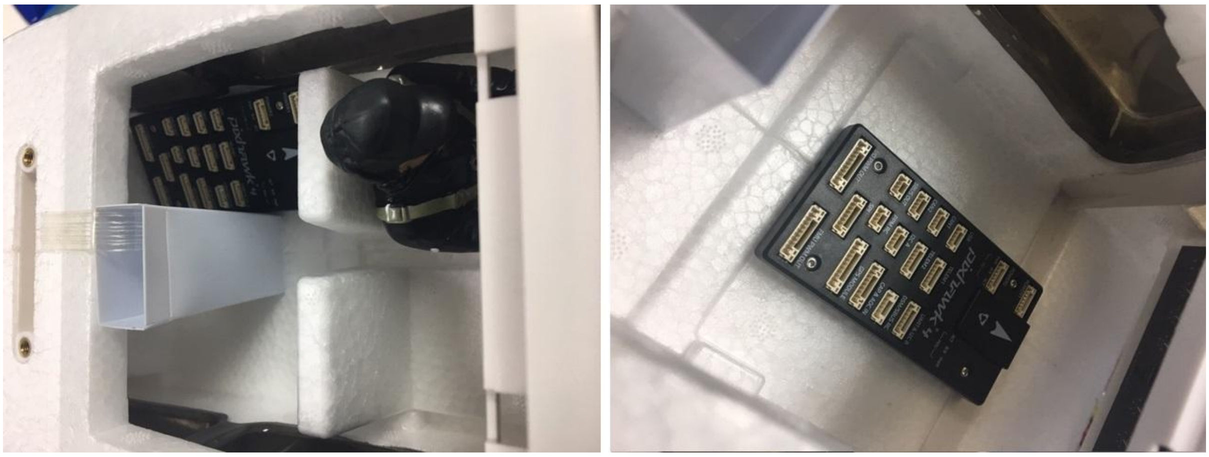


Figure 3.6 Space freed inside the cockpit of the FMS 182 for avionics

As seen in the above image, this process resulted in a reasonable amount of space being freed up for any other electronics the airplane may have to carry when in an autonomous mission. The removed components were not weighed because all necessary measurements for the aircraft was done in detail for a different configuration as detailed in the following sections.

3.1.2 Geometric and Inertial Measurements of the Target Airframe

The aircraft and its components were measured both while fully assembled and dismantled. In the absence of adequate and accurate data on the parameters of our interest, the following methods were used to aid in the XFLR5 and X-Plane modelling process.

3.1.2.1 Component Measurement:

The aircraft was sectioned off into the following major components:

- Horizontal Stabiliser
- Vertical Stabiliser 1
- Vertical Stabiliser 2
- Main Wing
- Fuselage
- Motor (including the mount)
- Propeller (including the mount)
- Left Landing Gear
- Right Landing Gear
- Front Landing Gear
- Wing Struts
- Radio Receiver

XFLR5 and X-Plane [3][4] uses different means and information for their respective modelling processes. As such, the measurement process was carried out such that we may derive an appropriate dataset on this particular aircraft that may aid in the modelling process. The following details the processes used. It should be noted that the Vertical Stabilizer has been split into two projections (Vertical Stabilizer 1 & Vertical Stabilizer 2) because of different sets of modelling simplification techniques that were needed to model this aircraft appropriately in different platforms.

3.1.2.1.1 Main Wing

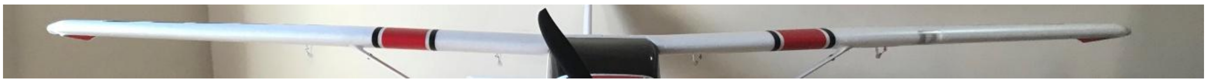


Figure 3.7 FMS 182 Main Wing

The main wing is composed of two geometrically symmetric half-wings that meet on the midpoint of the fuselage. Both sections produced identical measurements for all parameters of interest and as such only one wing information is presented.

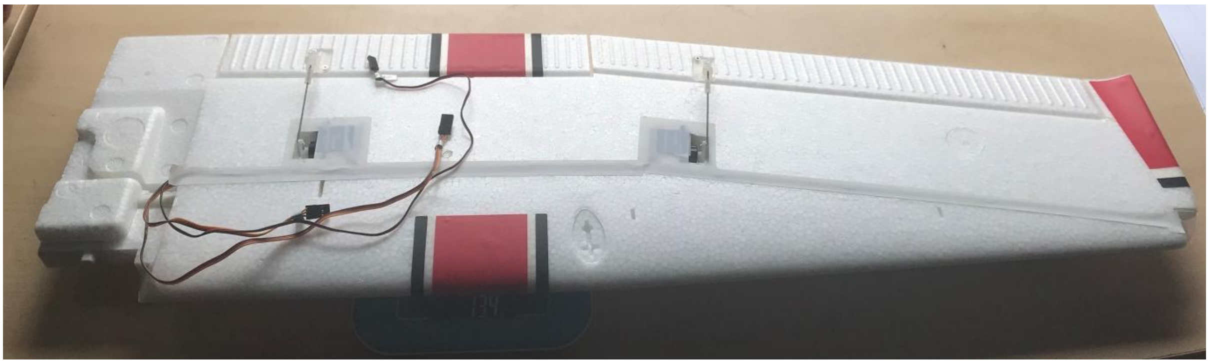


Figure 3.8 FMS 182 starboard wing bottom side

The right wing (starboard) was weighed on a digital scale to be 134 grams. It should be noted that this weight includes the weight of the two 9 grams servo installations. As such, the weight of this wing was recorded as 116 grams.

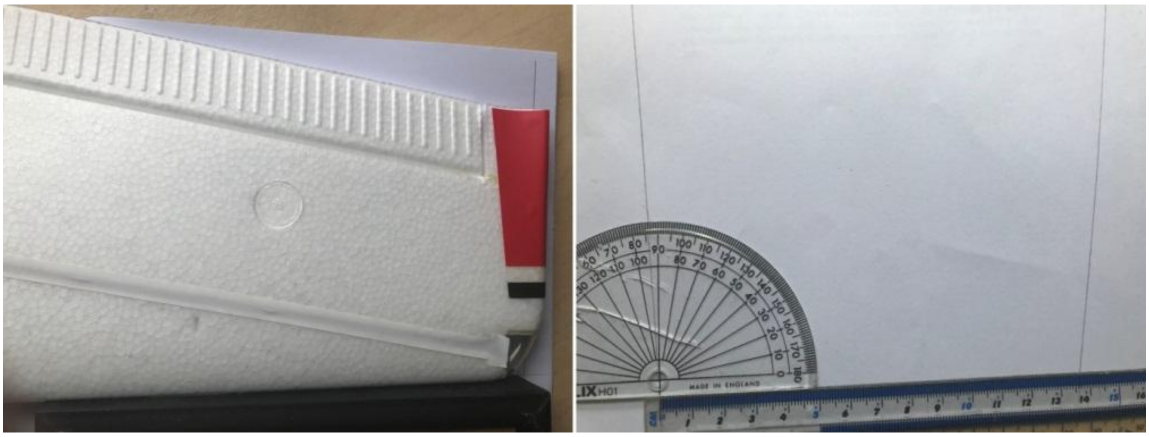


Figure 3.9 Use of tracing to record wing geometry

Using a square block for alignment, the wing tip profile was traced onto an A4 sheet of paper and measured. From the outer section of the wing, the sweep was measured to be 4 degrees with a protractor. The tip chord was obtained from the tracing by measuring it with a ruler and found to be 137 mm. The wing offset was calculated with Equation 3.1.

$$\tan(sweep\ angle) = \frac{offset}{(half - span)} \qquad \text{[Eq. 3.1]}$$

$$\tan(4) = \frac{offset}{(705 - 322)}$$

$$offset = 26.782\ mm$$

Important positions in the wing were marked and their positions from the wing tip and root were measured using a measuring tape and a triangle. The inner-wing (with zero taper) spans 322 mm from the centreline (wing root). This also implies that the tapering point starts at 322 mm from the wing root. The inboard servo (left in the picture) has been measured to 159 mm from the wing root (120 mm for the other wing), 110 mm from the leading edge of the wing and positioned 5 mm below the chord. The outboard servo (right in the picture) has been measured to be 372 mm away from the centreline (wing root), 100 mm from the leading edge of the wing and in the plane of the root chord (z=0). Part of the inner-section of this wing that covers the fuselage is 73mm in span.

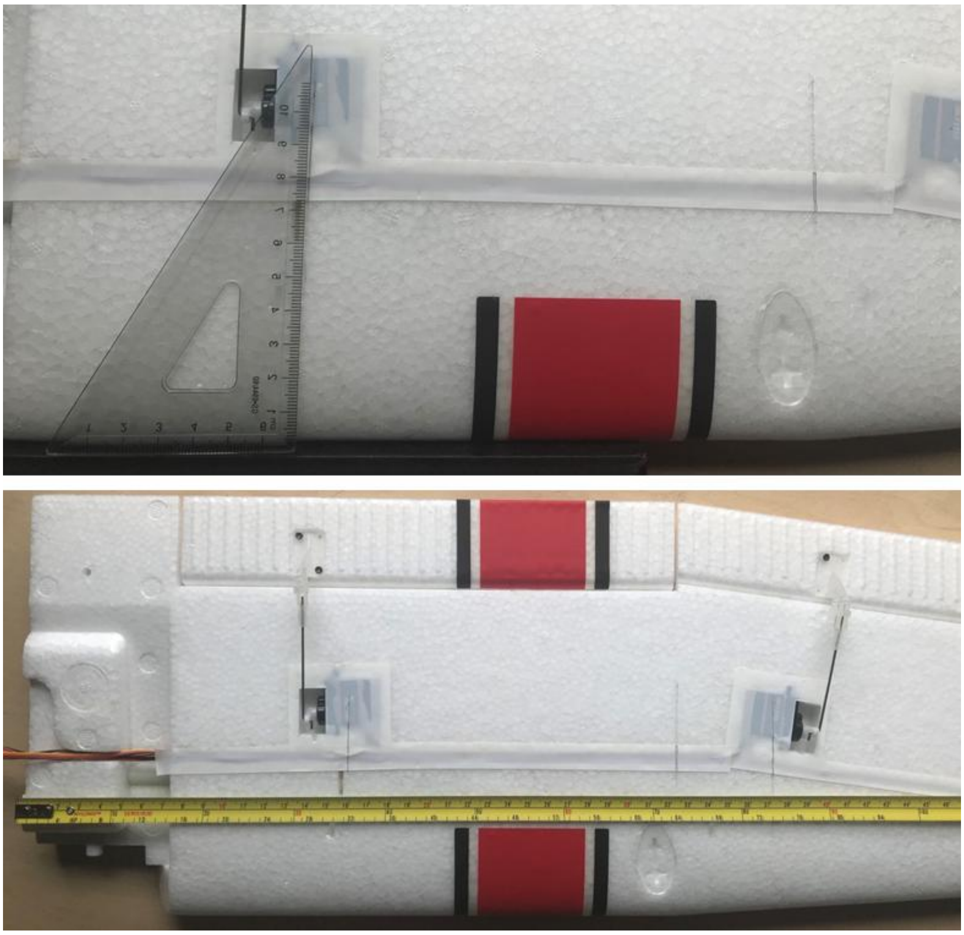


Figure 3.10 Measuring and marking the specific locations of components on the wing

Following the same methods as before, the inner wing geometry was measured with a ruler after tracing it on to an A4 sheet of paper via the use of markers and a square block. The wing root chord was measured to be 210 mm and the flap was measured to be positioned 163 mm from the leading edge (77.619% of the chord). The flap was also measured to be 47 mm from the trailing edge, starting at 73 mm from the centreline (wing root) and ending 320 mm from the centreline (wing root). The aileron starts from 168 mm measured from the leading edge of the wing (80% of the chord). Measured from the trailing edge, it starts at 42 mm. From the centreline, it's 323 mm and ends 671 mm from the centre (wing root).

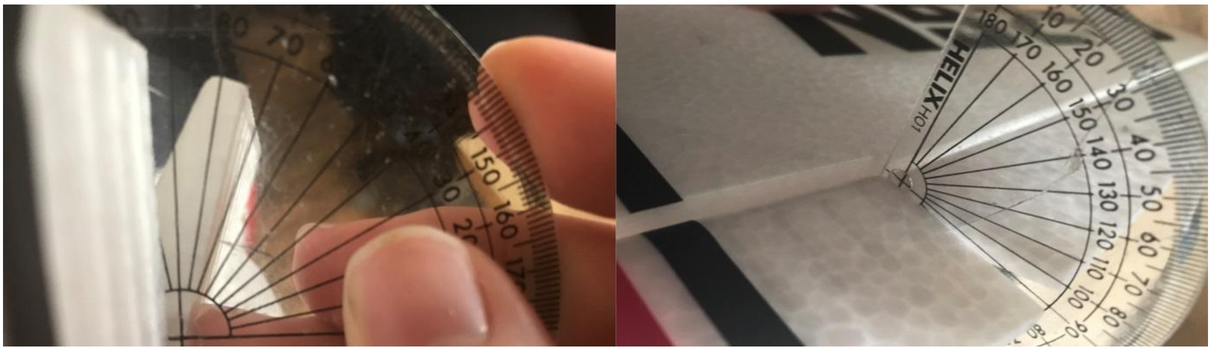


Figure 3.11 Control surface deflection measurement with protractor

The aileron and flap deflections were both measured at 15 degrees. For the flap, this deflection only happens in the downward direction (+15 degrees) and for the aileron it happens in both directions (+ and - 15 degrees). To obtain these control surface deflection angles, the protractor was first vertically aligned to the bottom of the wing (as a reference point) and subsequently translated until it lined up its reference point with the deflective surfaces.

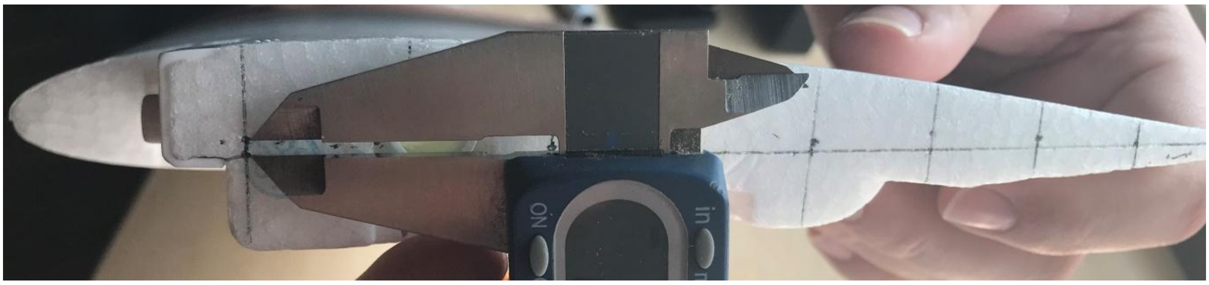


Figure 3.12 Measurement of wing section for aerofoil determination

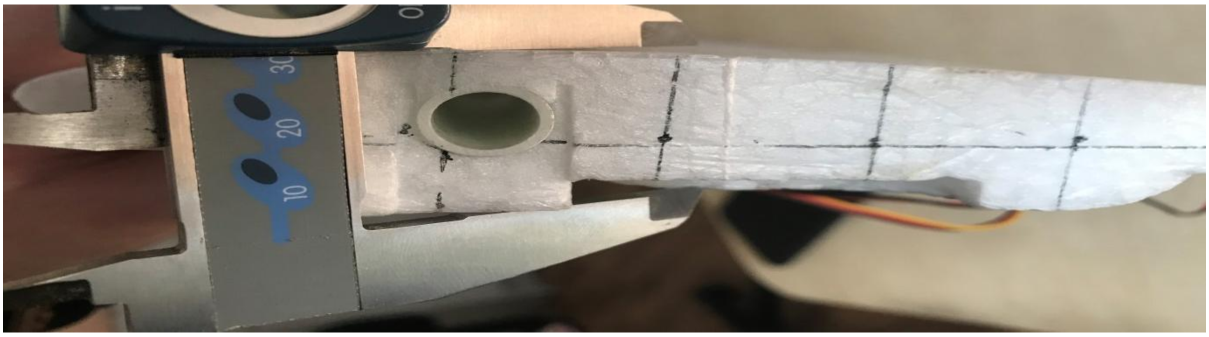


Figure 3.13 Wing thickness measurement using callipers

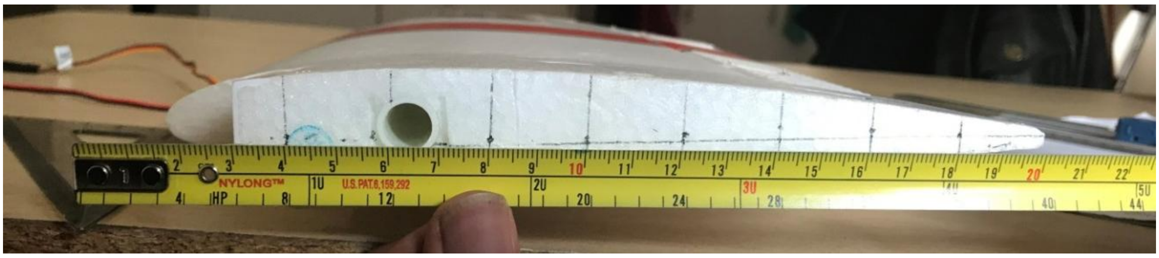


Figure 3.14 Root chord measurement

With a calliper, markers and measuring tape, a maximum camber of 2 mm (equating to 1% of the chord length) at the 20% interval along the chord line (210 mm) was measured. Maximum thickness of the wing section was measured to be 28.58 mm. This equates to around 14% of the chord length. By standard NACA 4-digit aerofoil designation convention [5], this identifies it as a NACA 1214 aerofoil.

This process of using simple tools and methods were used to measure every relevant detail of the FMS 182. The techniques used for all other specific components measured are well illustrated and explained in the Appendix (Chapter 7.5.1). The following sections summarise the measurements obtained for the components.

3.1.2.1.2 Vertical Stabiliser 1 & 2

Vertical Stabiliser 1		**Vertical Stabiliser 2**	
Span (mm)	195	Root chord (mm)	360
Root chord (mm)/Ref. chord	174/127.39	Tip Chord (mm)	242
Tip Chord (mm)	86	Max thickness (mm)	21.82
Ref. Max thickness (mm)	15.28	Relative sweep (degrees)	37.5
Offset	149.6	Total sweep (degrees)	88.5
Foil Designation	NACA 0012		
		Total Vertical Stabiliser weight (grams)	25
Rudder			
Distance from wing root (mm)	70		
Distance from wing tip (mm)	43		
Span (mm)	195		
Tip Chord (mm)	86		
Trailing tip location (mm)	822		
Leading edge location (mm)	586.37		
Offset	149.629		
Foil Designation	NACA 0012		

Table 3-2 Vertical Stabiliser measurements summary

3.1.2.1.3 Main Wing

Right wing weight with 2 servos (grams)	134	Wing section covering fuselage (mm)	73
servo weight (grams)	9	Wing root chord (mm)	210
Right wing weight w/o servos (grams)	116	Flap position from leading edge (mm)	163
Sweep from outer section (degrees)	4	Flap position from trailing edge (mm)	47
Tip chord (mm)	137	Flap starting point from root (mm)	73
Offset (mm)	26.782	Flap ending point from root (mm)	320
Inner wing span (mm)	322	Aileron starting point from L.E.(mm)	168
Tapering point from root (mm)	322	Aileron starting point from T.E (mm)	42
Inboard servo from root (mm)	159	Aileron start location from root (mm)	323
Inboard servo from leading edge (mm)	110	Aileron end location from root (mm)	671
Inboard servo below chord (mm)	5	Aileron deflection (degrees)	15
Outboard servo from root (mm)	372	Flap deflection (degrees)	15
Outboard servo from leading edge (mm)	100	Wing foil designation	NACA 1214
Outboard servo from chord(mm)	0		

Table 3-3 Main Wing measurement summary

3.1.2.1.4 Fuselage

Total Weight (grams)	649
Centre of gravity location from nose (mm)	316.75
Nose offset (mm)	204
C.G. located aft of wing reference (mm)	112.75
Diameter (m)	0.159
Height (m)	0.185
Width (m)	0.137
Length (m)	0.944

Table 3-4 Fuselage measurement summary

3.1.2.1.5 Radio receiver and Motor:

Radio Receiver Weight (grams)	16
Motor (including mount) weight (grams)	186
Motor x-C.G. behind nose (mm)	45
Motor position x-y-z (mm)	204-0-0

Table 3-5 Measured data for the receiver and motor

3.1.2.1.6 Propeller

The following table summarises the measurements taken for the propeller.

Propeller Weight (grams)	36
Position from nose (mm)	15
Position ahead of wing (mm)	219
Position from x-axis (mm)	0
Position below wing tip (mm)	75

Table 3-6 Propeller measurement summary

3.1.2.1.7 Landing Gears

Left landing gear			
Left Landing Gear Weight (grams)	47	**Front landing gear**	
Right Landing Gear Weight (grams)	41	Weight (grams)	37
C.G. positions of Right Landing Gear		**C.G. position**	
x (mm)	125	x (mm)	100
y (mm)	142.5	y (mm)	0
z (mm)	200	z (mm)	200

Table 3-7 Landing Gears measurement summary

3.1.2.1.8 Wing Struts

Weight (grams)	25
C.G. positions	
Left strut x-position (mm)	25
Left strut y-position (mm)	190
Left strut z-position (mm)	77.5

Table 3-8 Wing struts measurements summary

Similar measurement techniques as shown previously were utilised to measure all relevant physical detail of the FMS 182. The tables provided here only summarises the key measurements in the absence of the context and the actual techniques used to obtain them. Those details are documented in full detail in the Appendix (Chapter 7.5)

3.2 XFLR5 Methods

3.2.1 Aerofoil Creation

To investigate the aerodynamic behaviour of our model in XFLR5, the aerofoils had to be identified as previously shown and modelled. The three primary foils were the NACA 1214 (main wing), NACA 0010 (horizontal stabiliser) and NACA 0012 (vertical stabiliser). These primary foils had to be modified to model the representative foils over the sections of the modelled surfaces where various control surfaces exist. As such, the following individual foils were necessary and generated.

- NACA 1214 FMS
- NACA 1214 FMS 15 deg aileron down
- NACA 1214 FMS 15 deg aileron up
- NACA 1214 FMS 15 deg flap down
- NACA 1214 FMS 0 aileron
- NACA 0010 FMS
- NACA 0010 FMS down
- NACA 0010 FMS up
- NACA 0010 FMS 0 elevator
- NACA 0012 FMS
- NACA 0012 FMS Root down
- NACA 0012 FMS Root up
- NACA 0012 FMS Tip down
- NACA 0012 FMS Tip up
- NACA 0012 FMS 0 Root Rudder
- NACA 0012 FMS 0 Tip Rudder

All the aerofoils were generated with initial assignment of 200 panels and Global Refinement was used to ensure even panel distribution prior to analysis as it aids the solver.

3.2.1.1 Main Wing Aerofoils:

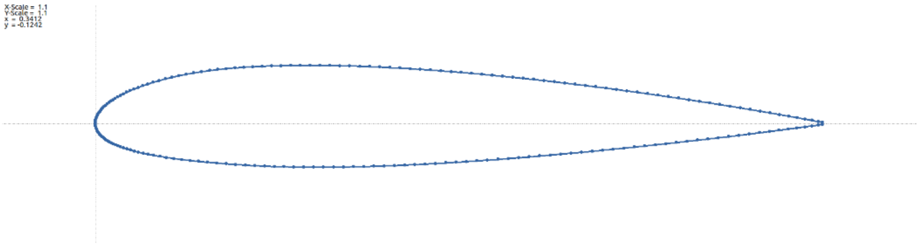


Figure 3.15 NACA 1214 FMS aerofoil

The standard NACA 1214 was used for the main wing section. However, due to the existence of the ailerons and flaps, additional modified foils were generated to capture the changed aerodynamic properties of based on control surface deflections.

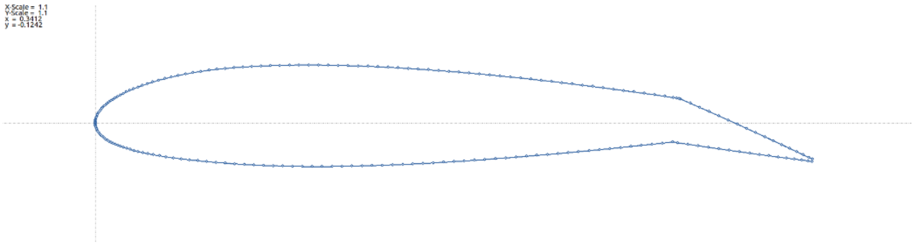


Figure 3.16 NACA 1214 FMS 15 degrees aileron down aerofoil

Based on the measurements, the foil representing the section over the aileron was modified to represent a maximum control surface deflection of 15 degrees at the previously measured hinge positions (expressed in terms of percentage of the chord).

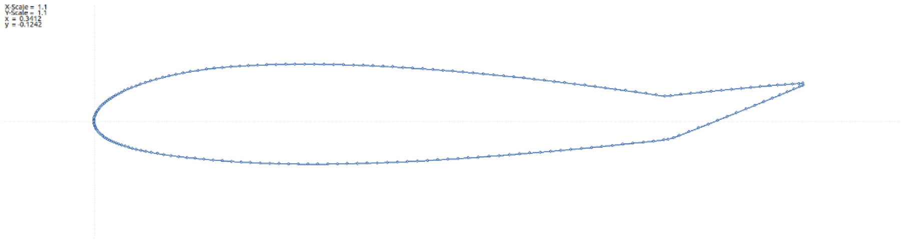


Figure 3.17 NACA 1214 FMS 15 degrees aileron up aerofoil

The same was done for the aileron's hinge positions and deflection in the opposite direction (represented with -15 degrees). An extra version of the foil was generated for the flaps with a downward deflection of 15 degrees but different hinge locations.

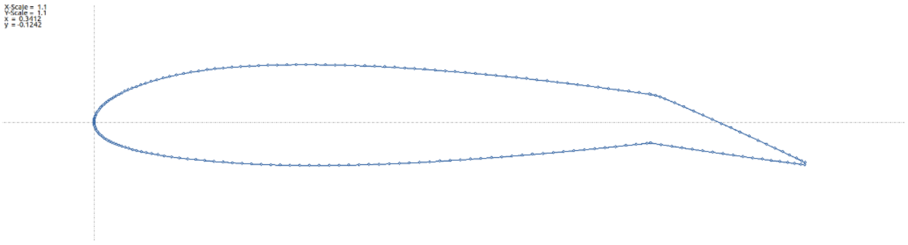


Figure 3.18 NACA 1214 FMS 15 degrees flaps down aerofoil

3.2.1.2 Horizontal Stabiliser Aerofoils

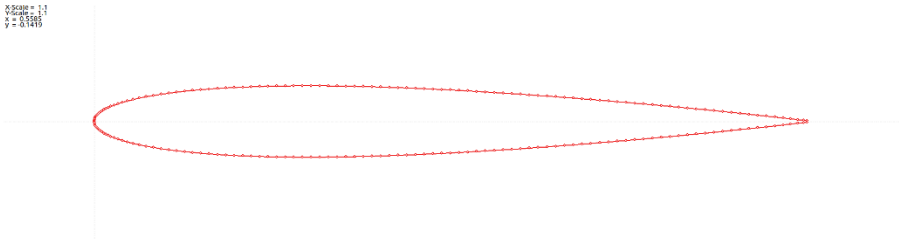


Figure 3.19 NACA 0010 FMS aerofoil

Based on the measurements, the standard NACA 0010 aerofoil was used for the symmetrical horizontal stabiliser. However, this aerofoil needed modifications to represent the control surface deflections of the elevator. As such, two further variant foils were created to represent the positive and negative maximum deflections.

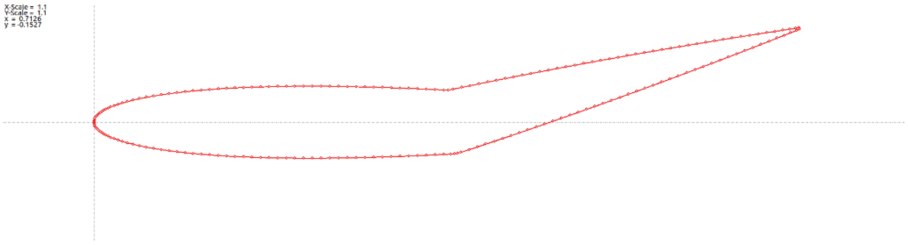


Figure 3.20 NACA 0010 FMS 15 degrees elevator up aerofoil

Based on the previous measurements, the NACA 0010 foil was modified to have the appropriate hinge position for the elevator along with a 15 degree upward deflection.

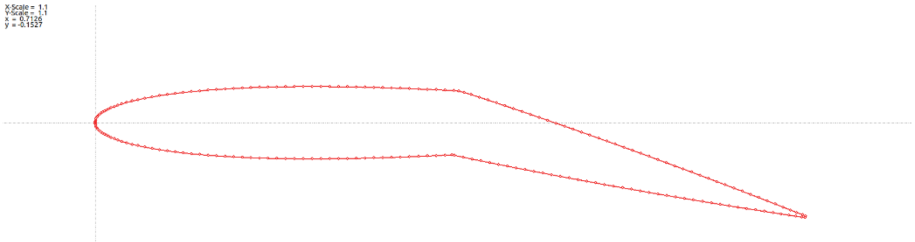


Figure 3.21 NACA 0010 FMS 15 degrees elevator down aerofoil

The same procedure was followed for the foil representation for the downward deflection of the elevator control surface. Since the deflection angle was the same but in the opposite direction, the same magnitude of 15 degrees was used with the sign inverted to match XFLR5's conventions.

3.2.1.3 Vertical Stabiliser Aerofoils:

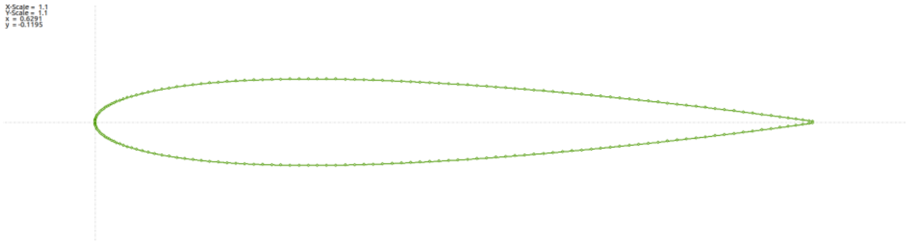


Figure 3.22 NACA 0012 FMS aerofoil

Based on the previous measurments, the symmetrical aerofoil for the vertical stabiliser was created as the standard NACA 0012. The representation of the aerofoil with both positive and negative control surface deflection was handled differently for this particular aerofoil. This is due to the fact that the vertical offset of this section creates a situation where the hinge's root x-coordinate no longer coincides with the tip x-coordinate. That led to the following modelling choices.

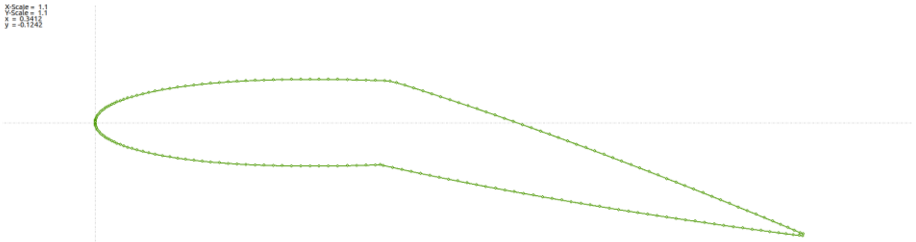


Figure 3.23 NACA 0012 FMS 15 degrees down up aerofoil

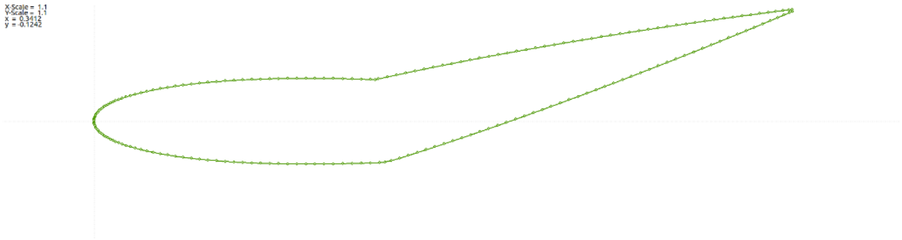


Figure 3.24 NACA 0012 FMS 15 degrees root up aerofoil

Based on the previous measurements of the root section of the vertical stabiliser, the hinge position was used to create a two variants of the aerofoil to be used for the stabiliser's root. Positive and negative 15 degrees were assigned as the maximum deflection angles for these surfaces.

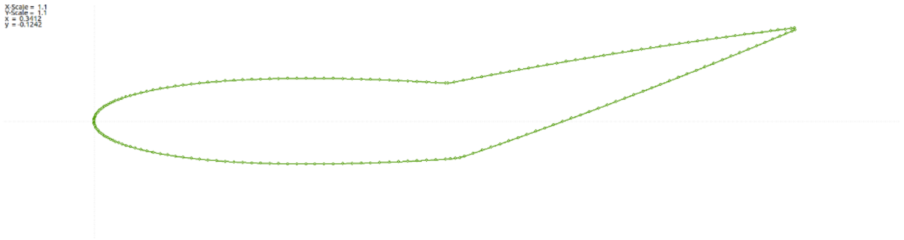


Figure 3.25 NACA 0012 FMS 15 degrees tip up aerofoil

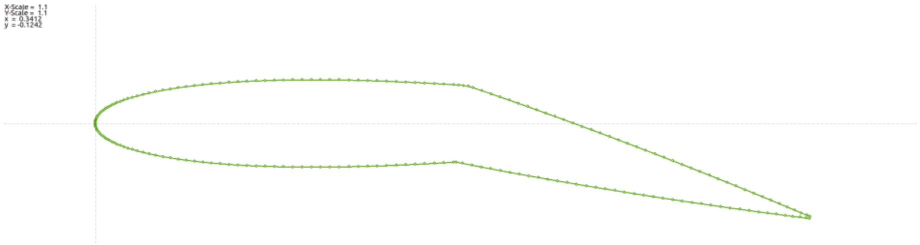
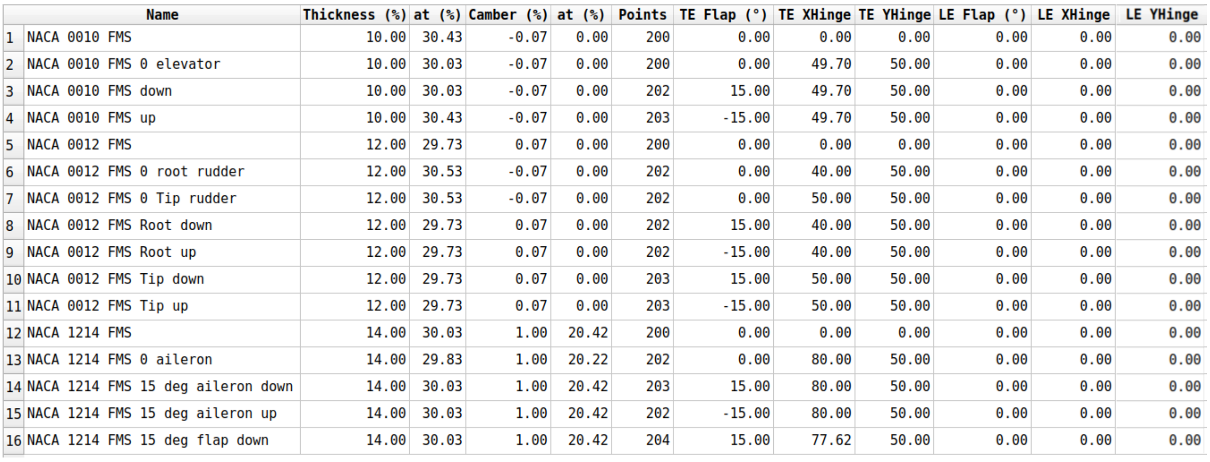


Figure 3.26 NACA 0012 FMS 15 degrees tip up aerofoil

The same process was used to generate the two foil variants needed to represent the aerofoil section at the tip of the vertical stabiliser. The same positive and negative 15 degrees of maximum control surface deflection was used. 3 extra foils with control surfaces defined but at zero deflection needed to be modelled for the control tests but their geometry were identical to that of the standard aerofoils (NACA 1214, 0012 and 0010). As such, they are not presented here. The following presents a summary of the aerofoil parameters.



	Name	Thickness (%)	at (%)	Camber (%)	at (%)	Points	TE Flap (°)	TE XHinge	TE YHinge	LE Flap (°)	LE XHinge	LE YHinge
1	NACA 0010 FMS	10.00	30.43	-0.07	0.00	200	0.00	0.00	0.00	0.00	0.00	0.00
2	NACA 0010 FMS 0 elevator	10.00	30.03	-0.07	0.00	200	0.00	49.70	50.00	0.00	0.00	0.00
3	NACA 0010 FMS down	10.00	30.03	-0.07	0.00	202	15.00	49.70	50.00	0.00	0.00	0.00
4	NACA 0010 FMS up	10.00	30.43	-0.07	0.00	203	-15.00	49.70	50.00	0.00	0.00	0.00
5	NACA 0012 FMS	12.00	29.73	0.07	0.00	200	0.00	0.00	0.00	0.00	0.00	0.00
6	NACA 0012 FMS 0 root rudder	12.00	30.53	-0.07	0.00	202	0.00	40.00	50.00	0.00	0.00	0.00
7	NACA 0012 FMS 0 Tip rudder	12.00	30.53	-0.07	0.00	202	0.00	50.00	50.00	0.00	0.00	0.00
8	NACA 0012 FMS Root down	12.00	29.73	0.07	0.00	202	15.00	40.00	50.00	0.00	0.00	0.00
9	NACA 0012 FMS Root up	12.00	29.73	0.07	0.00	202	-15.00	40.00	50.00	0.00	0.00	0.00
10	NACA 0012 FMS Tip down	12.00	29.73	0.07	0.00	203	15.00	50.00	50.00	0.00	0.00	0.00
11	NACA 0012 FMS Tip up	12.00	29.73	0.07	0.00	203	-15.00	50.00	50.00	0.00	0.00	0.00
12	NACA 1214 FMS	14.00	30.03	1.00	20.42	200	0.00	0.00	0.00	0.00	0.00	0.00
13	NACA 1214 FMS 0 aileron	14.00	29.83	1.00	20.22	202	0.00	80.00	50.00	0.00	0.00	0.00
14	NACA 1214 FMS 15 deg aileron down	14.00	30.03	1.00	20.42	203	15.00	80.00	50.00	0.00	0.00	0.00
15	NACA 1214 FMS 15 deg aileron up	14.00	30.03	1.00	20.42	202	-15.00	80.00	50.00	0.00	0.00	0.00
16	NACA 1214 FMS 15 deg flap down	14.00	30.03	1.00	20.42	204	15.00	77.62	50.00	0.00	0.00	0.00

Table 3-9 Summary of all aerofoils created for the FMS 182 XFLR5 model

3.2.2 Aerofoil 2D Analysis

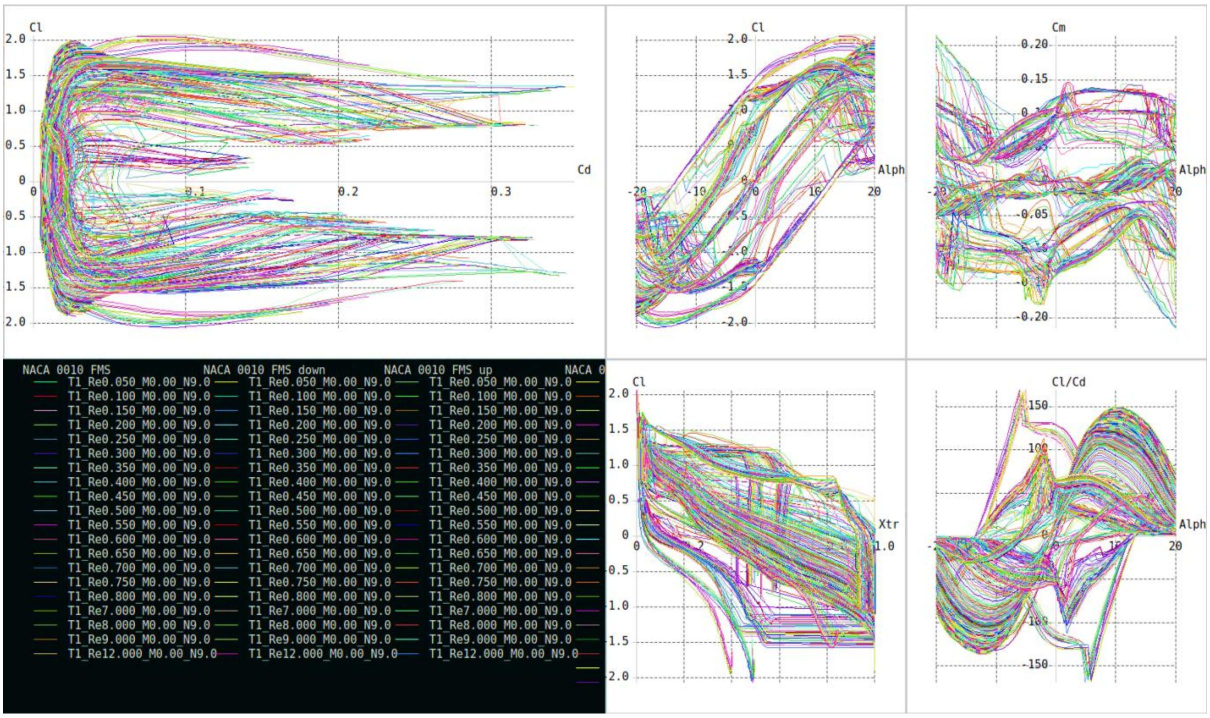
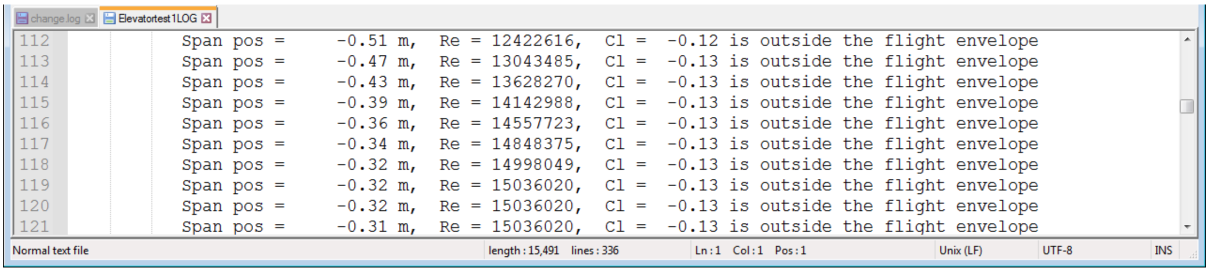


Figure 3.27 An example of the large number of results from aerofoil analysis

To obtain all the necessary aerodynamic data on the identified aerofoils and all of their variants that were created, a 2D analysis of these aerofoils were conducted. The following methods were utilised to understand and interpret the XFLR5 outputs:

- **Aerofoil Polars:** For the range of Reynolds numbers and angle of attack, associated aerodynamic parameters were calculated. Within XFLR5, these sets of parameters for a particular aerofoil are referred to as their polar objects. Once these polar objects are generated and assigned to individual aerofoils, further analysis of 3D sections and entire models benefit from this dataset. Initially a Reynolds number range of 50,000 to 500,000 was used due to expected operational range found in the literature [6]. However, it was experimentally discovered that certain sections of the aerofoils can have a local Reynolds number which is far above/below that range. Therefore, separate sets of polars were generated to patch those regions and ensure higher degree of simulation fidelity.

- **Filling in the gaps in data set:**



```
         Span pos =     -0.51 m,  Re = 12422616,  Cl =  -0.12 is outside the flight envelope
         Span pos =     -0.47 m,  Re = 13043485,  Cl =  -0.13 is outside the flight envelope
         Span pos =     -0.43 m,  Re = 13628270,  Cl =  -0.13 is outside the flight envelope
         Span pos =     -0.39 m,  Re = 14142988,  Cl =  -0.13 is outside the flight envelope
         Span pos =     -0.36 m,  Re = 14557723,  Cl =  -0.13 is outside the flight envelope
         Span pos =     -0.34 m,  Re = 14848375,  Cl =  -0.13 is outside the flight envelope
         Span pos =     -0.32 m,  Re = 14998049,  Cl =  -0.13 is outside the flight envelope
         Span pos =     -0.32 m,  Re = 15036020,  Cl =  -0.13 is outside the flight envelope
         Span pos =     -0.32 m,  Re = 15036020,  Cl =  -0.13 is outside the flight envelope
         Span pos =     -0.31 m,  Re = 15036020,  Cl =  -0.13 is outside the flight envelope
```

Figure 3.28 XFLR5 solver problems for lack of data points

The Figure 3.28 shows a typical problem encountered while attempting to perform viscous/3D analysis on the aerofoil for performance and stability/control simulations. These problem areas had to be checked for every simulation run for every foil/model configuration to ensure simulation integrity and thus appropriate sets of new foil data were generated such that the simulation results have a higher degree of fidelity.

3.2.3 Plane Modelling

With the aerofoils generated and their 2D viscous analysis performed, various 3D models were created in XRLR5 for the following analysis:

- Performance Analysis: This is where various tests across a range of parameters of interest were conducted to obtain estimation of aerodynamic performance of the model variants discussed in the Results section.
- Stability Analysis: This is where static and dynamic tests were performed to obtain useful stability parameters based on model variants discussed in the Results section.
- Control Analysis: This is where dynamic responses to control inputs were simulated to obtain useful control parameters based on model variants discussed in the Results section**.**

3.2.3.1 Modelling Components

For modelling considerations, the aircraft was sectioned off into multiple components that were modelled individually and assembled as complete model variants. The following major components were modelled as 3D surfaces:

- Main Wing
- Horizontal Stabiliser
- Vertical Stabiliser

The following contributors to the aircraft mass, moment and inertia were modelled as point masses:

- Fuselage
- Wing Struts
- Wing Connector Tube
- Propeller + Propeller Mount
- Motor +Motor Mount + Propeller Shaft
- Landing Gears
- Battery
- Servos

The fuselage geometry was not modelled as a 3D surface. Due to XFLR5's analytical methods introducing numerical problems in the solutions when a fuselage is added, its 3D representation was left out of the modelling. However, to improve the fidelity of the dynamic modelling, various other components were all modelled as point masses along with the fuselage based on previous measurements.

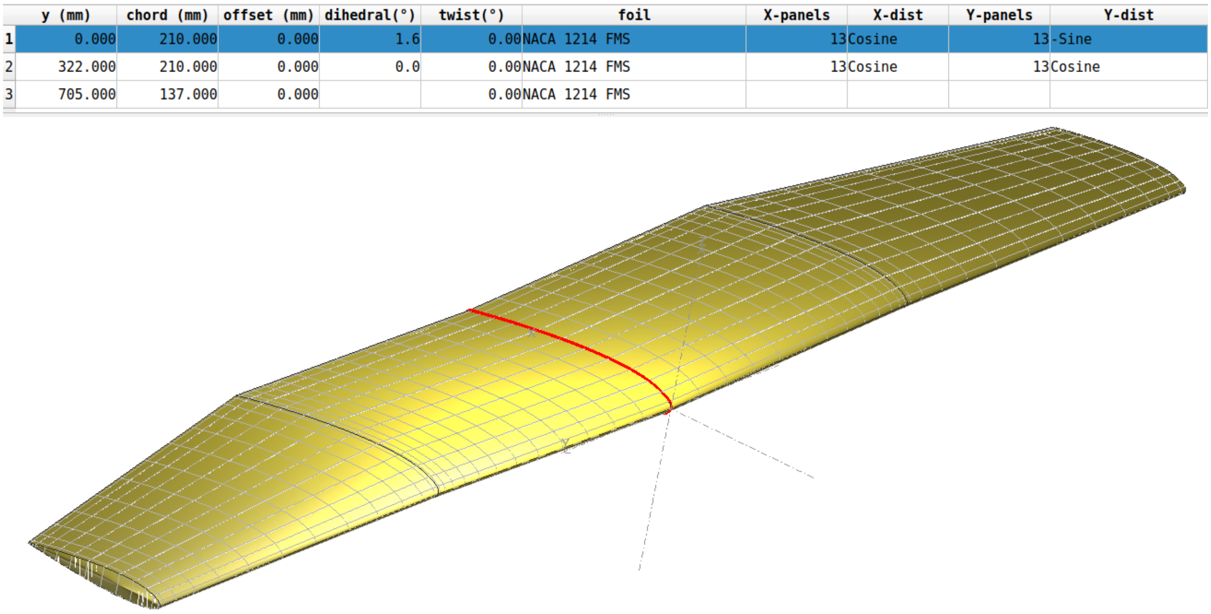
3.2.3.2 Base Model and variants

For all the various performance, stability and control tests conducted, a range of models in different configurations were created.

3.2.3.2.1 Base Model (FMS 182b)

The base model comprises of the main wing, horizontal and vertical stabilisers along with all the point masses.

3.2.3.2.1.1 Main Wing

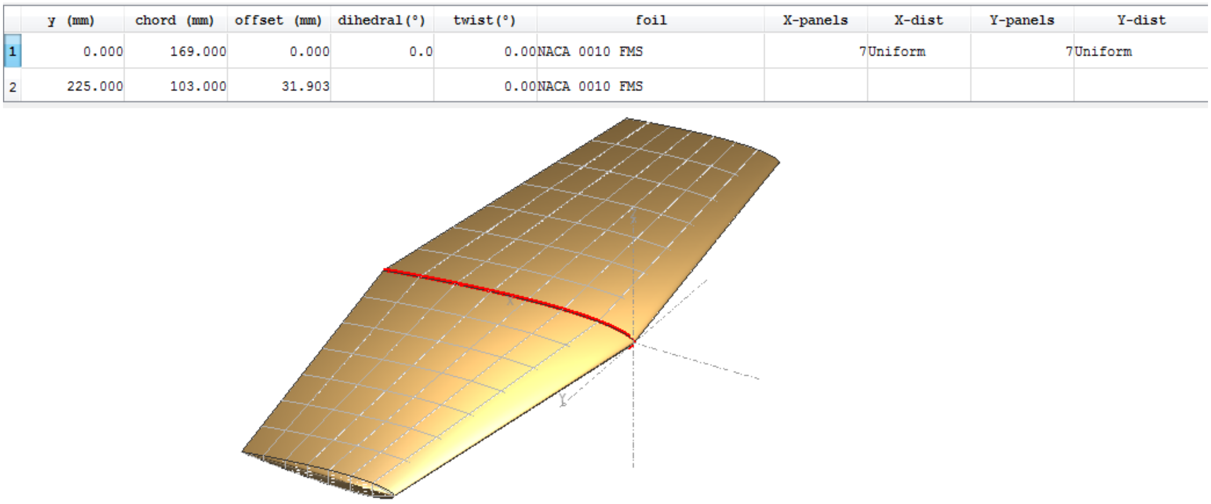


	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	210.000	0.000	1.6	0.00	NACA 1214 FMS	13	Cosine	13	-Sine
2	322.000	210.000	0.000	0.0	0.00	NACA 1214 FMS	13	Cosine	13	Cosine
3	705.000	137.000	0.000		0.00	NACA 1214 FMS				

Figure 3.29 Modelling of the main wing of the FMS 182

The main wing for the base model was modelled with 13 panels in x and y directions. The foil selected was the NACA 1214 standard foil. The panel distribution was done such that the panel concentration is higher in the regions of intersections where the geometry changes.

3.2.3.2.1.2 Horizontal Stabiliser

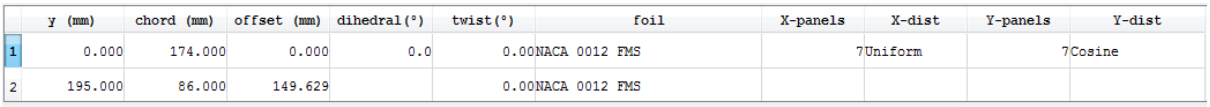


	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	169.000	0.000	0.0	0.00	NACA 0010 FMS	7	Uniform	7	Uniform
2	225.000	103.000	31.903		0.00	NACA 0010 FMS				

Figure 3.30 Modelling of the Horizontal stabiliser of the FMS 182

The generated NACA 0010 foil was assigned to the created horizontal stabiliser section (with 7 x and y panels with uniform distribution).

3.2.3.2.1.3Vertical Stabiliser



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	174.000	0.000	0.0	0.00	NACA 0012 FMS	7	Uniform	7	Cosine
2	195.000	86.000	149.629		0.00	NACA 0012 FMS				

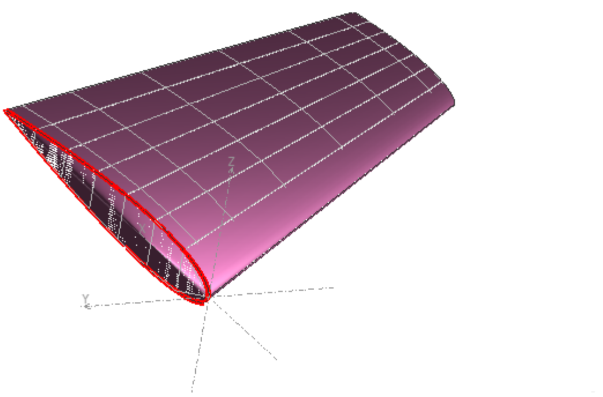


Figure 3.31 Modelling of the Vertical Stabiliser of the FMS 182

The vertical stabiliser was modelled in a similar way except that the NACA 0012 aerofoil was used instead.

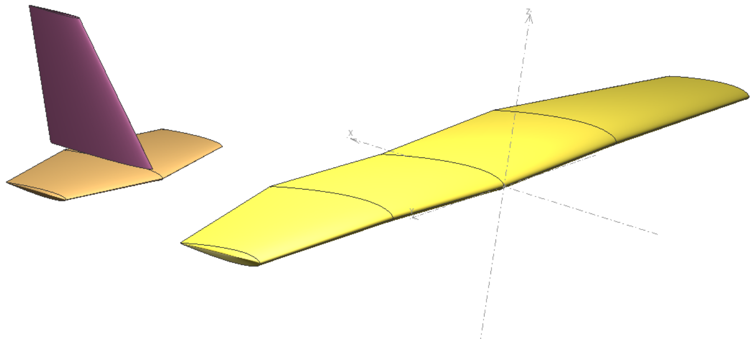
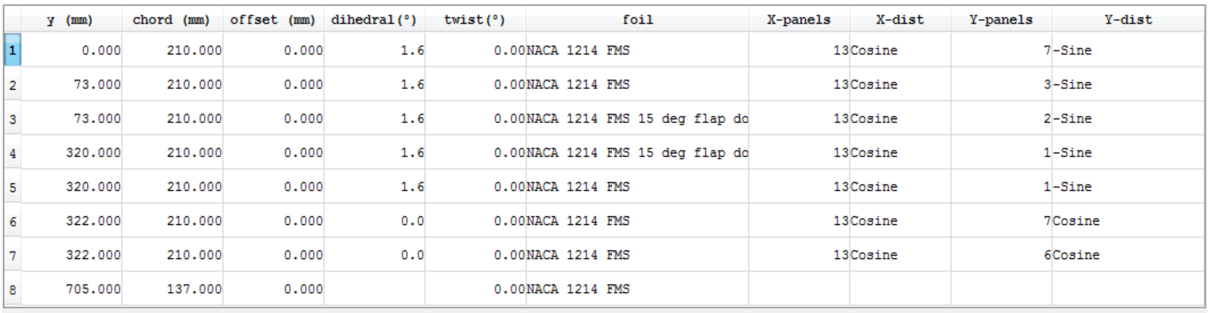


Figure 3.32 Complete base model (FMS 182b)

3.2.3.2.2 FMS 182b Flap

A flapped variant of the base model was modelled to aid in the subsequent analysis. The only section change for the flapped model was the wing.



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	210.000	0.000	1.6	0.00	NACA 1214 FMS	13	Cosine	7	-Sine
2	73.000	210.000	0.000	1.6	0.00	NACA 1214 FMS	13	Cosine	3	-Sine
3	73.000	210.000	0.000	1.6	0.00	NACA 1214 FMS 15 deg flap do	13	Cosine	2	-Sine
4	320.000	210.000	0.000	1.6	0.00	NACA 1214 FMS 15 deg flap do	13	Cosine	1	-Sine
5	320.000	210.000	0.000	1.6	0.00	NACA 1214 FMS	13	Cosine	1	-Sine
6	322.000	210.000	0.000	0.0	0.00	NACA 1214 FMS	13	Cosine	7	Cosine
7	322.000	210.000	0.000	0.0	0.00	NACA 1214 FMS	13	Cosine	6	Cosine
8	705.000	137.000	0.000		0.00	NACA 1214 FMS				

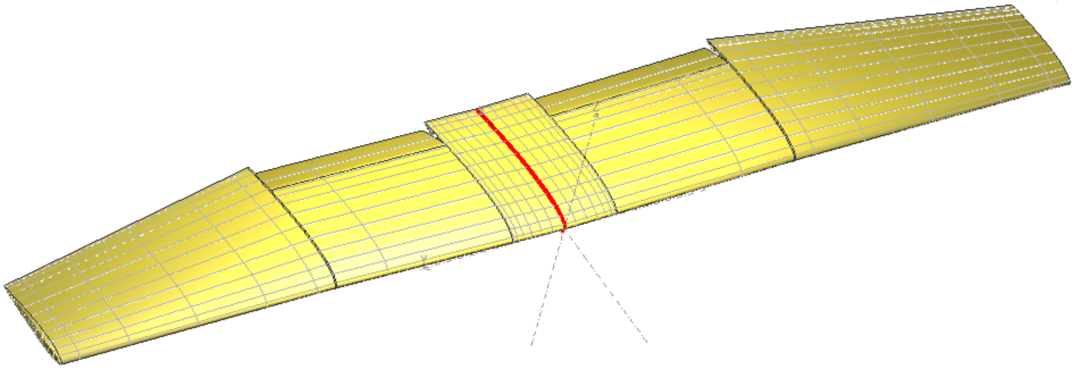


Figure 3.33 Modelling of the flapped version of the FMS 182

Multiple separate sections had to be defined for the flaps to section of the wing in a way where the solvers know the clear boundary between sections and can model the airflow above those regions appropriately. There were some panel density related problems leading to instability and therefore appropriate panels were selected to avoid the crashes. The version of the aerofoil NACA 1214 with the flap modelled was used for the region of the wing where the flaps sit.

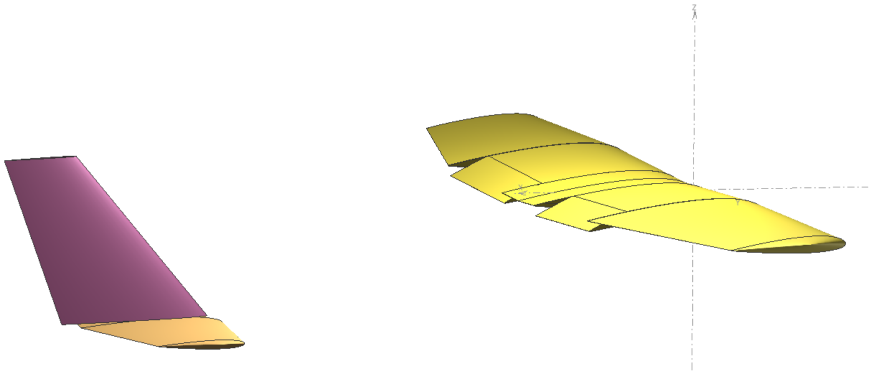
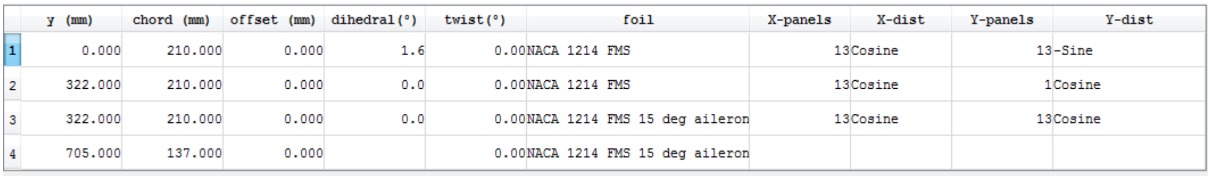


Figure 3.34 Completed model variant of the FMS 182b with full flap deflection.

3.2.3.2.3 FMS 182b Aileron Down/Up

The base model was altered to create two variants with positive and negative aileron deflections.



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	210.000	0.000	1.6	0.00	NACA 1214 FMS	13	Cosine	13	-Sine
2	322.000	210.000	0.000	0.0	0.00	NACA 1214 FMS	13	Cosine	1	Cosine
3	322.000	210.000	0.000	0.0	0.00	NACA 1214 FMS 15 deg aileron	13	Cosine	13	Cosine
4	705.000	137.000	0.000		0.00	NACA 1214 FMS 15 deg aileron				

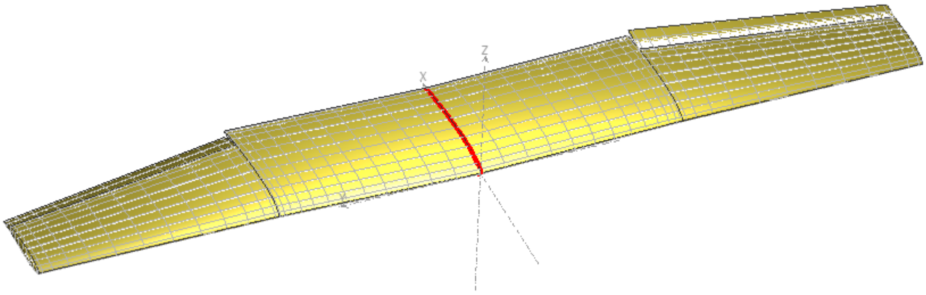
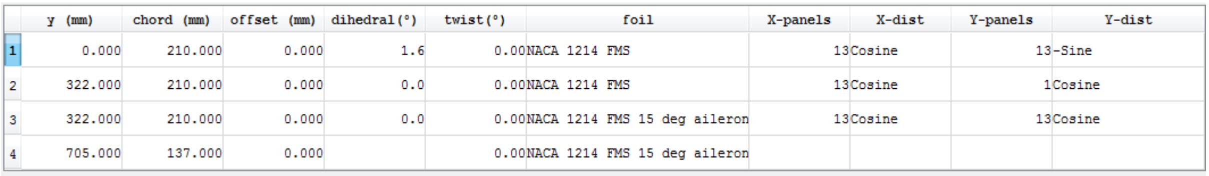


Figure 3.35 Model variant with positive aileron deflection (aileron down)



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	210.000	0.000	1.6	0.00	NACA 1214 FMS	13	Cosine	13	-Sine
2	322.000	210.000	0.000	0.0	0.00	NACA 1214 FMS	13	Cosine	1	Cosine
3	322.000	210.000	0.000	0.0	0.00	NACA 1214 FMS 15 deg aileron	13	Cosine	13	Cosine
4	705.000	137.000	0.000		0.00	NACA 1214 FMS 15 deg aileron				

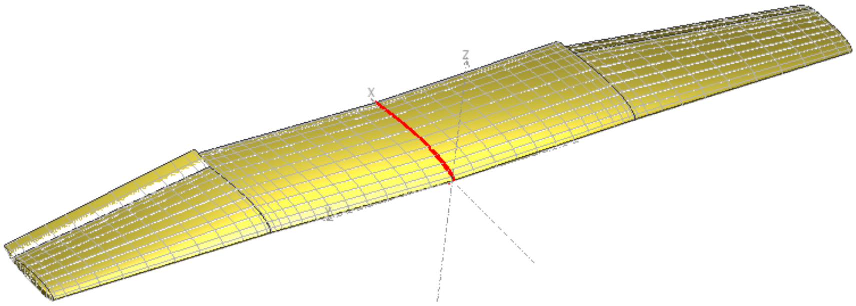


Figure 3.36 Model variant with positive aileron deflection (aileron up)

Both the variants were created in similar ways utilising the NACA 1214 deflected foils previously generated for the ailerons. Separate sections to represent the ailerons were created and these foils were assigned. In case of the flaps, the symmetrical option was valid. However, in case of ailerons, the right and the left had to be assigned aerofoils separately to account for their opposing deflection directions.

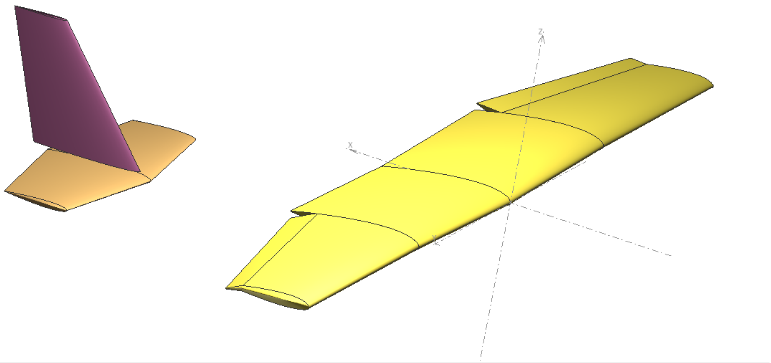


Figure 3.37 FMS 182b Aileron down

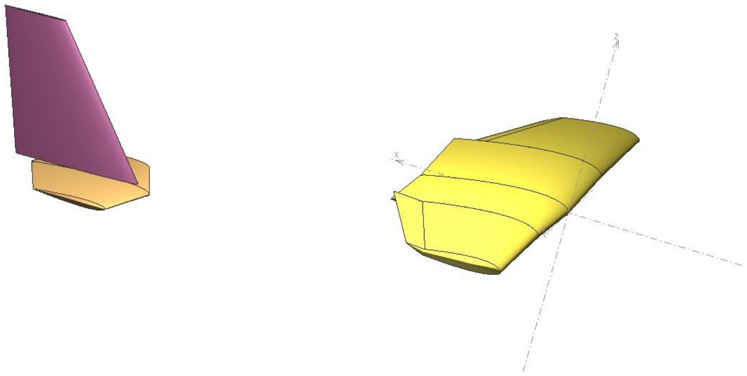
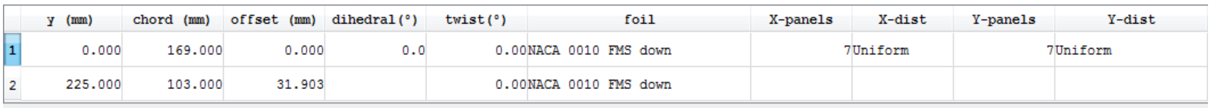


Figure 3.38 FMS 182b Aileron up

3.2.3.2.4 FMS 182b Elevator Down/Up

Two additional model variants were created to represent the base model with positive and negative elevator deflections.



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	169.000	0.000	0.0	0.00	NACA 0010 FMS down	7	Uniform	7	Uniform
2	225.000	103.000	31.903		0.00	NACA 0010 FMS down				

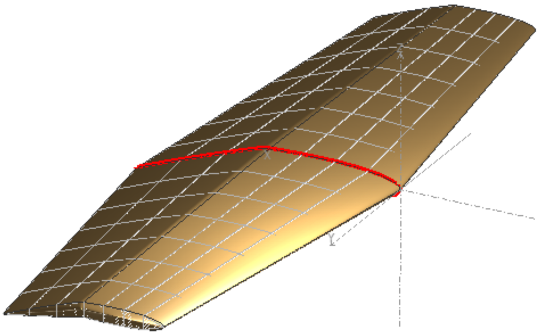
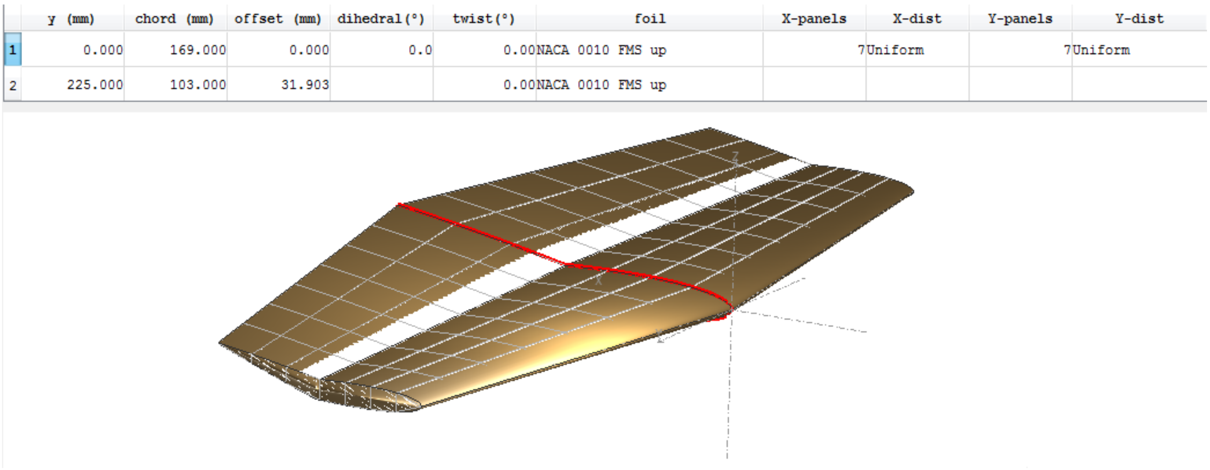


Figure 3.39 Modelling full positive deflection of elevator



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	169.000	0.000	0.0	0.00	NACA 0010 FMS up	7	Uniform	7	Uniform
2	225.000	103.000	31.903		0.00	NACA 0010 FMS up				

Figure 3.40 Modelling full negative deflection of the elevator

The same process as before as followed except for the use of the NACA 0010 foil variants with the deflection modelled.

3.2.3.2.5 FMS 182b Elevator Down/Up full

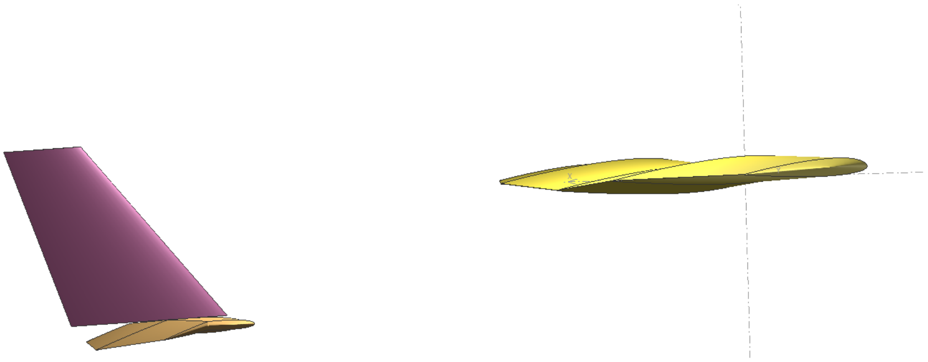


Figure 3.41 Complete FMS 182b Elevator Down model

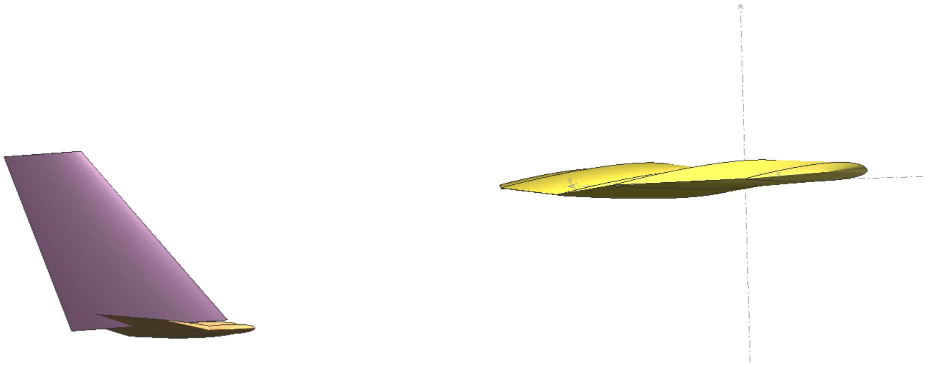
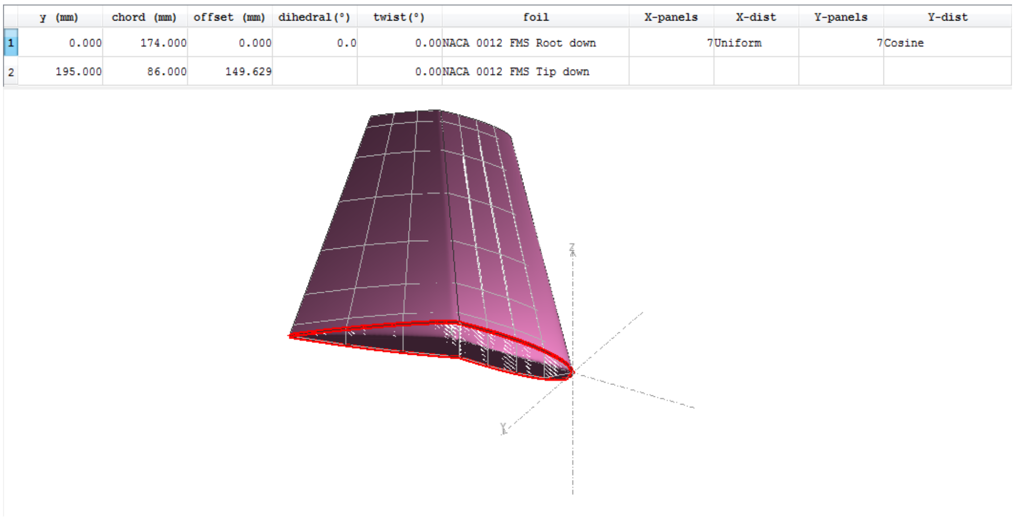


Figure 3.42 Complete FMS 182b Elevator Up model

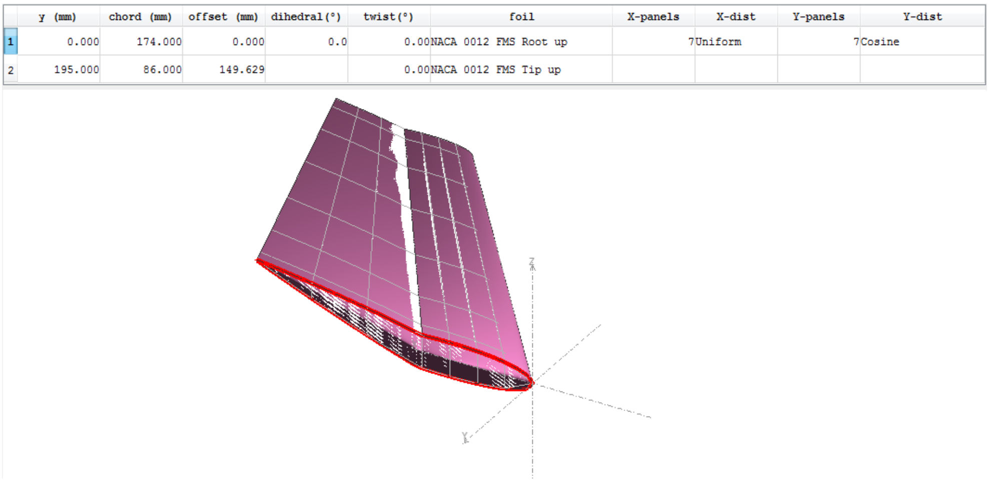
3.2.3.2.6 FMS 182b Rudder Down/Up

Two additional model variants were generated to represent the base model with the rudder deflections.



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	174.000	0.000	0.0	0.00	NACA 0012 FMS Root down	7	Uniform	7	Cosine
2	195.000	86.000	149.629		0.00	NACA 0012 FMS Tip down				

Figure 3.43 Modelling FMS rudder deflection downward



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	174.000	0.000	0.0	0.00	NACA 0012 FMS Root up	7	Uniform	7	Cosine
2	195.000	86.000	149.629		0.00	NACA 0012 FMS Tip up				

Figure 3.44 Modelling FMS rudder deflection upward

The same process as before was utilised except for the use of different aerofoils that were previously generated. An important difference in the modelling of the rudder is its use of different aerofoil variants for the root and the tip. The reason this needed to be modelled this way is due to the hinge positions at the root and the tip having different x-values. As such, separate sets of aerofoils representing the difference in hinge location were used to improve the modelling accuracy.

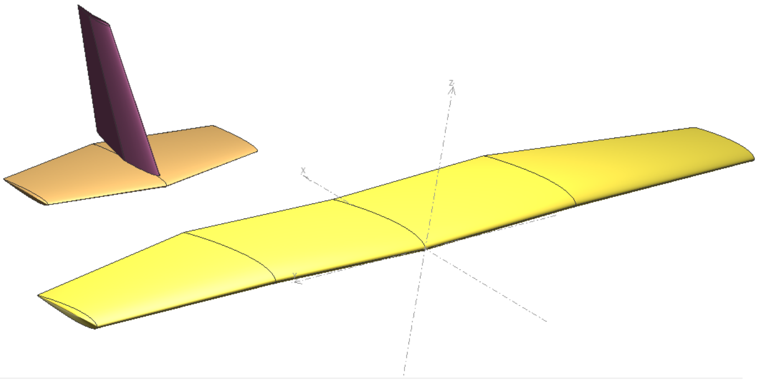


Figure 3.45 FMS 182b Rudder Down model

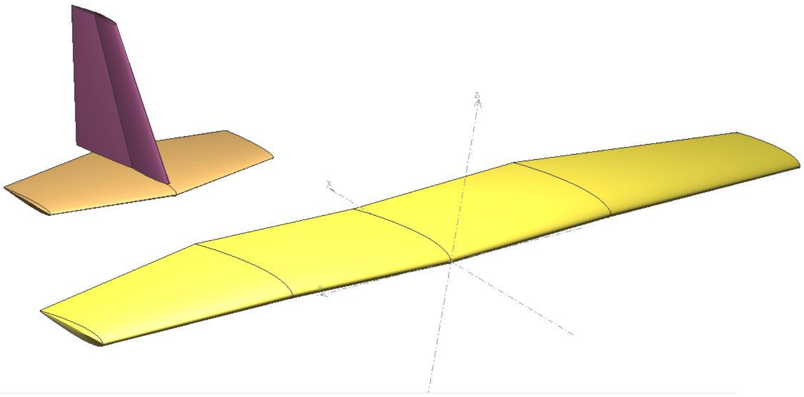
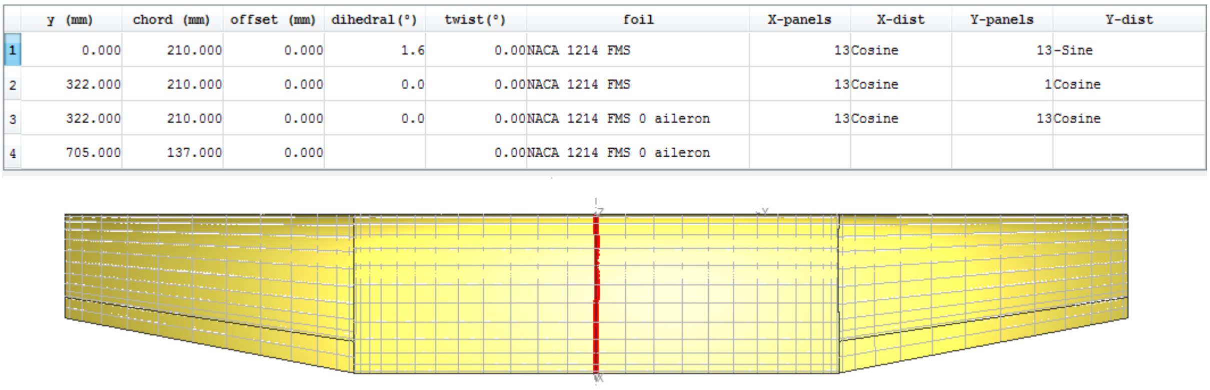


Figure 3.46 FMS 182b Rudder Up model

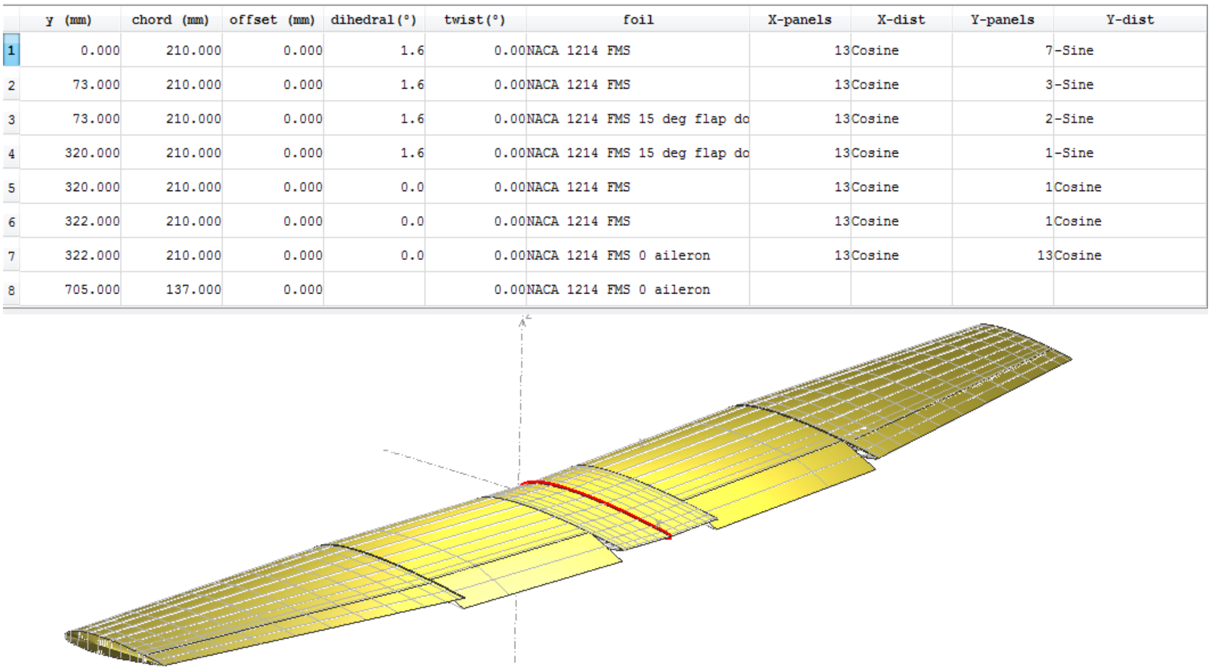
3.2.3.2.7 FMS 182b SC0/SC0 Flap

For the stability and control tests performed, two additional model variants (with and without flaps) needed to be created to account for the way XFLR5 performs these tests.



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	210.000	0.000	1.6	0.00	NACA 1214 FMS	13	Cosine	13	-Sine
2	322.000	210.000	0.000	0.0	0.00	NACA 1214 FMS	13	Cosine	1	Cosine
3	322.000	210.000	0.000	0.0	0.00	NACA 1214 FMS 0 aileron	13	Cosine	13	Cosine
4	705.000	137.000	0.000		0.00	NACA 1214 FMS 0 aileron				

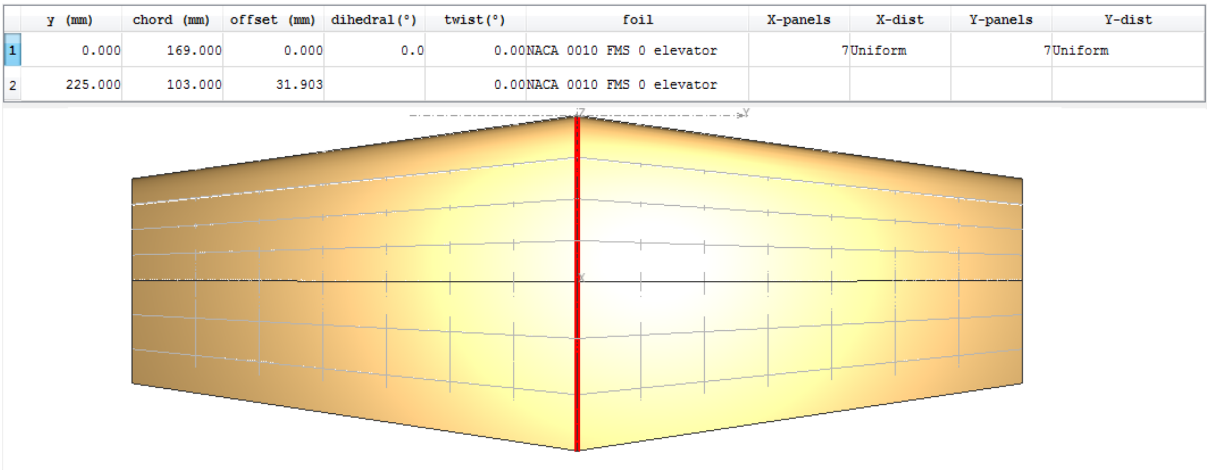
Figure 3.47 FMS 182b SC0 Wing design



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	210.000	0.000	1.6	0.00	NACA 1214 FMS	13	Cosine	7	-Sine
2	73.000	210.000	0.000	1.6	0.00	NACA 1214 FMS	13	Cosine	3	-Sine
3	73.000	210.000	0.000	1.6	0.00	NACA 1214 FMS 15 deg flap do	13	Cosine	2	-Sine
4	320.000	210.000	0.000	1.6	0.00	NACA 1214 FMS 15 deg flap do	13	Cosine	1	-Sine
5	320.000	210.000	0.000	0.0	0.00	NACA 1214 FMS	13	Cosine	1	Cosine
6	322.000	210.000	0.000	0.0	0.00	NACA 1214 FMS	13	Cosine	1	Cosine
7	322.000	210.000	0.000	0.0	0.00	NACA 1214 FMS 0 aileron	13	Cosine	13	Cosine
8	705.000	137.000	0.000		0.00	NACA 1214 FMS 0 aileron				

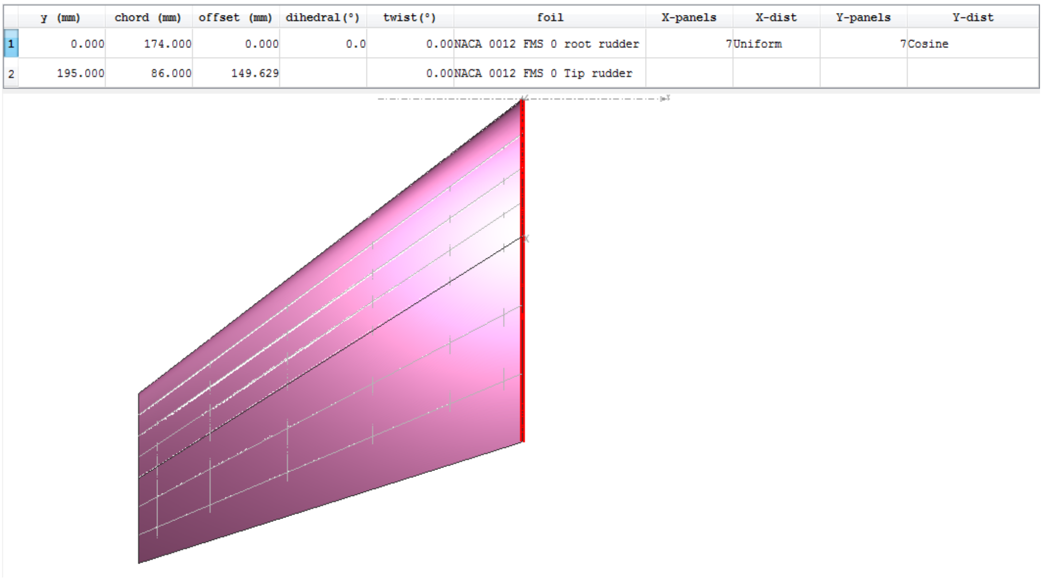
Figure 3.48 FMS 182 SC0 Flap Wing design

In this case, all the control surfaces had to be defined in one single model. However, another separate identical variant for the flapped configuration had to be created. For these two variants, complete new sets of aerofoil had to be generated and configured to have all the hinge positions in the right places but the deflection set to 0.



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	169.000	0.000	0.0	0.00	NACA 0010 FMS 0 elevator	7	Uniform	7	Uniform
2	225.000	103.000	31.903		0.00	NACA 0010 FMS 0 elevator				

Figure 3.49 FMS 182 SC0/SC0 Flap Horizontal Stabiliser model



	y (mm)	chord (mm)	offset (mm)	dihedral(°)	twist(°)	foil	X-panels	X-dist	Y-panels	Y-dist
1	0.000	174.000	0.000	0.0	0.00	NACA 0012 FMS 0 root rudder	7	Uniform	7	Cosine
2	195.000	86.000	149.629		0.00	NACA 0012 FMS 0 Tip rudder				

Figure 3.50 FMS 182 SC0/S0 Flap Vertical Stabiliser model

The horizontal and vertical stabilisers were modelled as before for both variants except with the zero deflection NACA foils.

3.2.3.2.8 FMS 182b SC0/SC0 Flap full

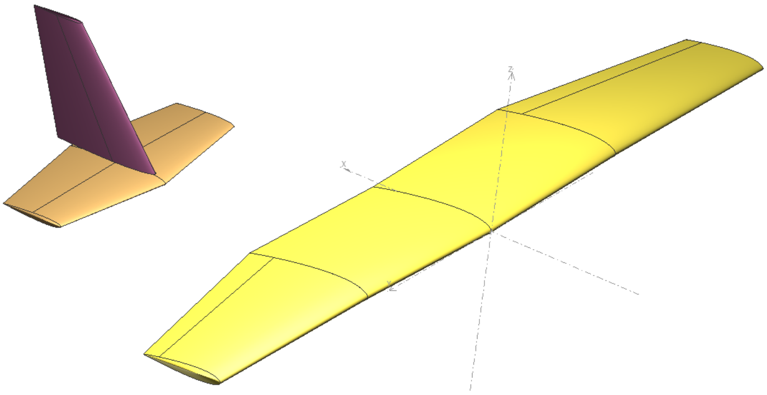


Figure 3.51 Complete FMS 182b SC0 model

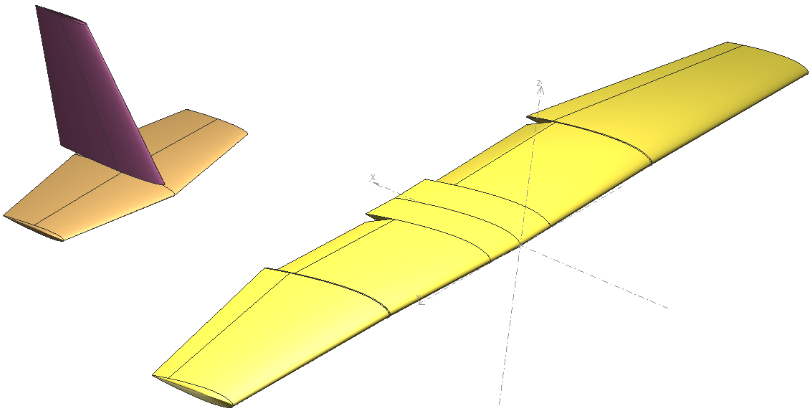


Figure 3.52 Complete FMS 182b SC0 Flap model

3.2.3.3 Inertial Considerations

Accurate modelling of the dynamic responses and behaviour of the FMS 182 required various inertial factors and their inclusion in the model. While it is possible to use the entire aircraft's weight as the sole quantity for weight, significant improvement to the modelling can be achieved through carefully placing individual point masses where they really are located. The individual moments created by all these masses can therefore contribute to a much more accurate analysis. The following images depict the point mass distributions. For the individual labels of the point masses, Figures 3.57-3.58 should be consulted as reading them all from the XFLR5 screenshots without some overlap is impossible to avoid.

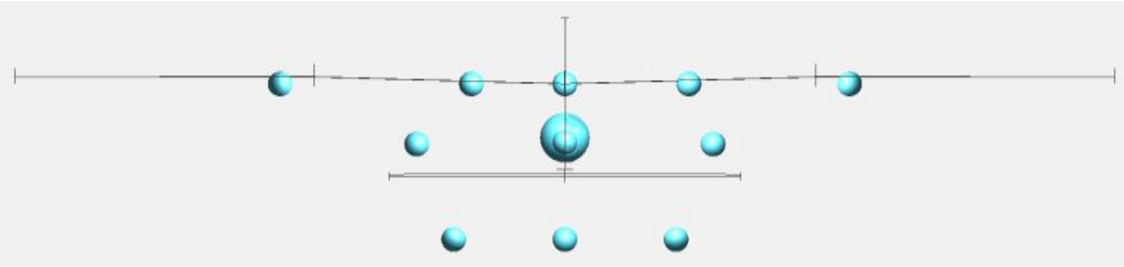


Figure 3.53 Distribution of point masses for inertial modelling of the FMS 182b (front view)

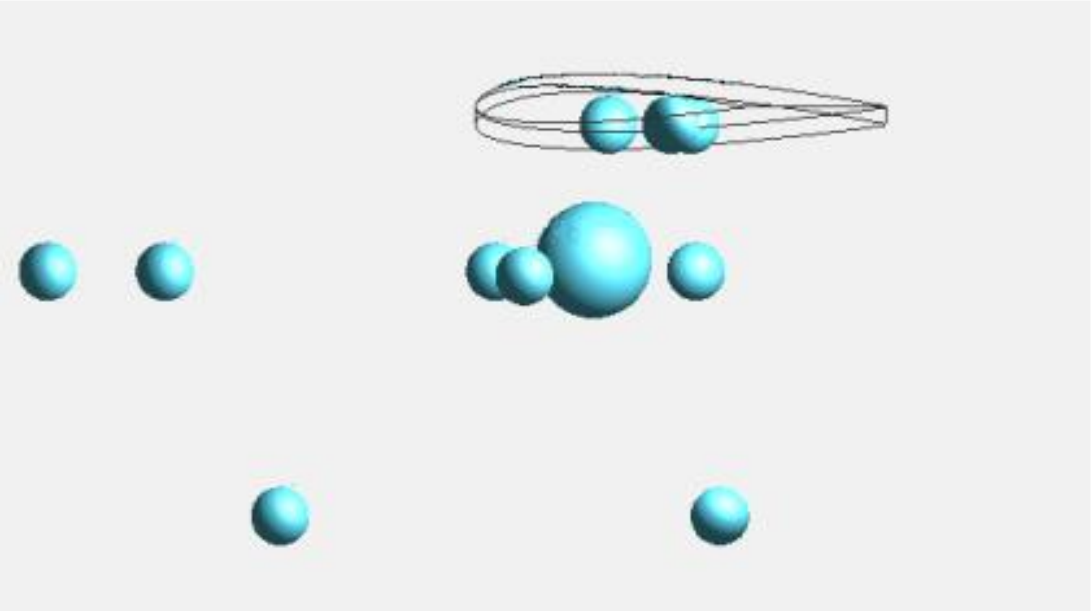


Figure 3.54 FMS 182b Point mass distribution side view (excluding stabiliser)

All the manufacturing and assembling errors found in the measurement phase of the project have been previously documented. These masses were modelled in the simulation based on that data set which already captured problems such as uneven weight distribution or asymmetric load from misplaced servo location or landing hear with different weights. The individual masses and location of all the physical components were used for this. The main wing needed additional point masses for the additional point masses it contains (see Figure 3.57).

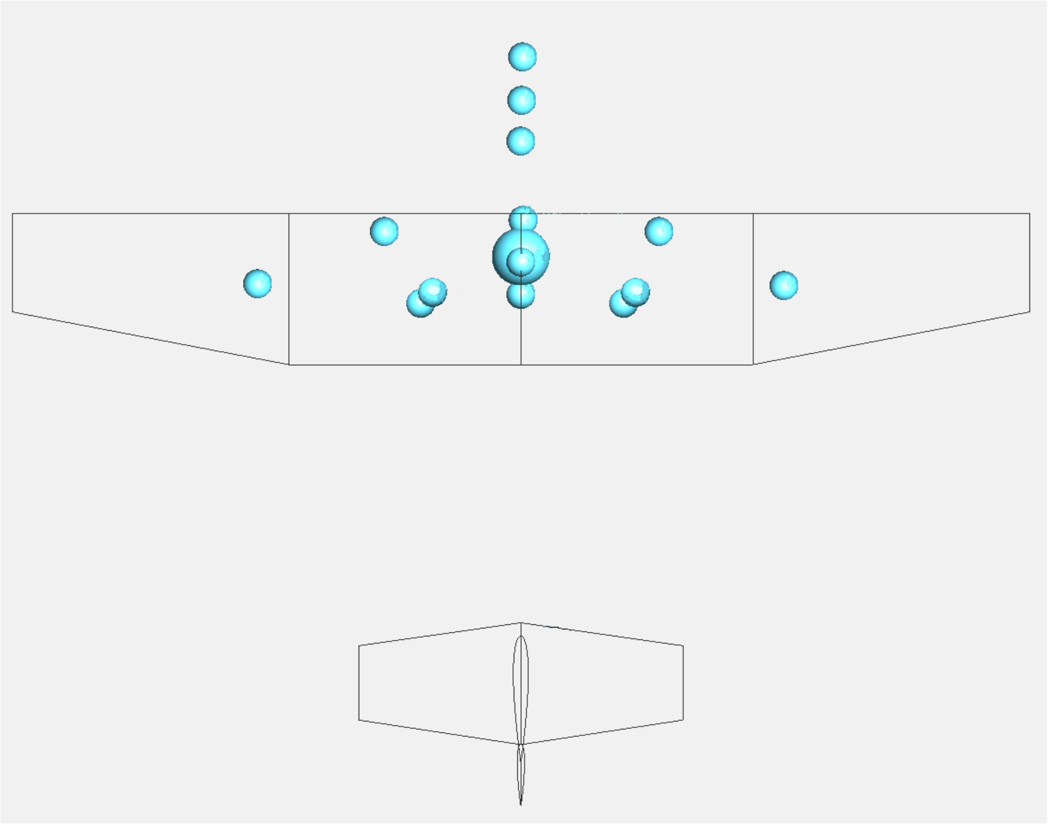


Figure 3.55 FMS 182b point mass distribution top view

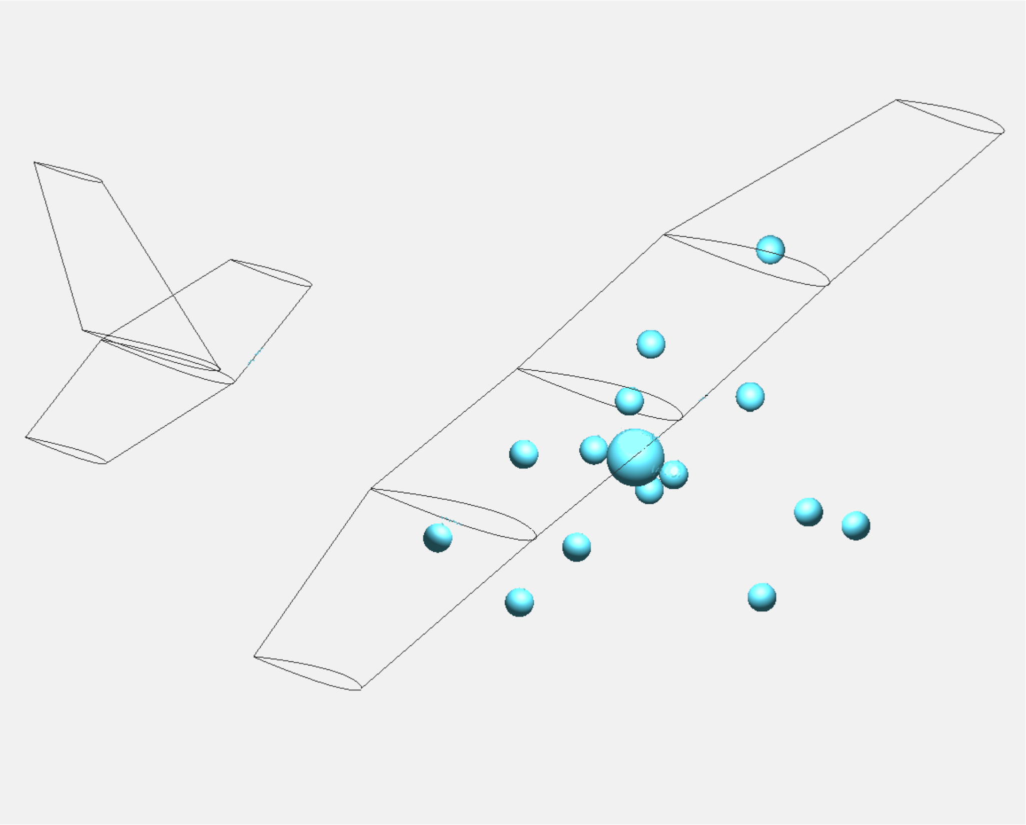
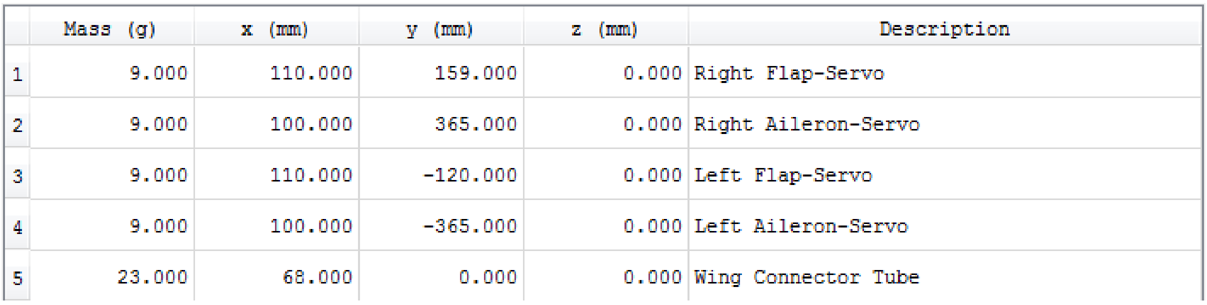
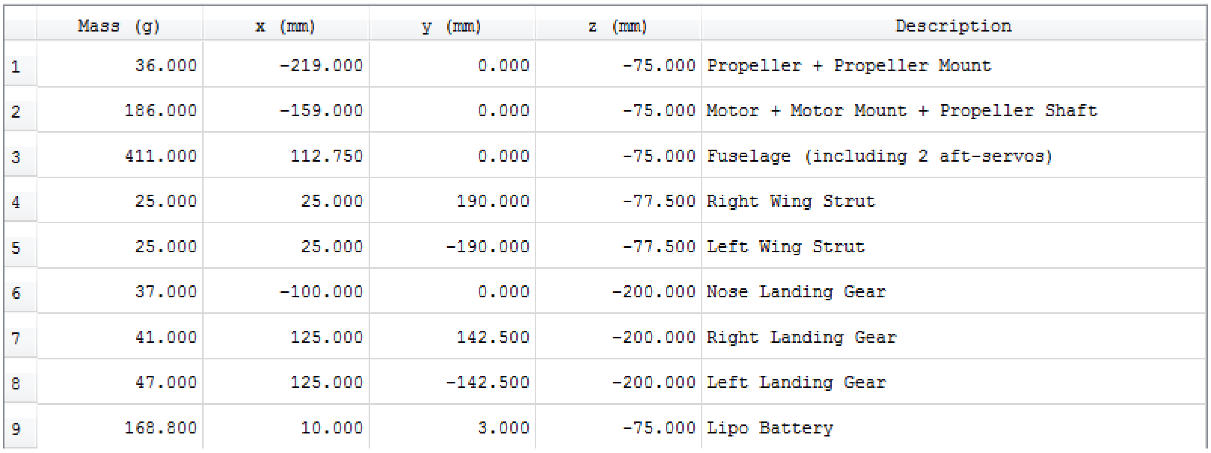


Figure 3.56 FMS 182b full inertial model



	Mass (g)	x (mm)	y (mm)	z (mm)	Description
1	9.000	110.000	159.000	0.000	Right Flap-Servo
2	9.000	100.000	365.000	0.000	Right Aileron-Servo
3	9.000	110.000	-120.000	0.000	Left Flap-Servo
4	9.000	100.000	-365.000	0.000	Left Aileron-Servo
5	23.000	68.000	0.000	0.000	Wing Connector Tube

Figure 3.57 Point mass distribution on wings



	Mass (g)	x (mm)	y (mm)	z (mm)	Description
1	36.000	-219.000	0.000	-75.000	Propeller + Propeller Mount
2	186.000	-159.000	0.000	-75.000	Motor + Motor Mount + Propeller Shaft
3	411.000	112.750	0.000	-75.000	Fuselage (including 2 aft-servos)
4	25.000	25.000	190.000	-77.500	Right Wing Strut
5	25.000	25.000	-190.000	-77.500	Left Wing Strut
6	37.000	-100.000	0.000	-200.000	Nose Landing Gear
7	41.000	125.000	142.500	-200.000	Right Landing Gear
8	47.000	125.000	-142.500	-200.000	Left Landing Gear
9	168.800	10.000	3.000	-75.000	Lipo Battery

Figure 3.58 Point mass distribution within the fuselage

3.2.4 Performance Analysis

In order to obtain the results shown later in the thesis, the following 8 variants of the model were used.

- FMS 182b
- FMS 182b Flaps Down
- FMS 182b Aileron Down
- FMS 182b Aileron Up
- FMS 182b Elevator Down
- FMS 182b Elevator Up
- FMS 182b Rudder Down
- FMS 182b Rudder Up

Types of tests performed:

- Type 1: A fixed speed was assigned with a free stream velocity. The sideslip angle is set to 0. The Analysis method was set to Ring Vortex (VLM2). The polars from aerofoil analysis was available so viscous analysis was setup based on interpolation of that data. The inertial properties of the models were used.

Instead of the wing planform area, the projected wing planform area on the x-y plane was used. The density and speed of sound was set to sea level. This test was later utilised to estimate pitch stability derivatives due to the limitation of the program.

- Type 2: This is the test where fixed lift was used instead of fixed speed. Most performance and stability analysis relied on this. The setup is similar to that of Type 1. However, in this case, the program solves for flight conditions for every angle of attack where the resulting lift matches the weight of the aircraft.

3.2.4.1 Performance Analysis Visualisation

3D Visualisations of various important aerodynamic parameters and phenomenon across the defined range of alpha (-20 to +20) were generated within XFLR5 based on the previously mentioned 2D analysis and performance tests. It was discovered during the test that while XFLR5 is capable of conducting full 3D panel tests for specific geometric sections [3], the software fails to generate accurate predictions when these sections are all brought together for an overall output. As such, to reduce error and allow for the simulation of interactions between sections, the 3D panelling is converted to 2D for this analysis.

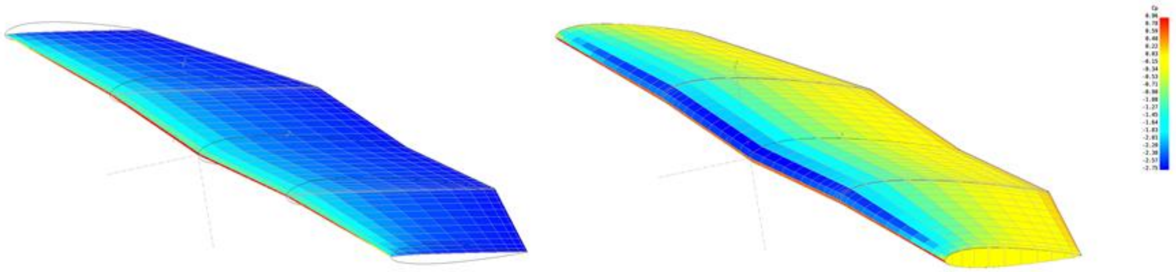


Figure 3.59 2D panelling (left) and 3D panelling (right)

Nevertheless, the modelled sections were put through both 2D and 3D analysis to obtain a sense for the magnitude of difference caused by this change. The following graphs summarize the test results.

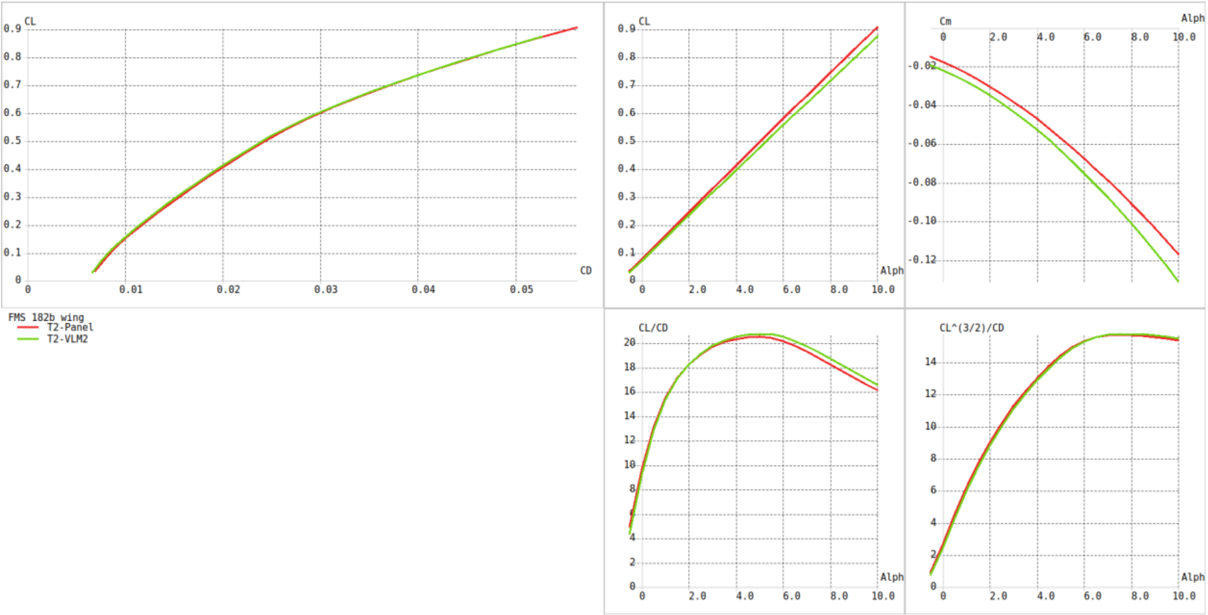


Figure 3.60 Performance results comparison between 2D panels (green) and 3D panel (red)

As it can be seen from the above summary, the difference in performance prediction for virtually all relevant parameters of interest is low enough for this to not render the subsequent analysis unreliable.

3.2.5 Prerequisites for Stability Analysis

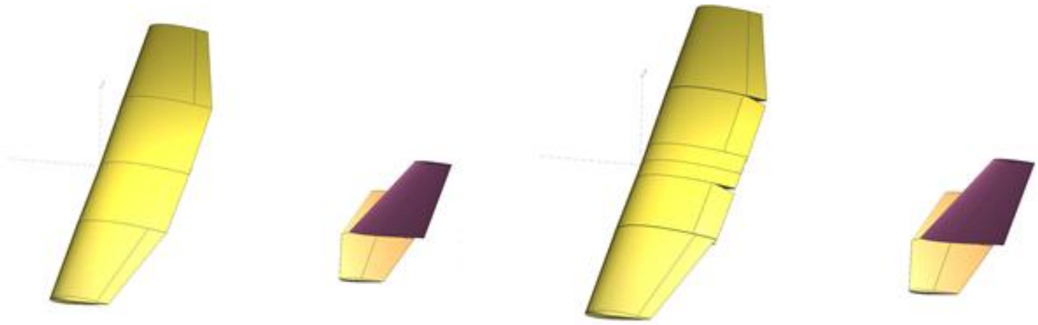


Figure 3.61 FMS 182b SC0 (left) and FMS 182b SC0 Flap (right)

The two additional variants were tested for additional performance data based on weight and COG variations:

1. FMS 182b SC0 : This variants is the combined version of all models except for the flaps. The other difference is that this model has the control surfaces defined and aerofoils modelled but without any deflection angle set.
2. FMS 128b SC0 flapped: This variant is identical to the previous one except that it has the flaps fully deflected.

These are the same model variants that are used for various stability and control tests along with the other models.

3.2.5.1 Stability and Control Tests:

The following are the stability test results for the Stability test base models (with and without flaps). The modes tested are illustrated below. The numbering system inside the images represent arbitrary time with where low to high is in the same direction as the forward direction of time. The animations cannot be captured well in paper however the following images are provided for reference. In the following images showing the observed dynamic modes of the FMS 182, the numbers are chronological arbitrary time samples.

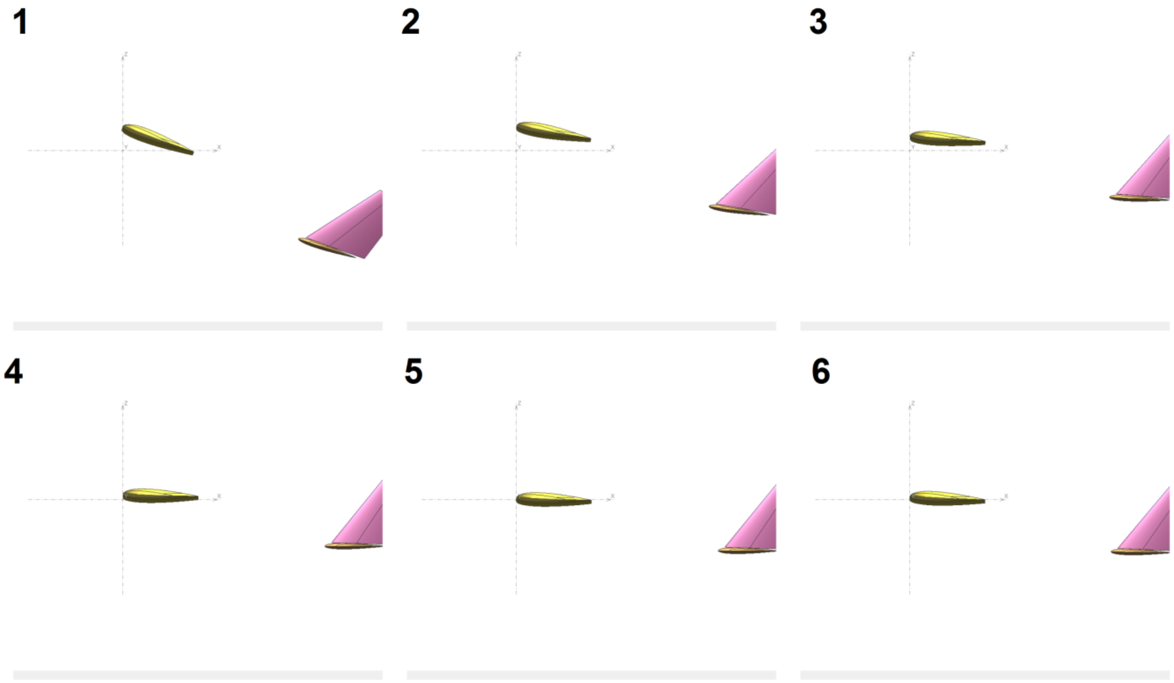


Figure 3.62 Short period pitching mode

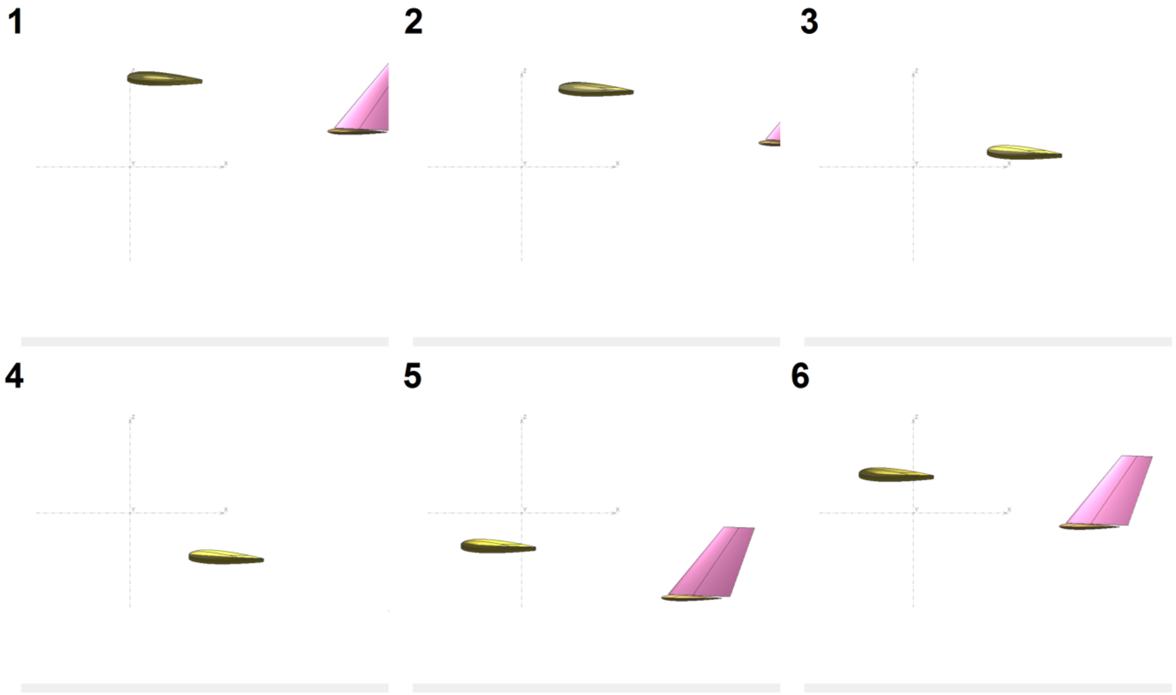


Figure 3.63 Phugoid mode

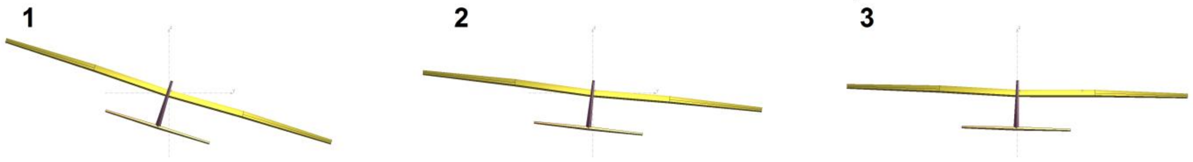


Figure 3.64 Roll damping

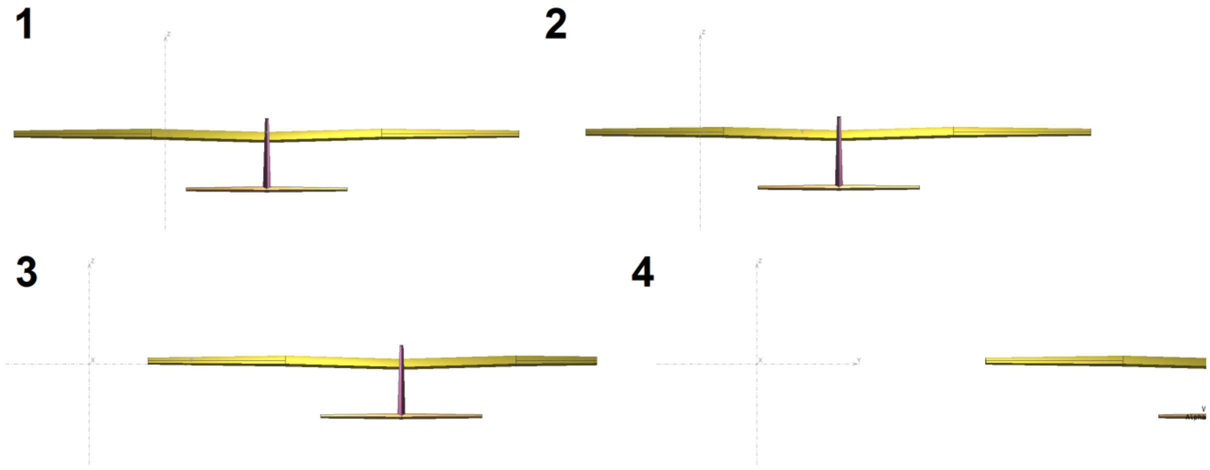


Figure 3.65 Spiral mode

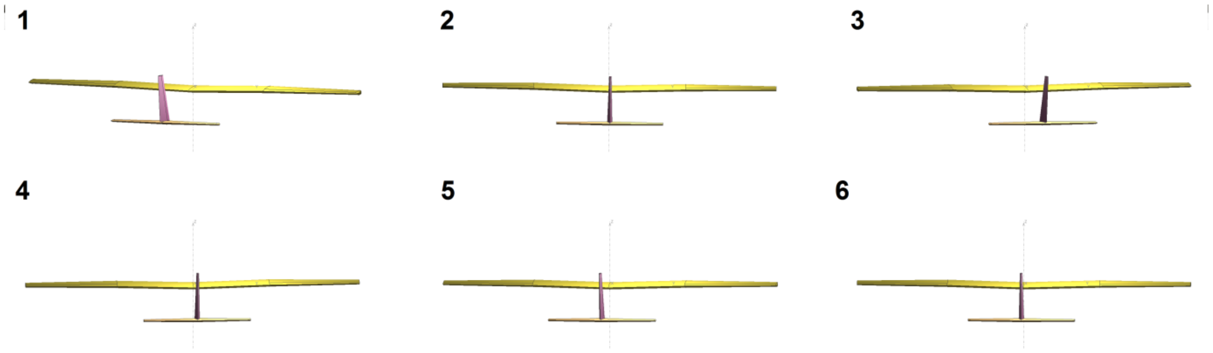


Figure 3.66 Dutch Roll

A range of stability tests were conducted on the models to check for both static and dynamic stability in both longitudinal and lateral planes. To obtain the results presented in the results section Chapter 4.1 the following tests were conducted using the T7 test in XFLR5 (Mix 3D panels/VLM2):

- Standard stability test where the aircraft's responses were measured for a small disturbance.
- Stability test for an incremental range of masses.
- Stability test for an incremental range of CoG/CG positions.
- Control tests for a range of control surface deflections for the ailerons, elevator and rudder.
- The previous test on both flapped and non-flapped model variants.

3.2.5.1.1 Considerations for the tests:

Mass range: The range chosen for all mass altering experiments were started from the base model mass of 1371.3 grams and 50 grams were added to it at the CoG/CG location until a maximum mass of 1921.8 grams. The same masses were used for both the model variants used.

Centre of Gravity range: For all experiments involving a shift in the Centre of Gravity, different ranges were picked for the flapped and non-flapped model variants. The following tables provide a guide to these ranges.

Static Margin (%)	X_CG
0	105.599
5	95.9401
10	86.2812
15	76.6223
20	66.9634
23.42	60.364
25	57.3045
30	47.6456
35	37.9867
40	28.3278
45	18.6689
50	9.01

Table 3-10 Static margin and C.G. ranges for FMS 182b SC0

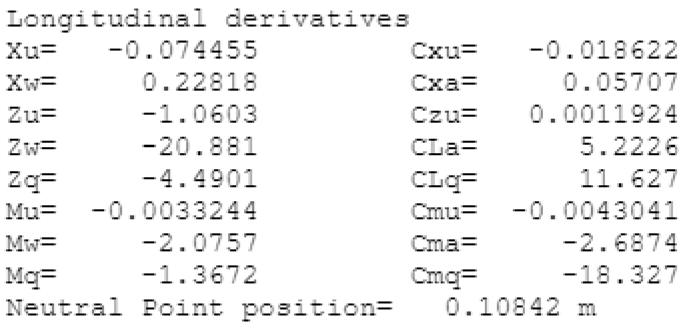
Static Margin (%)	X_CG
0	112.381
5	102.7221
10	93.0632
15	83.4043
20	73.7454
25	64.0865
26.93	60.364
30	54.4276
35	44.7687
40	35.1098
45	25.4509
50	15.792
55	6.1311

Table 3-11 Static margin and C.G. ranges for FMS 182b SC0 Flap

Control Surface Deflections: Since all the control surfaces of the FMS 182 showed a maximum deflection of 15 degrees, all control tests were done with appropriate gains to reflect a positive control surface deflection of 15 degrees and a negative deflection of 15 degrees. The only exception is the flap, which only has the +15 degrees. As a result, for the control surfaces (not the flap), a total of 31 control points were tested for with 1 degree increments.

Post Processing of test results:

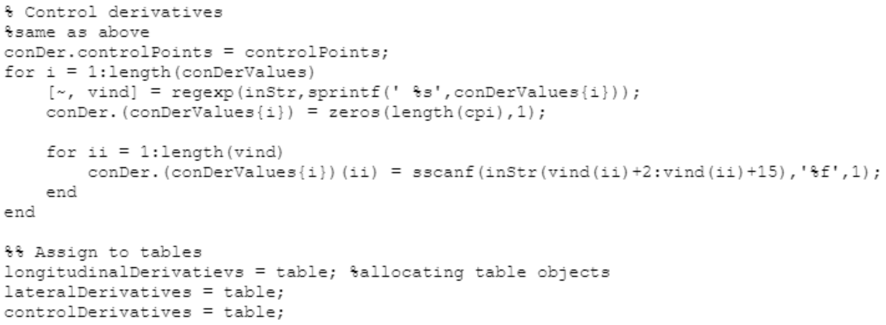
Once the modelling and tests parameters are all set and the tests are run, XFLR5 dumps certain outputs into its log file. For the post processing, generally the log files of XFLR5 are checked and relevant outputs are obtained from them.



```
Longitudinal derivatives
Xu=    -0.074455          Cxu=   -0.018622
Xw=      0.22818          Cxa=     0.05707
Zu=      -1.0603          Czu=   0.0011924
Zw=      -20.881          CLa=      5.2226
Zq=      -4.4901          CLq=      11.627
Mu=   -0.0033244          Cmu=  -0.0043041
Mw=      -2.0757          Cma=     -2.6874
Mq=      -1.3672          Cmq=     -18.327
Neutral Point position=   0.10842 m
```

Figure 3.67 Sample test results from stability test conducted in XFLR5

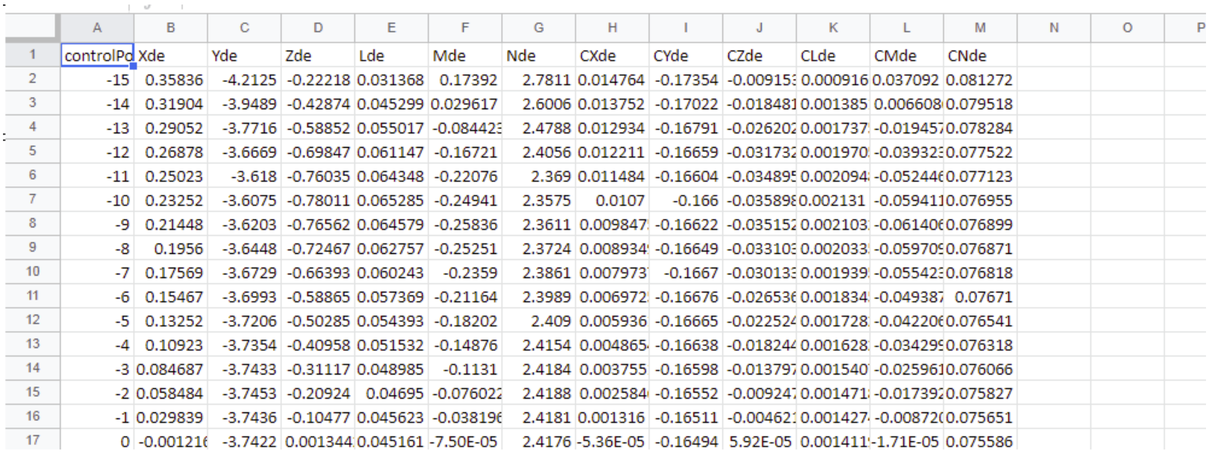
These logs are available in text format. The amount of data generated by these tests results in log files sizes that simply cannot be processed manually. For every control point of every model variant, the data has to be manually obtained from the logs.



```
% Control derivatives
%same as above
conDer.controlPoints = controlPoints;
for i = 1:length(conDerValues)
    [~, vind] = regexp(inStr,sprintf(' %s',conDerValues{i}));
    conDer.(conDerValues{i}) = zeros(length(cpi),1);

    for ii = 1:length(vind)
        conDer.(conDerValues{i})(ii) = sscanf(inStr(vind(ii)+2:vind(ii)+15),'%f',1);
    end
end

%% Assign to tables
longitudinalDerivatievs = table; %allocating table objects
lateralDerivatives = table;
controlDerivatives = table;
```

Figure 3.68 MATLAB code excerpt for the import function for handling XFLR5 data

Since the log files are available in text format, MATLAB code (see Appendix) was written to aid in importing all these logs into MATLAB. The MATLAB code imports the log files into MATLAB and performs a range of operations to look for all the defined parameters within the log files and arrange them into tables.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	controlPo	Xde	Yde	Zde	Lde	Mde	Nde	CXde	CYde	CZde	CLde	CMde	CNde			
2	-15	0.35836	-4.2125	-0.22218	0.031368	0.17392	2.7811	0.014764	-0.17354	-0.00915	0.000916	0.037092	0.081272			
3	-14	0.31904	-3.9489	-0.42874	0.045299	0.029617	2.6006	0.013752	-0.17022	-0.018481	0.001385	0.006608	0.079518			
4	-13	0.29052	-3.7716	-0.58852	0.055017	-0.08442	2.4788	0.012934	-0.16791	-0.02620	0.001737	-0.01945	0.078284			
5	-12	0.26878	-3.6669	-0.69847	0.061147	-0.16721	2.4056	0.012211	-0.16659	-0.03173	0.001970	-0.03932	0.077522			
6	-11	0.25023	-3.618	-0.76035	0.064348	-0.22076	2.369	0.011484	-0.16604	-0.03489	0.002094	-0.05244	0.077123			
7	-10	0.23252	-3.6075	-0.78011	0.065285	-0.24941	2.3575	0.0107	-0.166	-0.03589	0.002131	-0.05941	0.076955			
8	-9	0.21448	-3.6203	-0.76562	0.064579	-0.25836	2.3611	0.009847	-0.16622	-0.03515	0.002103	-0.06140	0.076899			
9	-8	0.1956	-3.6448	-0.72467	0.062757	-0.25251	2.3724	0.008934	-0.16649	-0.03310	0.002033	-0.05970	0.076871			
10	-7	0.17569	-3.6729	-0.66393	0.060243	-0.2359	2.3861	0.007973	-0.1667	-0.03013	0.001939	-0.05542	0.076818			
11	-6	0.15467	-3.6993	-0.58865	0.057369	-0.21164	2.3989	0.006972	-0.16676	-0.02653	0.001834	-0.04938	0.07671			
12	-5	0.13252	-3.7206	-0.50285	0.054393	-0.18202	2.409	0.005936	-0.16665	-0.02252	0.001728	-0.04220	0.076541			
13	-4	0.10923	-3.7354	-0.40958	0.051532	-0.14876	2.4154	0.004865	-0.16638	-0.01824	0.001628	-0.03429	0.076318			
14	-3	0.084687	-3.7433	-0.31117	0.048985	-0.1131	2.4184	0.003755	-0.16598	-0.01379	0.001540	-0.02596	0.076066			
15	-2	0.058484	-3.7453	-0.20924	0.04695	-0.07602	2.4188	0.002584	-0.16552	-0.00924	0.001471	-0.01739	0.075827			
16	-1	0.029839	-3.7436	-0.10477	0.045623	-0.03819	2.4181	0.001316	-0.16511	-0.00462	0.001427	-0.00872	0.075651			
17	0	-0.00121	-3.7422	0.001344	0.045161	-7.50E-05	2.4176	-5.36E-05	-0.16494	5.92E-05	0.001411	-1.71E-05	0.075586			

Figure 3.69 Sample processed data table from Stability/Control Test

The MATLAB code generates the CSV files which provide the data tables for a given model in a given configuration in a given test (under a given range of control points). In the results sections, many of the data tables are processed versions of these data tables generated by MATLAB.

3.3 X-Plane Modelling Method

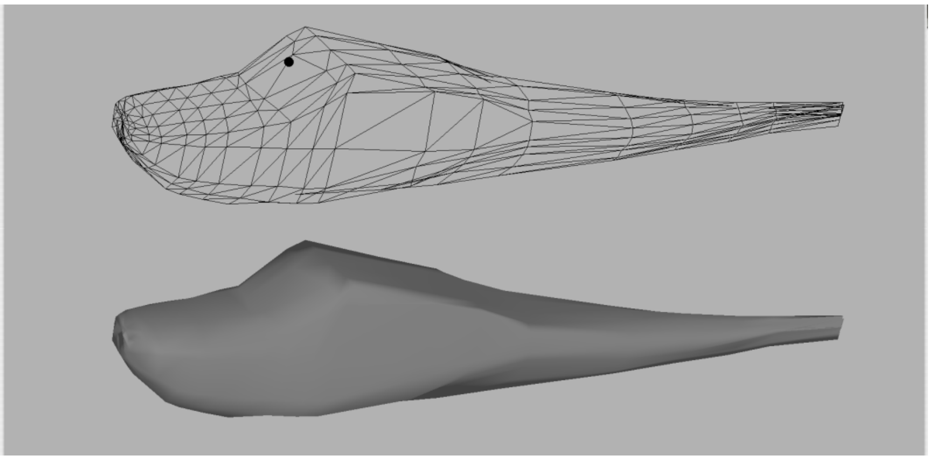


Figure 3.70 FMS 182 fuselage modelling in X-Plane

The modelling process in X-Plane (X-Plane 10) starts with prior knowledge of the target airframe that is to be modelled. In this research, the important physical and geometric characteristics of the FMS SkyTrainer 182 model plane [1] were recorded. Although, it should be noted here that certain modifications to the default airframe was made. However, that does not alter the methods used to construct an accurate X-Plane model of the airframe. The key parameters relating to the airframe geometry and mass has been obtained both through the manufacturer's product specification sheet and physical measurements. That data has been used to construct an X-Plane model of the plane. The software XFLR5 [3] was also greatly utilised in improving the aerodynamic modelling within X-Plane [4] (discussed in detail later).

The following components were modelled using Plane Maker:

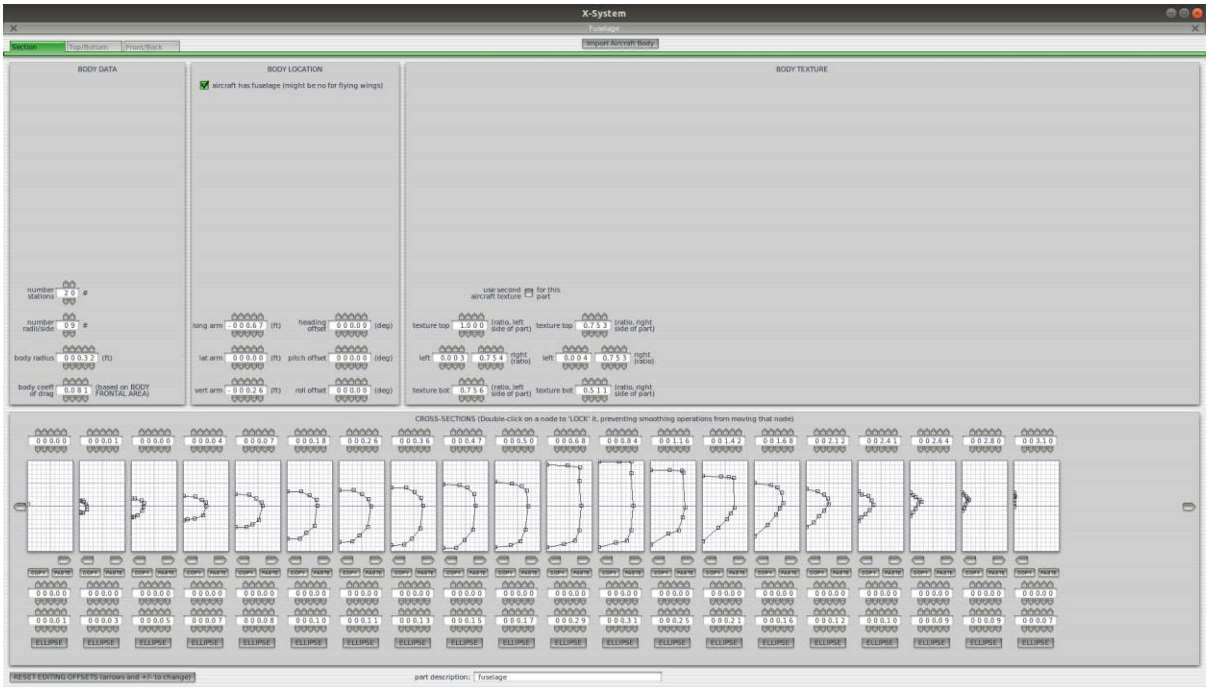
- Fuselage
- Wings
- Horizontal Stabiliser
- Vertical Stabiliser
- Landing Gears

- Wing Struts
- Propeller
- Propeller Spinner

3.3.1 Fuselage modelling process

The construction process is initiated using the Plane Maker software. The mesh design for the fuselage is handled by the following three subsections:

- Section
- Top/Bottom
- Front/Back


X-System
Fuselage
Section
Top/Bottom
Front/Back
Import Aircraft Body
BODY DATA
BODY LOCATION
BODY TEXTURE
CROSS-SECTIONS (Double-click on a node to 'LOCK' it, preventing smoothing operations from moving that node)
COPY
PASTE
ELLIPSE
RESET EDITING OFFSETS (arrows and +/- to change)

BODY DATA
number stations 20 #
number radii/side 09 #
body radius 000.32 (ft)
body coeff of drag 0.081 (based on BODY FRONTAL AREA)

BODY LOCATION
☑ aircraft has fuselage (might be no for flying wings)
long arm -000.67 (ft)
lat arm 000.00 (ft)
vert arm -000.26 (ft)
heading offset 000.00 (deg)
pitch offset 000.00 (deg)
roll offset 000.00 (deg)

BODY TEXTURE
use second aircraft texture □ for this part
texture top 1.000 (ratio, left side of part)
left 0.003 0.754 right (ratio)
texture bot 0.756 (ratio, left side of part)
texture top 0.753 (ratio, right side of part)
left 0.004 0.753 right (ratio)
texture bot 0.511 (ratio, right side of part)

CROSS-SECTIONS

000.00	000.01	000.00	000.04	000.07	000.18	000.26	000.36	000.47	000.50	000.68	000.84	001.16	001.42	001.68	002.12	002.41	002.64	002.80	003.10
000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00	000.00
000.01	000.03	000.05	000.07	000.08	000.10	000.11	000.13	000.15	000.17	000.29	000.31	000.25	000.21	000.16	000.12	000.10	000.09	000.09	000.07

part description: fuselage

Figure 3.71 Fuselage modelling settings for FMS 182 in Plane Maker

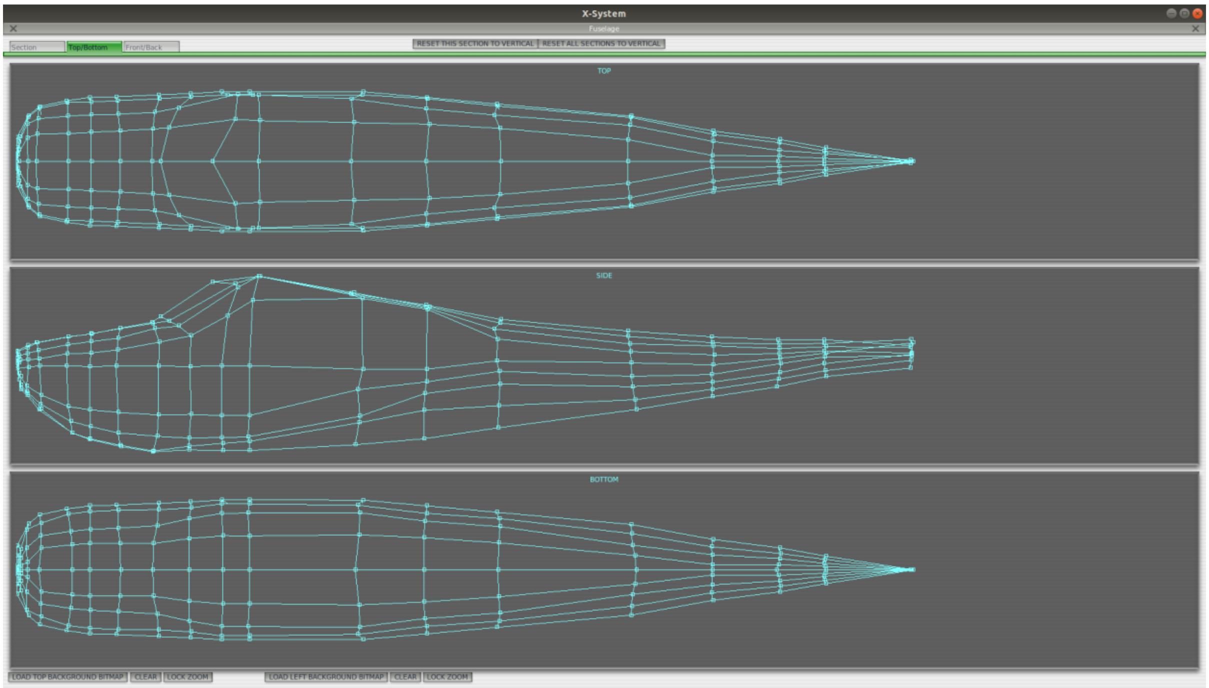
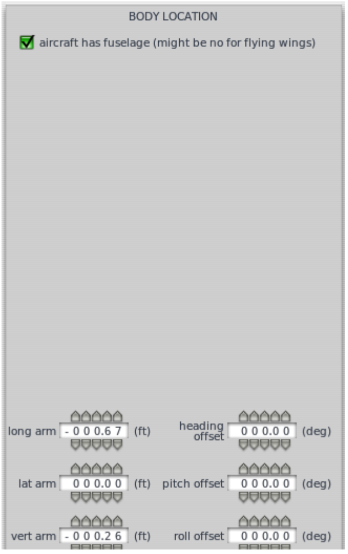

X-System
Fuselage
Section
Top/Bottom
Front/Back
RESET THIS SECTION TO VERTICAL
RESET ALL SECTIONS TO VERTICAL
TOP
SIDE
BOTTOM
LOAD TOP BACKGROUND BITMAP
CLEAR
LOCK ZOOM
LOAD LEFT BACKGROUND BITMAP
CLEAR
LOCK ZOOM

Figure 3.72 Fuselage wire mesh of the FMS 182 Plane Maker model

Plane Maker does not allow for import of 3D models for this process. Although it is possible to alter existing plane models from X-Plane's selection from within Plane Maker, the lack of any scaling feature means that small aircraft on the scale of the target airframe must be modelled from scratch. As such, as it can be seen in the "Section" tab in the above image, 20 cross sections were defined to represent the fuselage of the aircraft.



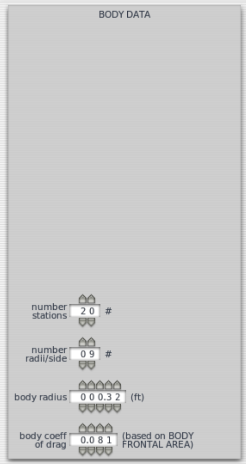
BODY LOCATION

☑ aircraft has fuselage (might be no for flying wings)

long arm - 0 0 0.6 7 (ft)
heading offset 0 0 0.0 0 (deg)
lat arm 0 0 0.0 0 (ft)
pitch offset 0 0 0.0 0 (deg)
vert arm - 0 0 0.2 6 (ft)
roll offset 0 0 0.0 0 (deg)

Figure 3.73 Fuselage positioning in Plane Maker modelling space

Since the target airframe has a fuselage, unlike various designs that rely on only a flying wing, the checkbox in Body Location tab is ticked. All the parameters found in this subsection were set by default to zero because we want our model's datum to start from where the nose of the aircraft is located. The longitudinal, lateral and vertical arm parameters specify the location in 3D space where the tip/nose of the fuselage is located. As such, these numbers were set according to the previous measurements. The target airframe is symmetrical and its traditional fuselage design does not require the design to be rotated in any of its axis by any amount. As such, the offsets for heading, pitch and roll were set to zero.



BODY DATA

number stations 2 0 #

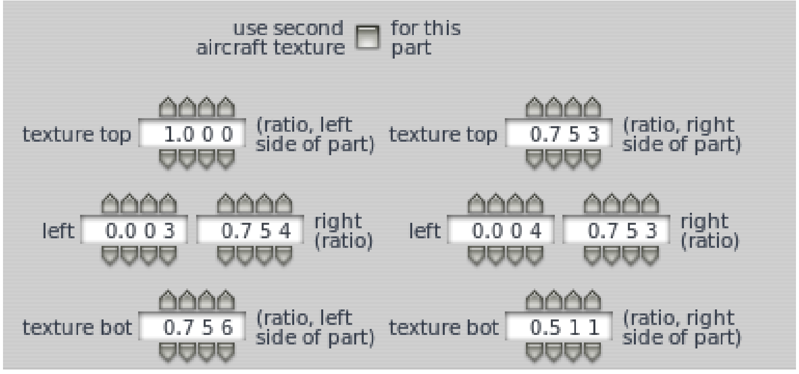
number radii/side 0 9 #

body radius 0 0 0.3 2 (ft)

body coeff of drag 0.0 8 1 (based on BODY FRONTAL AREA)

Figure 3.74 Fuselage construction parameters for the FMS 182 Plane Maker model

The subsection titled BODY DATA contains the 4 editable parameters shown in the above picture. The number of stations represents the number of slices of cross sections needed to represent the geometry of the aircraft (fuselage) with. The idea is that from the nose to the tail of the fuselage, the varying shape of the fuselage can be represented with discrete cross-sectional points (Stations). 20 such Stations were used. The number of sides/radii stands for the number of points that define each cross-section. These points are there to capture how the shape of a given cross-sectional point curves from the top to the bottom of the fuselage (viewed from the front). We have used 9 such points. Since the target airframe is symmetrical, our cross-sectional designs need only consider one side and Plane Maker will automatically mirror this design to generate the complete aerofoil shape. The parameter "body radius" represents the maximum width of the fuselage (set to 0.32 feet). This number is used to constrain the size of the boxes representing the individual stations. Since we are entering data about half the cross-section and then mirroring it due to symmetry, instead of diameter, we consider the radius of the circle originating at the centre of the fuselage at the point of maximum width of the fuselage geometry. The body drag coefficient is set to 0.081 instead of the default value based on calculations made available later in this section.



use second aircraft texture □ for this part

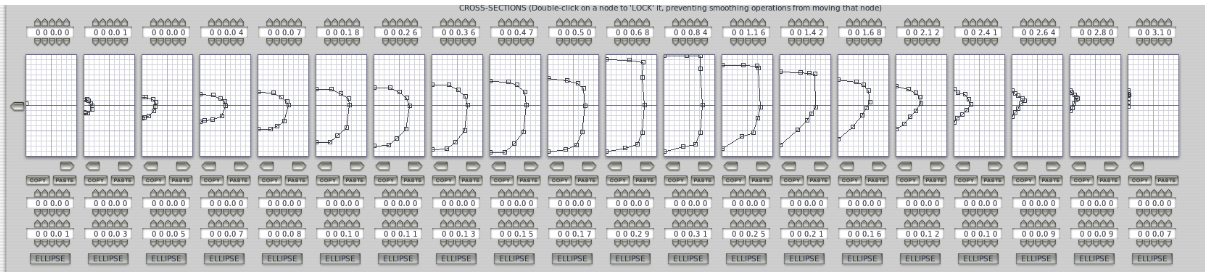
texture top 1.0 0 0 (ratio, left side of part) texture top 0.7 5 3 (ratio, right side of part)

left 0.0 0 3 0.7 5 4 right (ratio) left 0.0 0 4 0.7 5 3 right (ratio)

texture bot 0.7 5 6 (ratio, left side of part) texture bot 0.5 1 1 (ratio, right side of part)

Figure 3.75 Body texturing setting

The section titled BODY TEXTURE relates to the purely aesthetic attributes of the design and makes no functional difference to any of the performance or dynamics of the design. As such, the default values were left as is.



CROSS-SECTIONS (Double-click on a node to 'LOCK' it, preventing smoothing operations from moving that node)

0 0 0.0 0 | 0 0 0.0 1 | 0 0 0.0 0 | 0 0 0.0 4 | 0 0 0.0 7 | 0 0 0.1 8 | 0 0 0.2 6 | 0 0 0.3 6 | 0 0 0.4 7 | 0 0 0.5 0 | 0 0 0.6 8 | 0 0 0.8 4 | 0 0 1.1 6 | 0 0 1.4 2 | 0 0 1.6 8 | 0 0 2.1 2 | 0 0 2.4 1 | 0 0 2.6 4 | 0 0 2.8 0 | 0 0 3.1 0

0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0 | 0 0 0.0 0

0 0 0.0 1 | 0 0 0.0 3 | 0 0 0.0 5 | 0 0 0.0 7 | 0 0 0.0 8 | 0 0 0.1 0 | 0 0 0.1 1 | 0 0 0.1 3 | 0 0 0.1 5 | 0 0 0.1 7 | 0 0 0.2 9 | 0 0 0.3 1 | 0 0 0.2 5 | 0 0 0.2 1 | 0 0 0.1 6 | 0 0 0.1 2 | 0 0 0.1 0 | 0 0 0.0 9 | 0 0 0.0 9 | 0 0 0.0 7

Figure 3.76 Fuselage sectioned into cross-sections for capturing the 3D shape

Within the cross sections is where a visual preview of the 20 stations and 9 radii points selected for the design can be seen. The actual 3D shape is captured in the other two tabs titled "Top/Bottom" and "Front Back". In the above picture, we see some of the cross-sectional points starting from the tip of the nose of the airframe on the left. The first parameter on the top row is set based on the distance of that section from the reference point. The second editable feature is the curve with 9 points itself. These individual points are manipulated with a mouse to capture the shape of the aerofoil. The last two rows of editable parameters enable selection and moving of the radii points left-right/up-down with respect to the reference point for situations where the use of a mouse is less accurate. Despite the process being rather tedious and intimidating, there are techniques that can be used to make it easier and more accurate.

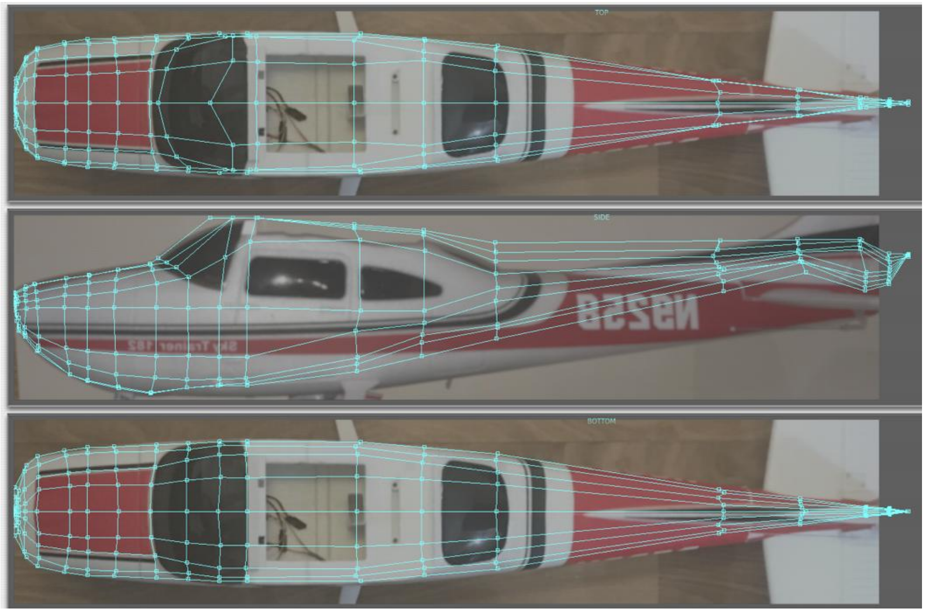


Figure 3.77 Use of photographs to aid in the process of manipulating the individual station points

The above picture depicts the construction process of the 3D mesh. All the stations that were defined and all the individual points contained within each station needs to be manually manipulated to capture the shape of the airframe. Within the tab titled "Top/Bottom, we get the top, side and bottom view of our design. Plane maker allows the loading of any Bitmap image to be used as the background for these sections. What it makes possible to do is to ensure the cross-sectional points line up well with the actual shape of the target airframe. As such, multiple photos of the fuselage was taken from appropriate views and processed until they were usable for this purpose.

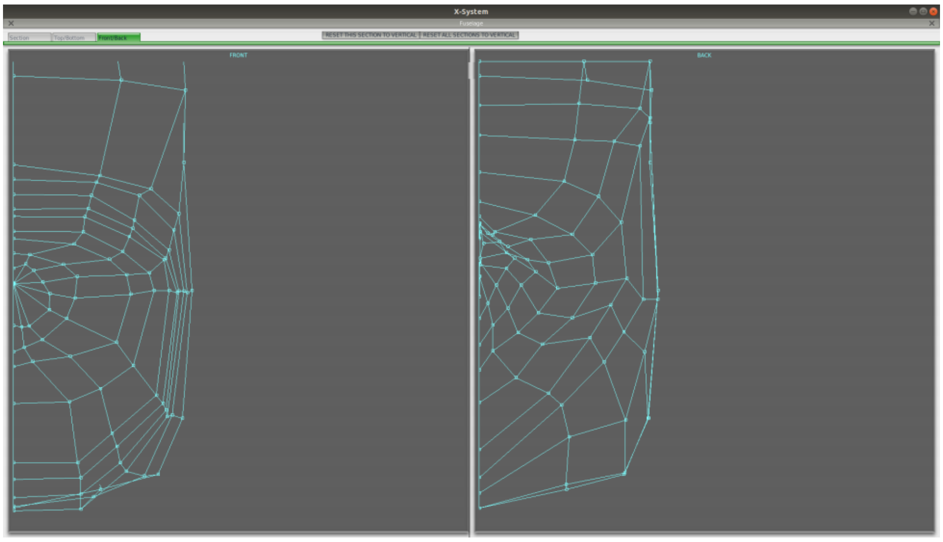
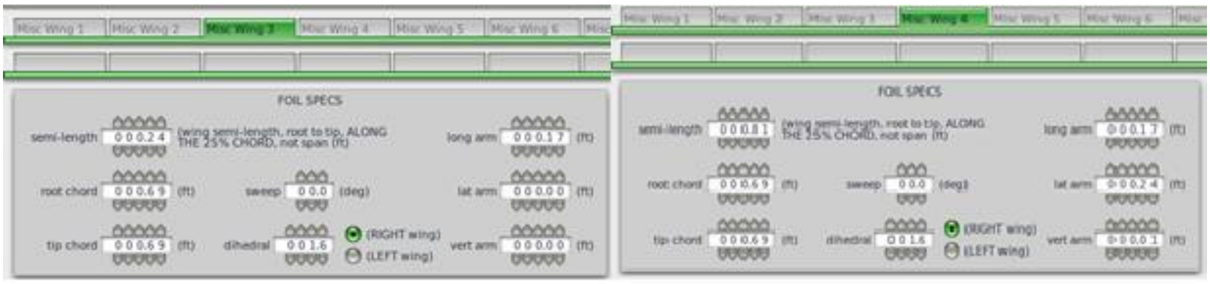

X-System
Fuselage
Section
Top/Bottom
Front/Back
RESET THIS SECTION TO VERTICAL
RESET ALL SECTIONS TO VERTICAL
FRONT
BACK

Figure 3.78 Fuselage front (left) and back wire-meshes for the FMS 182 Plane Maker model

The same process is used for the front and back modelling. The rendering is presented in the results section.

3.3.2 Wing modelling process

Within the GUI of the Plane Maker software, wing modelling is to be found under the menu options Standard>Wings. This is the general method of designing a custom wing where the designer enters geometric data on the wing on one side (left/right) of the airframe and it is automatically mirrored to the other side. However, this is NOT the method that was used. Instead, the section under “Misc Wings” was used to create the wing. The reason being that distinctly separate modelling allows for better diagnostics further down the line. Otherwise, tweaking any settings on one side of the craft is automatically mirrored by Plane Maker and as such no asymmetry can be modelled in easily. The wing was divided into 8 sections (4 on each side of the airplane). Misc Wing 3 - Misc Wing 10 represent these 8 sections.



Misc Wing 3

FOIL SPECS

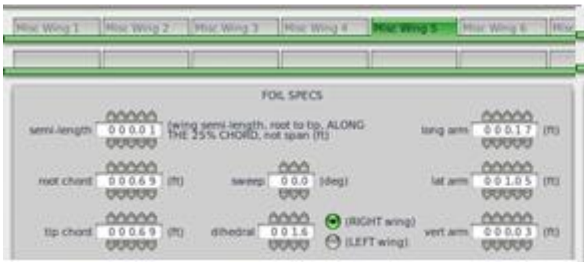
semi-length 0 0 0.2 4 (wing semi-length, root to tip, ALONG THE 25% CHORD, not span (ft)
long arm 0 0 0.1 7 (ft)
root chord 0 0 0.6 9 (ft)
sweep 0 0.0 (deg)
lat arm 0 0 0.0 0 (ft)
tip chord 0 0 0.6 9 (ft)
dihedral 0 0 1.6
● (RIGHT wing)
○ (LEFT wing)
vert arm 0 0 0.0 0 (ft)

Misc Wing 4

FOIL SPECS

semi-length 0 0 0.8 1 (wing semi-length, root to tip, ALONG THE 25% CHORD, not span (ft)
long arm 0 0 0.1 7 (ft)
root chord 0 0 0.6 9 (ft)
sweep 0 0.0 (deg)
lat arm 0 0 0.2 4 (ft)
tip chord 0 0 0.6 9 (ft)
dihedral 0 0 1.6
● (RIGHT wing)
○ (LEFT wing)
vert arm 0 0 0.0 1 (ft)

Figure 3.79 Misc Wing 3 & 4 settings

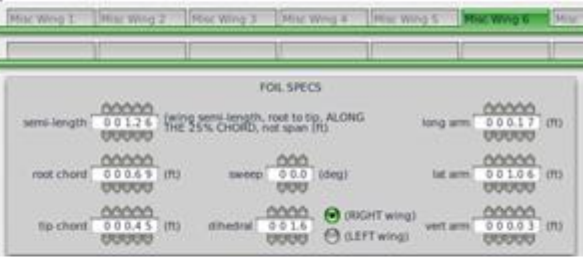


FOIL SPECS

semi-length 000.01 (wing semi-length, root to tip, ALONG THE 25% CHORD, not span (ft)) long arm 000.17 (ft)

root chord 000.69 (ft) sweep 00.0 (deg) lat arm 001.05 (ft)

tip chord 000.69 (ft) dihedral 001.6 ● (RIGHT wing) ○ (LEFT wing) vert arm 000.03 (ft)



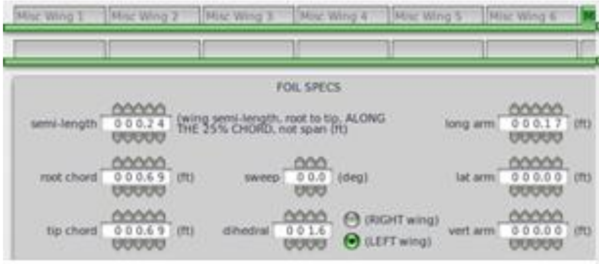
FOIL SPECS

semi-length 001.26 (wing semi-length, root to tip, ALONG THE 25% CHORD, not span (ft)) long arm 000.17 (ft)

root chord 000.69 (ft) sweep 00.0 (deg) lat arm 001.06 (ft)

tip chord 000.45 (ft) dihedral 001.6 ● (RIGHT wing) ○ (LEFT wing) vert arm 000.03 (ft)

Figure 3.80 Misc Wing 5 & 6 settings

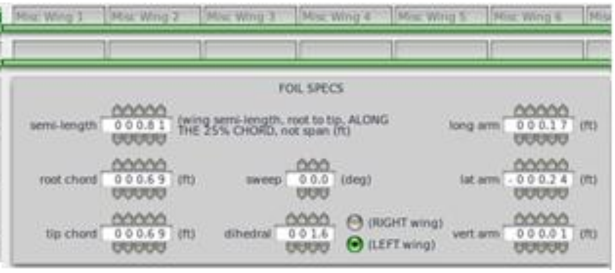


FOIL SPECS

semi-length 000.24 (wing semi-length, root to tip, ALONG THE 25% CHORD, not span (ft)) long arm 000.17 (ft)

root chord 000.69 (ft) sweep 00.0 (deg) lat arm 000.00 (ft)

tip chord 000.69 (ft) dihedral 001.6 ○ (RIGHT wing) ● (LEFT wing) vert arm 000.00 (ft)



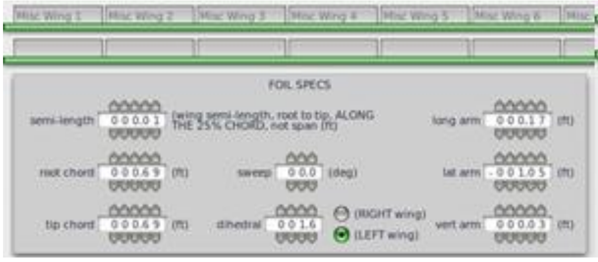
FOIL SPECS

semi-length 000.81 (wing semi-length, root to tip, ALONG THE 25% CHORD, not span (ft)) long arm 000.17 (ft)

root chord 000.69 (ft) sweep 00.0 (deg) lat arm -000.24 (ft)

tip chord 000.69 (ft) dihedral 001.6 ○ (RIGHT wing) ● (LEFT wing) vert arm 000.01 (ft)

Figure 3.81 Misc Wing 7 & 8 settings

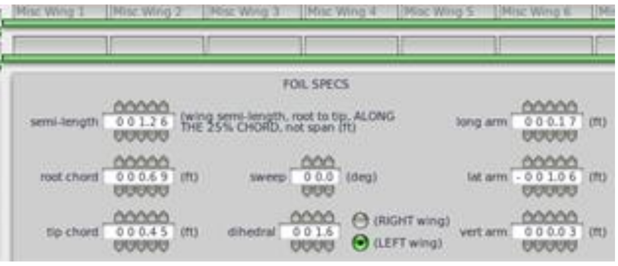


FOIL SPECS

semi-length 000.01 (wing semi-length, root to tip, ALONG THE 25% CHORD, not span (ft)) long arm 000.17 (ft)

root chord 000.69 (ft) sweep 00.0 (deg) lat arm -001.05 (ft)

tip chord 000.69 (ft) dihedral 001.6 ○ (RIGHT wing) ● (LEFT wing) vert arm 000.03 (ft)



FOIL SPECS

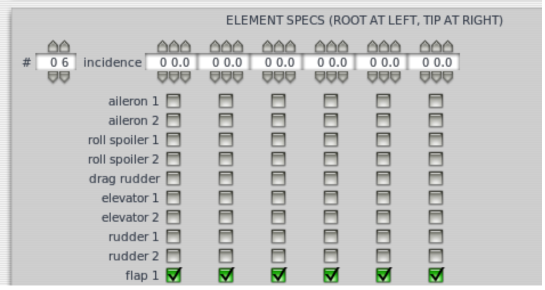
semi-length 001.26 (wing semi-length, root to tip, ALONG THE 25% CHORD, not span (ft)) long arm 000.17 (ft)

root chord 000.69 (ft) sweep 00.0 (deg) lat arm -001.06 (ft)

tip chord 000.45 (ft) dihedral 001.6 ○ (RIGHT wing) ● (LEFT wing) vert arm 000.03 (ft)

Figure 3.82 Misc Wing 9 & 10 settings

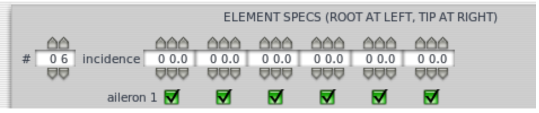
As the above images show, the previously measured geometric data was entered into the FOIL SPECS section of individual wing sections to appropriately position them.



ELEMENT SPECS (ROOT AT LEFT, TIP AT RIGHT)

# 06						
incidence	00.0	00.0	00.0	00.0	00.0	00.0
aileron 1	□	□	□	□	□	□
aileron 2	□	□	□	□	□	□
roll spoiler 1	□	□	□	□	□	□
roll spoiler 2	□	□	□	□	□	□
drag rudder	□	□	□	□	□	□
elevator 1	□	□	□	□	□	□
elevator 2	□	□	□	□	□	□
rudder 1	□	□	□	□	□	□
rudder 2	□	□	□	□	□	□
flap 1	☑	☑	☑	☑	☑	☑

Figure 3.83 Assigning control surfaces (eg. flap) to wing sections

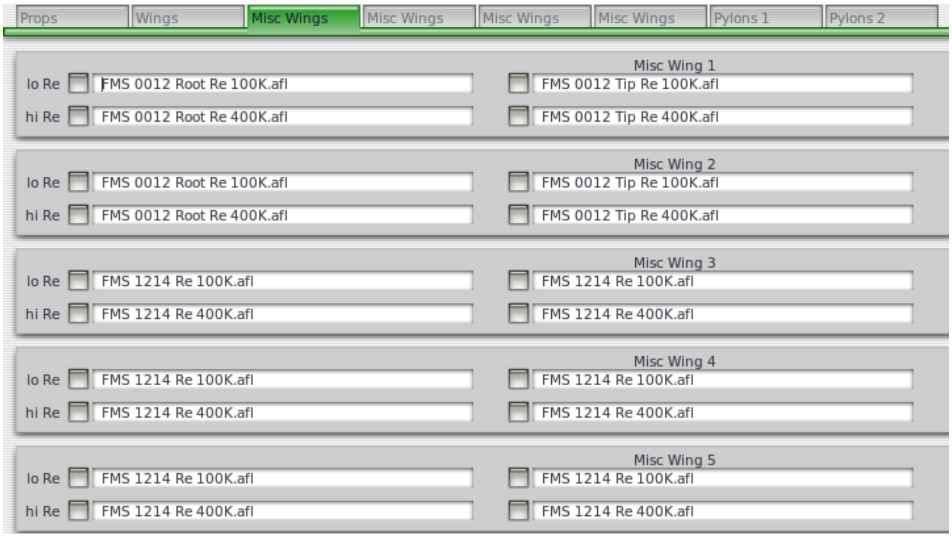


ELEMENT SPECS (ROOT AT LEFT, TIP AT RIGHT)

# 06						
incidence	00.0	00.0	00.0	00.0	00.0	00.0
aileron 1	☑	☑	☑	☑	☑	☑

Figure 3.84 Assigning aileron controls to wing sections

The subsection ELEMENT SPECS allows for appropriate definitions of control surface placements within individual wing sections (Misc Wing 4/6/810). This is vital to configure (based on measurements) for every wing section because the target airframe has control surfaces (ailerons and flaps) attached to them at certain points. It's also important to note here that these settings work based on the absolute values entered in the Control Geometry section. Misc Wing 3 is the section of the wing doubling as the right-half of the roof of the fuselage. Misc Wing 4 is the inner wing section containing the flap. Misc Wing 5 is the small slice of the wing between Misc Wing 4 and Misc Wing 6 (contains the ailerons). This exact pattern is repeated on the left side (port) of the aircraft.



Misc Wing 1
lo Re □ FMS 0012 Root Re 100K.afl □ FMS 0012 Tip Re 100K.afl
hi Re □ FMS 0012 Root Re 400K.afl □ FMS 0012 Tip Re 400K.afl

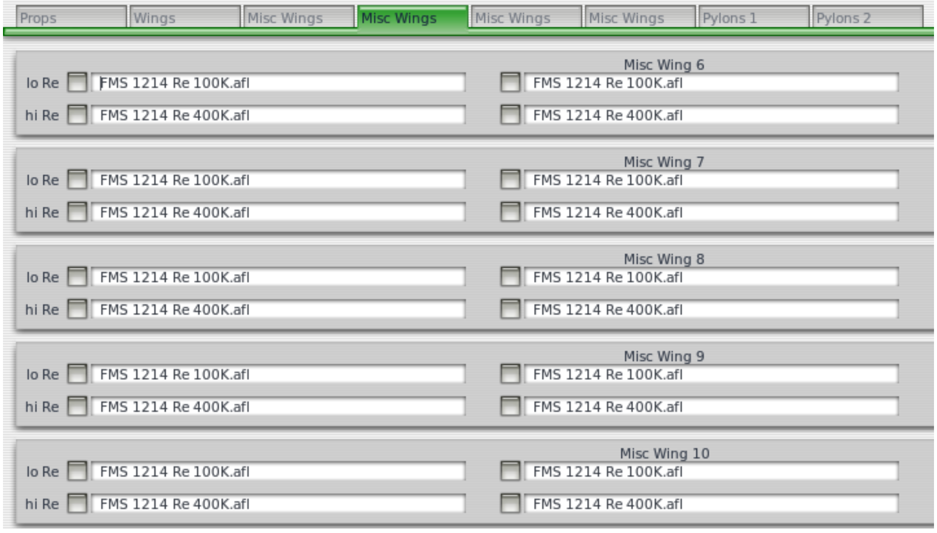
Misc Wing 2
lo Re □ FMS 0012 Root Re 100K.afl □ FMS 0012 Tip Re 100K.afl
hi Re □ FMS 0012 Root Re 400K.afl □ FMS 0012 Tip Re 400K.afl

Misc Wing 3
lo Re □ FMS 1214 Re 100K.afl □ FMS 1214 Re 100K.afl
hi Re □ FMS 1214 Re 400K.afl □ FMS 1214 Re 400K.afl

Misc Wing 4
lo Re □ FMS 1214 Re 100K.afl □ FMS 1214 Re 100K.afl
hi Re □ FMS 1214 Re 400K.afl □ FMS 1214 Re 400K.afl

Misc Wing 5
lo Re □ FMS 1214 Re 100K.afl □ FMS 1214 Re 100K.afl
hi Re □ FMS 1214 Re 400K.afl □ FMS 1214 Re 400K.afl

Figure 3.85 Assignment of custom FMS 182 aerofoils (from XFLR5/MATLAB) to Misc Wing 1-5



Misc Wing 6
lo Re □ FMS 1214 Re 100K.afl □ FMS 1214 Re 100K.afl
hi Re □ FMS 1214 Re 400K.afl □ FMS 1214 Re 400K.afl

Misc Wing 7
lo Re □ FMS 1214 Re 100K.afl □ FMS 1214 Re 100K.afl
hi Re □ FMS 1214 Re 400K.afl □ FMS 1214 Re 400K.afl

Misc Wing 8
lo Re □ FMS 1214 Re 100K.afl □ FMS 1214 Re 100K.afl
hi Re □ FMS 1214 Re 400K.afl □ FMS 1214 Re 400K.afl

Misc Wing 9
lo Re □ FMS 1214 Re 100K.afl □ FMS 1214 Re 100K.afl
hi Re □ FMS 1214 Re 400K.afl □ FMS 1214 Re 400K.afl

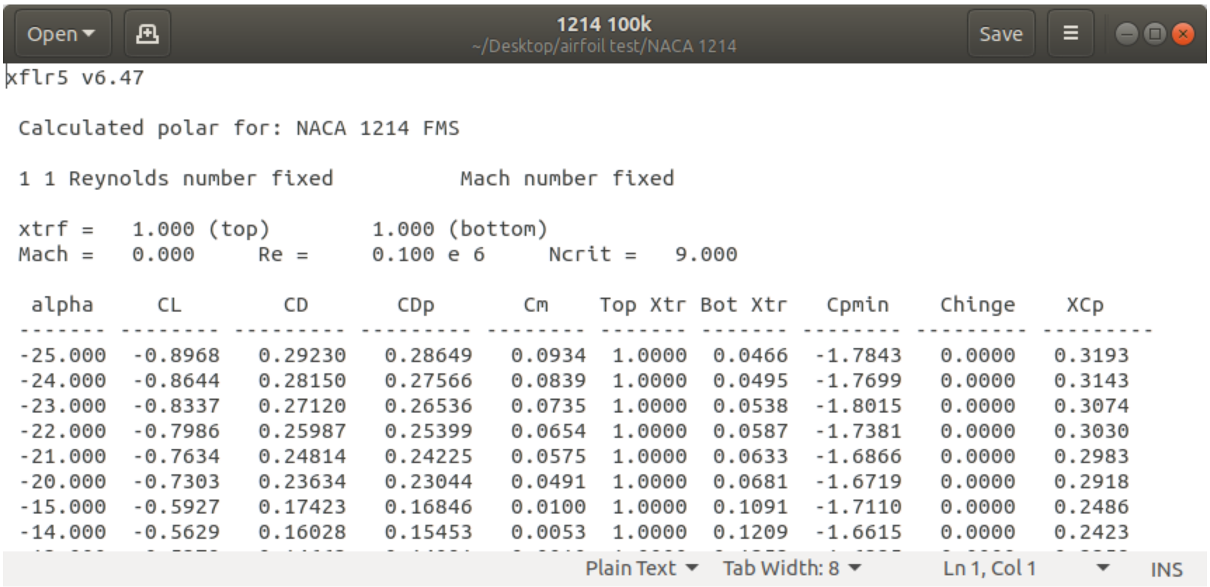
Misc Wing 10
lo Re □ FMS 1214 Re 100K.afl □ FMS 1214 Re 100K.afl
hi Re □ FMS 1214 Re 400K.afl □ FMS 1214 Re 400K.afl

Figure 3.86 Assignment of custom FMS 182 aerofoils (from XFLR5/MATLAB) to Misc Wing 6-10

For accurate aerodynamic behaviour of the wings, these wing sections need to be assigned aerofoils. While X-Plane does have similar aerofoils, they are unreliable as they almost always miss data within the operational Reynolds number ranges for such small aircraft. As such, XFLR5 was utilised to create custom aerofoils for this target airframe's X-Plane model. As it can be seen in the above image, every individual section had to be provided with 2 aerofoil files with both low and high Reynolds numbers. X-Plane interpolates between these two ranges to work out the overall aerodynamic behaviour.

3.3.3 Aerofoil Creation Process

There exists no documentation on this process. As such, extensive amounts of trial and error had to be done to find the following method of successful aerofoil generation.

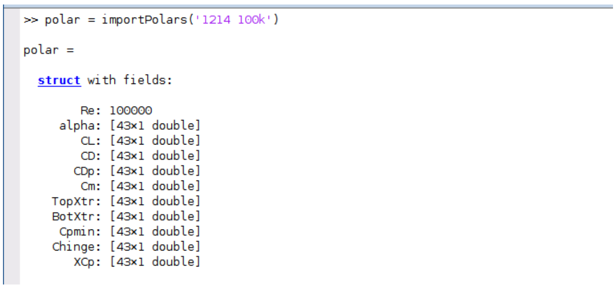


```
xflr5 v6.47

 Calculated polar for: NACA 1214 FMS

 1 1 Reynolds number fixed          Mach number fixed

 xtrf =   1.000 (top)        1.000 (bottom)
 Mach =   0.000     Re =     0.100 e 6     Ncrit =   9.000

  alpha     CL        CD       CDp       Cm    Top Xtr  Bot Xtr   Cpmin    Chinge     XCp
 ------- -------- --------- --------- -------- -------- -------- -------- --------- ---------
 -25.000  -0.8968   0.29230   0.28649   0.0934  1.0000  0.0466  -1.7843   0.0000   0.3193
 -24.000  -0.8644   0.28150   0.27566   0.0839  1.0000  0.0495  -1.7699   0.0000   0.3143
 -23.000  -0.8337   0.27120   0.26536   0.0735  1.0000  0.0538  -1.8015   0.0000   0.3074
 -22.000  -0.7986   0.25987   0.25399   0.0654  1.0000  0.0587  -1.7381   0.0000   0.3030
 -21.000  -0.7634   0.24814   0.24225   0.0575  1.0000  0.0633  -1.6866   0.0000   0.2983
 -20.000  -0.7303   0.23634   0.23044   0.0491  1.0000  0.0681  -1.6719   0.0000   0.2918
 -15.000  -0.5927   0.17423   0.16846   0.0100  1.0000  0.1091  -1.7110   0.0000   0.2486
 -14.000  -0.5629   0.16028   0.15453   0.0053  1.0000  0.1209  -1.6615   0.0000   0.2423
```

Figure 3.87 XFLR5 data for the main wing aerofoil creation

The previously generated XFLR5 aerofoils were utilised for this purpose. Custom MATLAB code was written to aid in the process. XFLR5 dumps CSV data in a very specific format and X-Plane's aerofoil files have to be written in a unique format. As such two functions (importPolars.m & dumpAfl.m) were written to make that process easier for any number of aerofoils in the future.

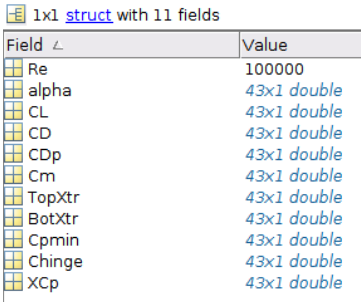
3.3.3.1 importPolars.m



```
>> polar = importPolars('1214 100k')

polar =

  struct with fields:

        Re: 100000
     alpha: [43×1 double]
        CL: [43×1 double]
        CD: [43×1 double]
       CDp: [43×1 double]
        Cm: [43×1 double]
    TopXtr: [43×1 double]
    BotXtr: [43×1 double]
     Cpmin: [43×1 double]
    Chinge: [43×1 double]
       XCp: [43×1 double]
```

Figure 3.88 MATLAB import process

The custom function imports all the data from XFLR5 onto MATLAB’s workspace and stores it as a structure.



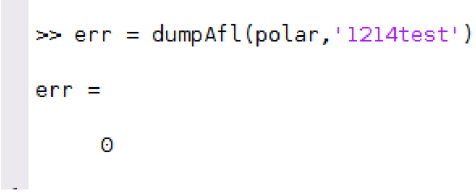
1x1 struct with 11 fields

Field	Value
Re	100000
alpha	43x1 double
CL	43x1 double
CD	43x1 double
CDp	43x1 double
Cm	43x1 double
TopXtr	43x1 double
BotXtr	43x1 double
Cpmin	43x1 double
Chinge	43x1 double
XCp	43x1 double

Figure 3.89 MATLAB workspace storing relevant XFLR5 data

As it can be seen from the above image, all the relevant values have been extracted from the XFLR5 csv file and stored in the form of tables that can be then used by the function dumpAfl.m.

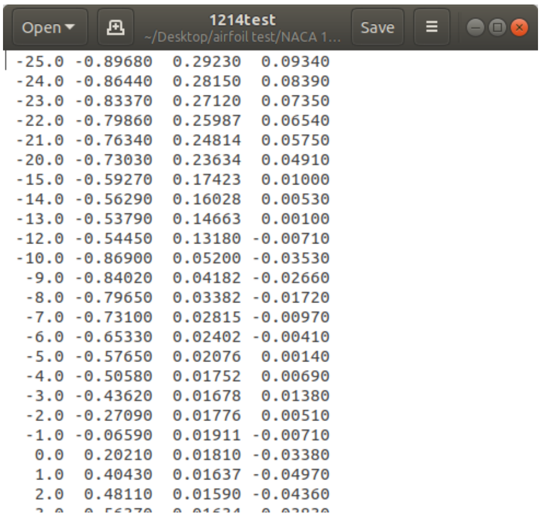
3.3.3.2 dumpAfl.m



```
>> err = dumpAfl(polar,'1214test')

err =

     0
```

Figure 3.90 MATLAB function to create .afl dataset

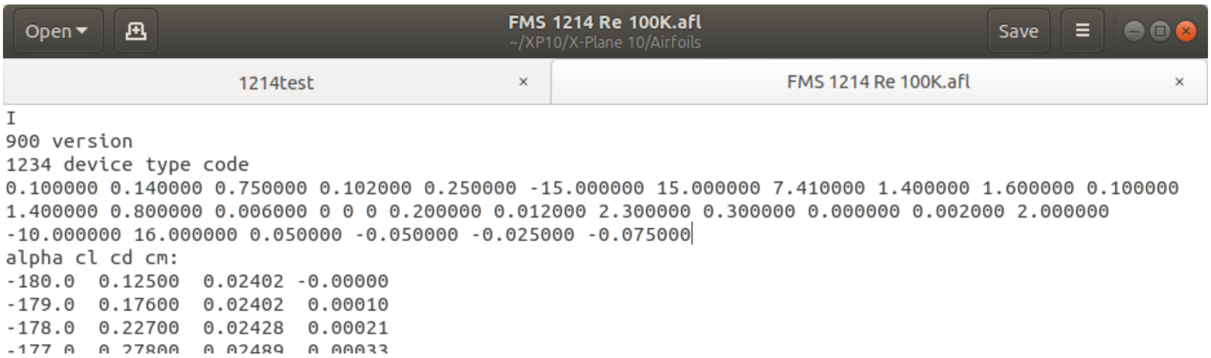
All this custom function does is to use the imported data (stored as a structure called polar in the screenshot) to generate a file with the columns that need to be injected on to an existing X-Plane aerofoil file. An error check has been built into it. It should always return 0 (see screenshot) if the process was successful. The output of this function looks like as shown in the following image (Figure 3.91).



```
-25.0 -0.89680  0.29230  0.09340
-24.0 -0.86440  0.28150  0.08390
-23.0 -0.83370  0.27120  0.07350
-22.0 -0.79860  0.25987  0.06540
-21.0 -0.76340  0.24814  0.05750
-20.0 -0.73030  0.23634  0.04910
-15.0 -0.59270  0.17423  0.01000
-14.0 -0.56290  0.16028  0.00530
-13.0 -0.53790  0.14663  0.00100
-12.0 -0.54450  0.13180 -0.00710
-10.0 -0.86900  0.05200 -0.03530
 -9.0 -0.84020  0.04182 -0.02660
 -8.0 -0.79650  0.03382 -0.01720
 -7.0 -0.73100  0.02815 -0.00970
 -6.0 -0.65330  0.02402 -0.00410
 -5.0 -0.57650  0.02076  0.00140
 -4.0 -0.50580  0.01752  0.00690
 -3.0 -0.43620  0.01678  0.01380
 -2.0 -0.27090  0.01776  0.00510
 -1.0 -0.06590  0.01911 -0.00710
  0.0  0.20210  0.01810 -0.03380
  1.0  0.40430  0.01637 -0.04970
  2.0  0.48110  0.01590 -0.04360
```

Figure 3.91 A test sample aerofoil file data output from MATLAB

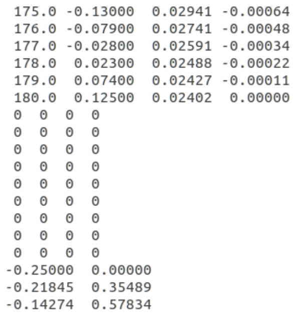
This is the general procedure followed for every custom aerofoil created for this research. One of the standard templates of aerofoil files provided by X-Plane (.afl format) is used for the final steps. The header contains information that is used by the simulator, some of which need to be altered to match the target aerofoil. Namely, the Reynolds number and Aspect Ratio. As they are not labelled in the .afl file header, they had to be experimentally identified. The rest of the other parameters in the header don’t interfere with the process and is only relevant if Airfoil Maker is used to edit the aerofoil file.



```
I
900 version
1234 device type code
0.100000 0.140000 0.750000 0.102000 0.250000 -15.000000 15.000000 7.410000 1.400000 1.600000 0.100000
1.400000 0.800000 0.006000 0 0 0 0.200000 0.012000 2.300000 0.300000 0.000000 0.002000 2.000000
-10.000000 16.000000 0.050000 -0.050000 -0.025000 -0.075000
alpha cl cd cm:
-180.0  0.12500  0.02402 -0.00000
-179.0  0.17600  0.02402  0.00010
-178.0  0.22700  0.02428  0.00021
```

Figure 3.92 X-Plane .afl file headers

The data prepared by the custom MATLAB functions were manually entered into the aerofoil file by replacing the existing data.



```
175.0 -0.13000  0.02941 -0.00064
176.0 -0.07900  0.02741 -0.00048
177.0 -0.02800  0.02591 -0.00034
178.0  0.02300  0.02488 -0.00022
179.0  0.07400  0.02427 -0.00011
180.0  0.12500  0.02402  0.00000
0  0  0  0
0  0  0  0
0  0  0  0
0  0  0  0
0  0  0  0
0  0  0  0
0  0  0  0
0  0  0  0
0  0  0  0
-0.25000  0.00000
-0.21845  0.35489
-0.14274  0.57834
```

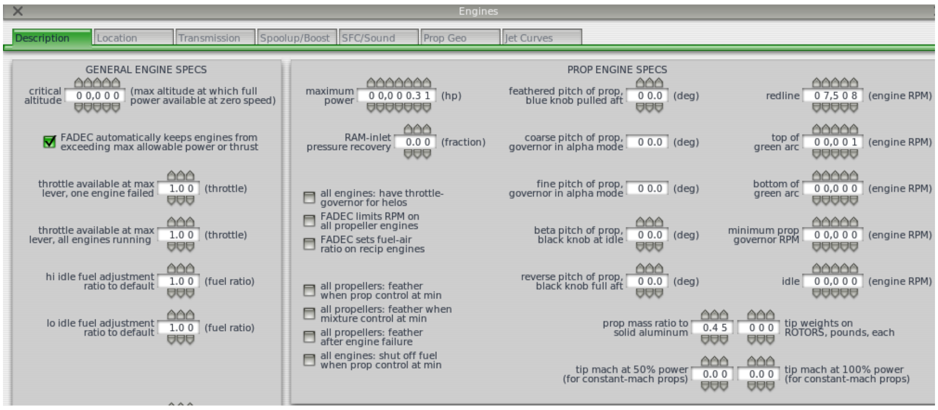
Figure 3.93 Padding the aerofoil file with rows of zeros

It was discovered that the modified aerofoil file doesn't work with X-Plane at this stage. On further experimentation, it was found that the total number of lines on that file must be 740. This is why the trailing "0 0 0 0" dummy values needed to be entered into the aerofoil file to make the file have a total number of lines equalling to 740. All the values following the "0 0 0 0" padding represent the visual rendering of the aerofoil shape. The imported polars can be seen in the results section.

3.3.4 Horizontal & Vertical Stabiliser modelling process

The modelling for these components follows a similar approach to wing modelling. As such, the details are left out of the main body of the thesis but included within the Appendix (Chapter 7.5)

3.3.5 Power plant modelling



GENERAL ENGINE SPECS

critical altitude 0 0,0 0 0 (max altitude at which full power available at zero speed)

☑ FADEC automatically keeps engines from exceeding max allowable power or thrust

throttle available at max lever, one engine failed 1.0 0 (throttle)

throttle available at max lever, all engines running 1.0 0 (throttle)

hi idle fuel adjustment ratio to default 1.0 0 (fuel ratio)

lo idle fuel adjustment ratio to default 1.0 0 (fuel ratio)

maximum power 0 0,0 0 0.3 1 (hp)

RAM-inlet pressure recovery 0.0 0 (fraction)

□ all engines: have throttle-governor for helos

□ FADEC limits RPM on all propeller engines

□ FADEC sets fuel-air ratio on recip engines

□ all propellers: feather when prop control at min

□ all propellers: feather when mixture control at min

□ all propellers: feather after engine failure

□ all engines: shut off fuel when prop control at min

PROP ENGINE SPECS

feathered pitch of prop, blue knob pulled aft 0 0.0 (deg)

coarse pitch of prop, governor in alpha mode 0 0.0 (deg)

fine pitch of prop, governor in alpha mode 0 0.0 (deg)

beta pitch of prop, black knob at idle 0 0.0 (deg)

reverse pitch of prop, black knob full aft 0 0.0 (deg)

redline 0 7,5 0 8 (engine RPM)

top of green arc 0 0,0 0 1 (engine RPM)

bottom of green arc 0 0,0 0 0 (engine RPM)

minimum prop governor RPM 0 0,0 0 0 (engine RPM)

idle 0 0,0 0 0 (engine RPM)

prop mass ratio to solid aluminum 0.4 5

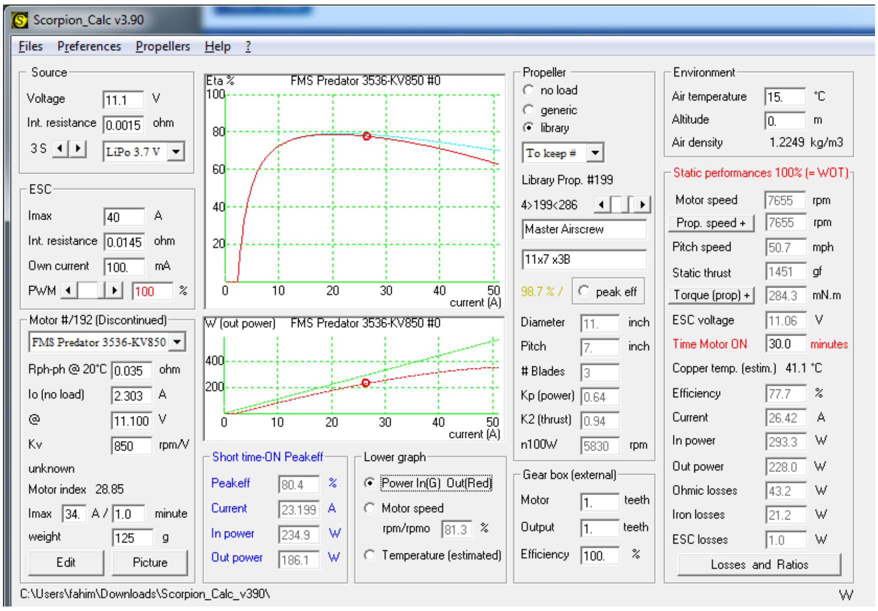
0 0 0 tip weights on ROTORS, pounds, each

tip mach at 50% power (for constant-mach props) 0.0 0

0.0 0 tip mach at 100% power (for constant-mach props)

Figure 3.94 Engine modelling settings in Plane Maker

X-Plane lacks any dedicated way of modelling the type of Brushless DC motor based engines used for such small fixed-wing aircraft. As shown in the above image (Figure 3.94), various details about generic aviation engines can be entered into the simulator. However, that would result in very low fidelity modelling. As such, alternative tools and methods were employed to determine equivalent values for the parameters that can be entered into Plane Maker.



Source
Voltage 11.1 V
Int. resistance 0.0015 ohm
3 S LiPo 3.7 V

ESC
Imax 40 A
Int. resistance 0.0145 ohm
Own current 100. mA
PWM 100 %

Motor #/192 (Discontinued)
FMS Predator 3536-KV850
Rph-ph @ 20°C 0.035 ohm
Io (no load) 2.303 A
@ 11.100 V
Kv 850 rpm/V
unknown
Motor index 28.85
Imax 34. A / 1.0 minute
weight 125 g

Short time-ON Peakeff
Peakeff 80.4 %
Current 23.199 A
In power 234.9 W
Out power 186.1 W

Lower graph
● Power In(G) Out(Red)
○ Motor speed
rpm/rpmo 81.3 %
○ Temperature (estimated)

Propeller
○ no load
○ generic
● library
To keep #
Library Prop. #199
4>199<286
Master Airscrew
11x7 x3B
98.7 % / ○ peak eff
Diameter 11. inch
Pitch 7. inch
Blades 3
Kp (power) 0.64
K2 (thrust) 0.94
n100W 5830 rpm

Gear box (external)
Motor 1. teeth
Output 1. teeth
Efficiency 100. %

Environment
Air temperature 15. °C
Altitude 0. m
Air density 1.2249 kg/m3

Static performances 100% (= WOT)
Motor speed 7655 rpm
Prop. speed + 7655 rpm
Pitch speed 50.7 mph
Static thrust 1451 gf
Torque (prop) + 284.3 mN.m
ESC voltage 11.06 V
Time Motor ON 30.0 minutes
Copper temp. (estim.) 41.1 °C
Efficiency 77.7 %
Current 26.42 A
In power 293.3 W
Out power 228.0 W
Ohmic losses 43.2 W
Iron losses 21.2 W
ESC losses 1.0 W

C:\Users\fahim\Downloads\Scorpion_Calc_v390\

Figure 3.95 Scorpion_calc results for FMS 182 power plant setup

Scorpion_Calc v 3.90 [7] was used to estimate the overall power output and RPM limits. The software is free and has a database of brushless DC motors, propellers and ESCs all of which makes the process of obtaining estimation much easier. However, this wasn’t used alone.

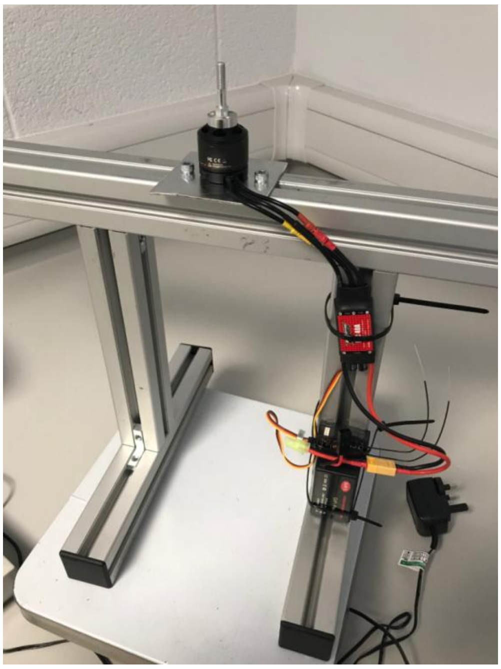


Figure 3.96 FMS 182’s power plant setup on a test rig

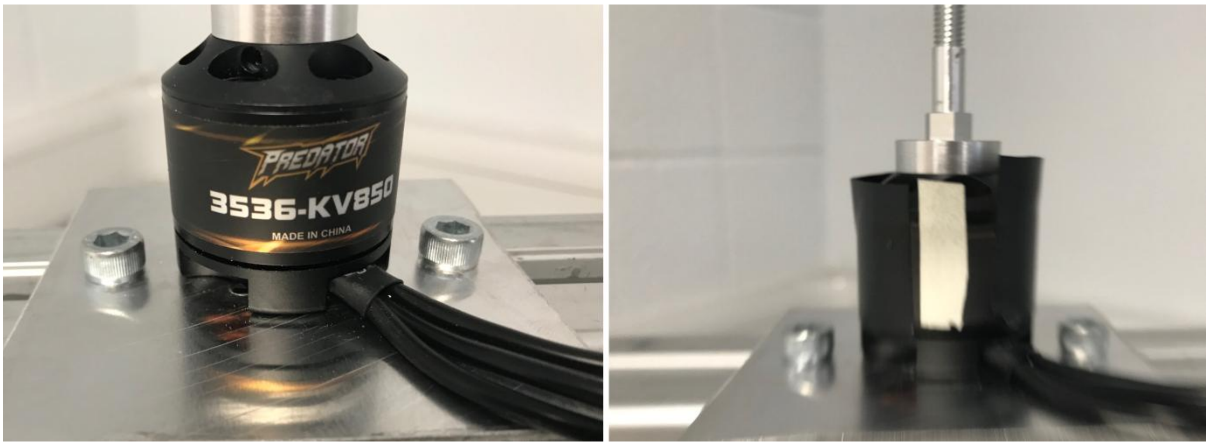


Figure 3.97 Brushless DC motor mounting (with black masking tape and reflective strip on right)

The Motor, Battery, ESC, Radio receiver and transmitter and propeller were tested in two different conditions. In the first test, the setup without the propeller was mounted on a test rig as shown in the above images. A custom attachment plate was necessary due to the test facility’s attachments not fitting this small a motor (see Appendix). A reflective tape was placed on the spinning body of the motor. To reduce interference from any shiny part of the body, black tape was used to mask the motor’s body and then the reflective tape was placed on it. A Rotaro digital optical tachometer [8] was used to read the readings from the spinning motor in max throttle situation.

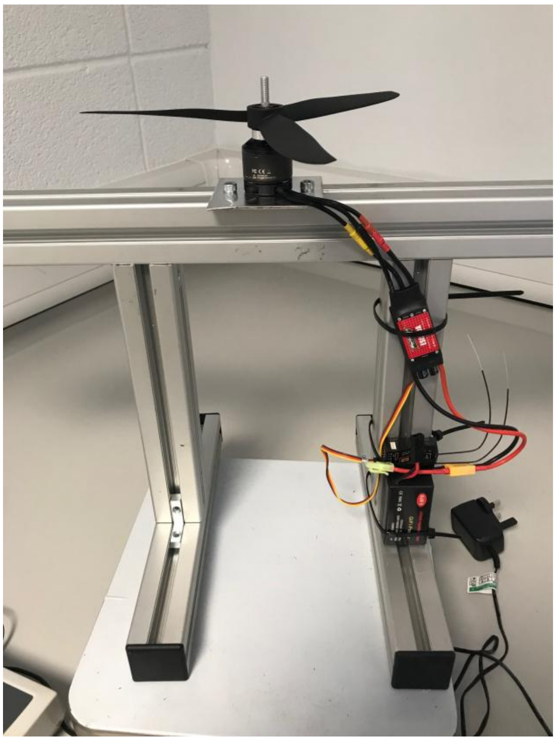


Figure 3.98 Power plant test setup with propeller on

For the second test, the entire setup including the propeller was mounted on the test rig which was resting on a digital scale. The propeller required to be mounted opposite the usual direction.

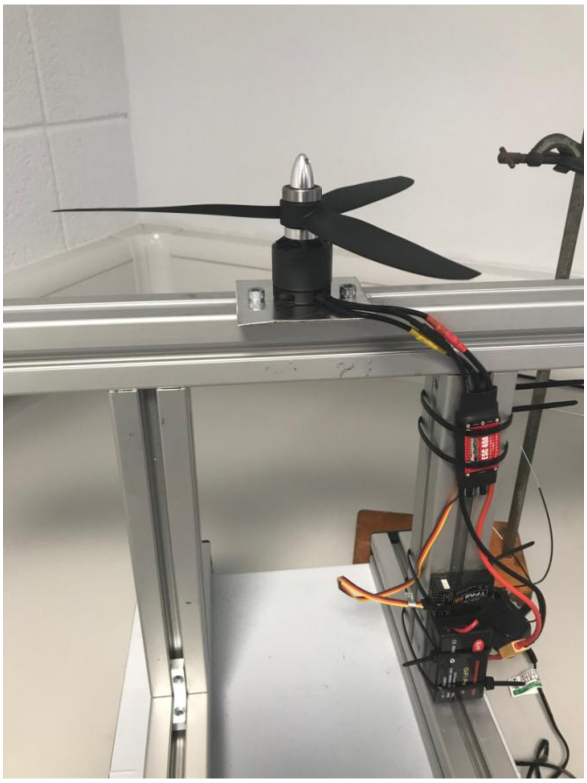


Figure 3.99 Powerplant setup on test rig with propeller and pinner on

A separate attachment was needed to secure the propeller in place along with spinner. The idea behind the test is to estimate the maximum thrust produced by making the propeller spin at peak throttle setting.

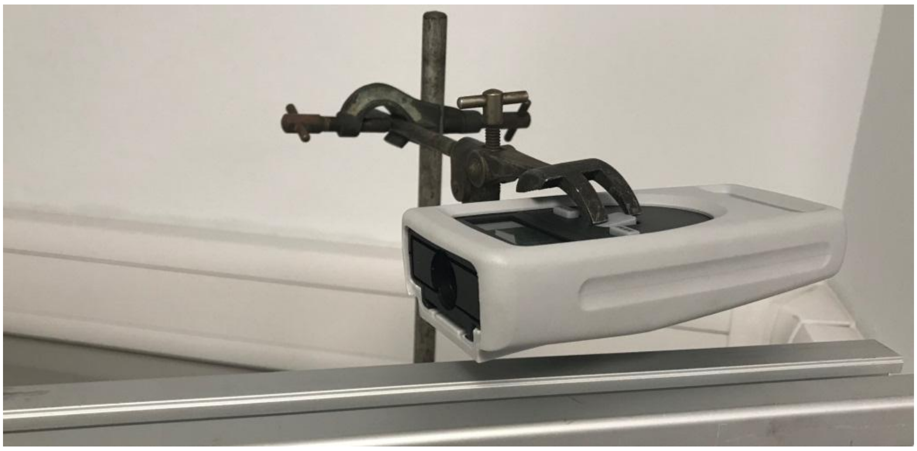


Figure 3.100 Clamp and optical tachometer

Given the dangerous nature of operating such propellers under such indoor test conditions, a metal clamp was used to hold the tachometer within effective range of the reflective tape while maintaining a safe distance away from the live test rig.

Figure 3.101 Digital scale zeroed before test

Since the test rig is mounted on digital scale (Figure 3.101), once the setup was mounted on the rig, the scale was set to zero to help in the measurement process. All the recorded test results are documented in the corresponding results section.

3.3.5.1 Propeller Modelling:

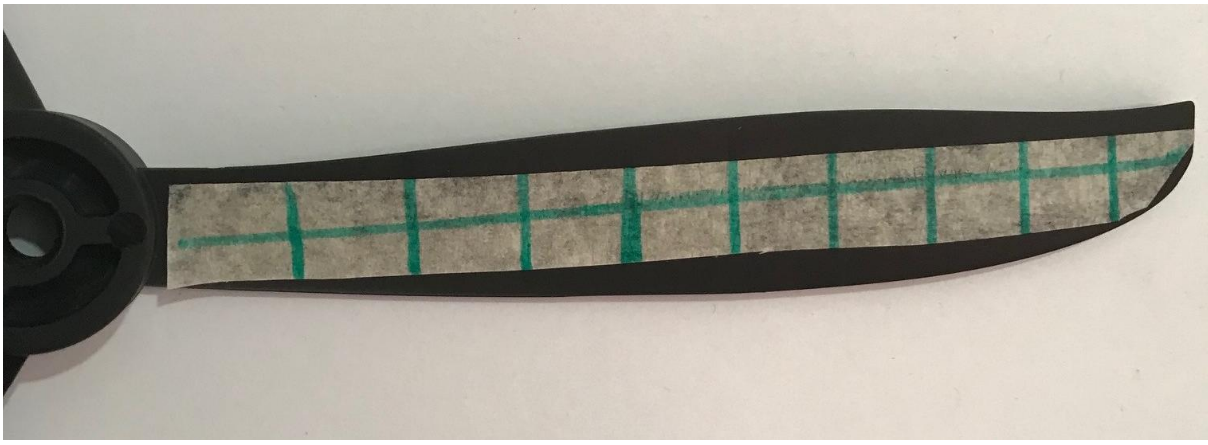
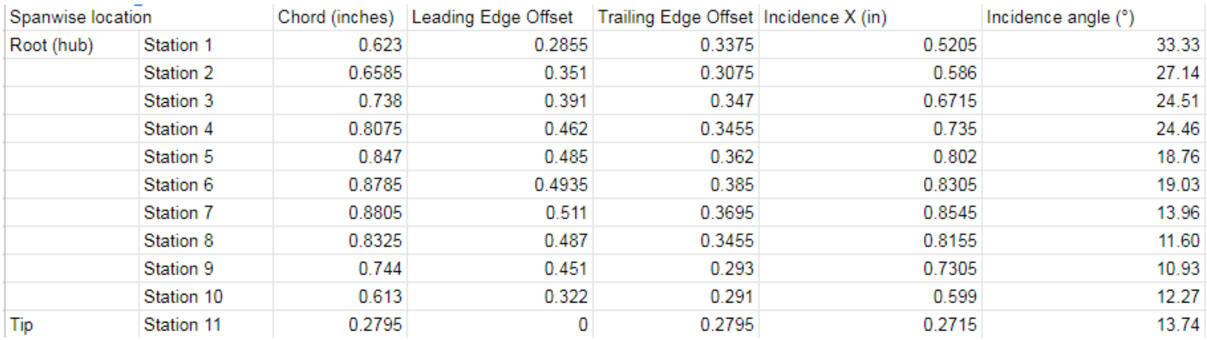


Figure 3.102 FMS 182 propeller with masking tape and green markers for measurements

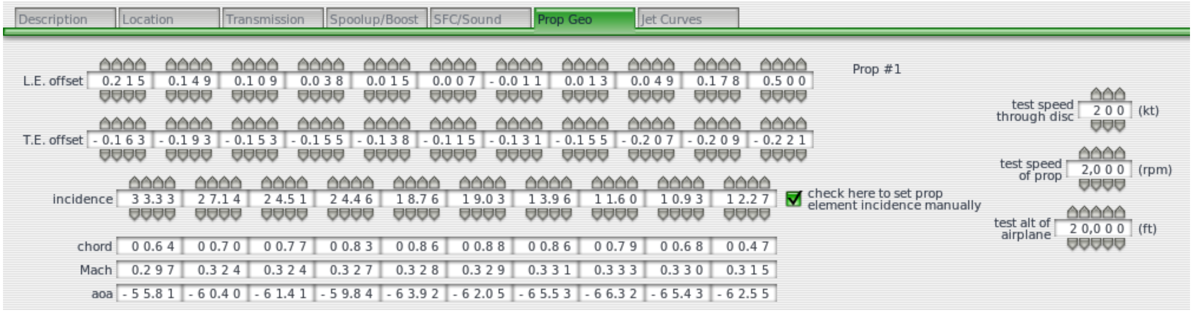
The way Plane Maker allows for propeller modelling is through segmentation of the aerofoil into 11 sections (Figure 3.102) from the hub to the tip of a propeller blade. As shown in the above image (Figure 3.102), white masking tape was used along with a green marker to create these sections and measurements were taken using a digital calliper and protractor.



Spanwise location		Chord (inches)	Leading Edge Offset	Trailing Edge Offset	Incidence X (in)	Incidence angle (°)
Root (hub)	Station 1	0.623	0.2855	0.3375	0.5205	33.33
	Station 2	0.6585	0.351	0.3075	0.586	27.14
	Station 3	0.738	0.391	0.347	0.6715	24.51
	Station 4	0.8075	0.462	0.3455	0.735	24.46
	Station 5	0.847	0.485	0.362	0.802	18.76
	Station 6	0.8785	0.4935	0.385	0.8305	19.03
	Station 7	0.8805	0.511	0.3695	0.8545	13.96
	Station 8	0.8325	0.487	0.3455	0.8155	11.60
	Station 9	0.744	0.451	0.293	0.7305	10.93
	Station 10	0.613	0.322	0.291	0.599	12.27
Tip	Station 11	0.2795	0	0.2795	0.2715	13.74

Figure 3.103 Data table used to calculate X-Plane inputs for the propeller geometry

The data were then entered onto the following table under Prop Geo section.



Prop #1

L.E. offset	0.215	0.149	0.109	0.038	0.015	0.007	-0.011	0.013	0.049	0.178	0.500
T.E. offset	-0.163	-0.193	-0.153	-0.155	-0.138	-0.115	-0.131	-0.155	-0.207	-0.209	-0.221
incidence	33.33	27.14	24.51	24.46	18.76	19.03	13.96	11.60	10.93	12.27	
chord	00.64	00.70	00.77	00.83	00.86	00.88	00.86	00.79	00.68	00.47	
Mach	0.297	0.324	0.324	0.327	0.328	0.329	0.331	0.333	0.330	0.315	
aoa	-55.81	-60.40	-61.41	-59.84	-63.92	-62.05	-65.53	-66.32	-65.43	-62.55	

☑ check here to set prop element incidence manually

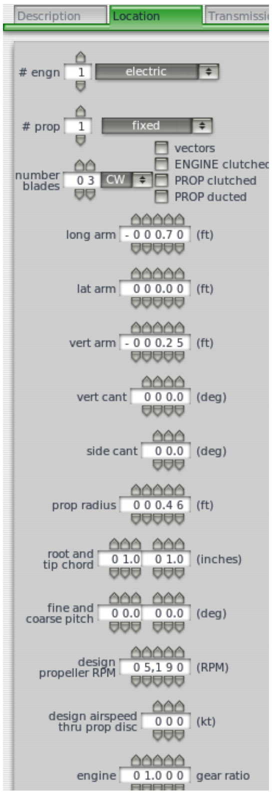
test speed through disc 200 (kt)

test speed of prop 2,000 (rpm)

test alt of airplane 20,000 (ft)

Figure 3.104 Plane Maker propeller settings

The measured offsets from the long green centreline (running from the hub to the tip) of the blade were entered into the L.E and T.E offset fields. The incidence angles were set accordingly. As it can be seen in the above image, there are additional fields (chord, Mach, AoA, etc.). These fields are not editable and are meant to be used along with the values on the right to check the Mach numbers for sanity. This was used with varying inputs on the left to ensure the propeller tip doesn’t go supersonic. Unfortunately, Plane Maker doesn’t have a way of plotting these values across a range of variable to be checked graphically.

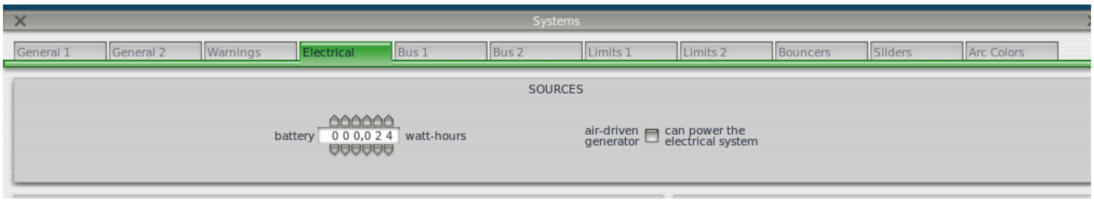


engn 1 electric
prop 1 fixed
□ vectors
□ ENGINE clutched
□ PROP clutched
□ PROP ducted
number blades 0 3 CW
long arm - 0 0 0.7 0 (ft)
lat arm 0 0 0.0 0 (ft)
vert arm - 0 0 0.2 5 (ft)
vert cant 0 0 0.0 (deg)
side cant 0 0.0 (deg)
prop radius 0 0 0.4 6 (ft)
root and tip chord 0 1.0 0 1.0 (inches)
fine and coarse pitch 0 0.0 0 0.0 (deg)
design propeller RPM 0 5,1 9 0 (RPM)
design airspeed thru prop disc 0 0 0 (kt)
engine 0 1.0 0 0 gear ratio

Figure 3.105 Plane Maker Engine and propeller position and settings for FMS 182 model

The actual number of propeller blades, the locations and other relevant parameters were entered into the same tab (Location) under Engine specs.

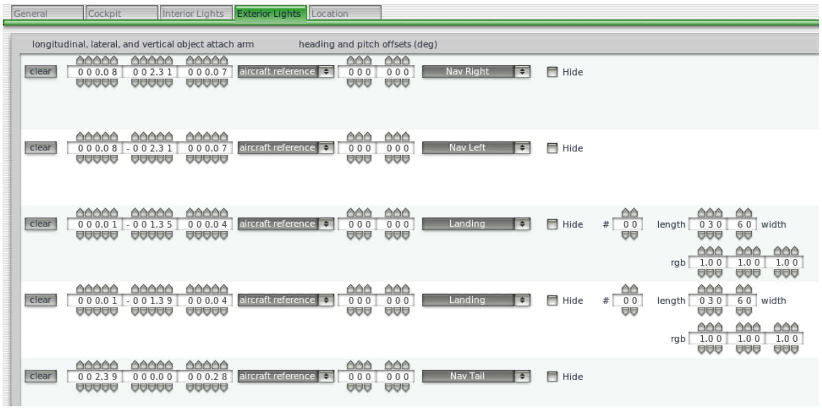
3.3.6 Electrical systems and external lights



SOURCES
battery 0 0 0,0 2 4 watt-hours
air-driven generator □ can power the electrical system

Figure 3.106 Electrical power settings in Plane Maker

For the battery modelling, a power source had to be defined under Systems/Electrical. This provides a ceiling of the max power made available to the systems. Any error or change of this number isn’t critical as the max available horsepower is also explicitly defined in the engine settings based on the results presented in the corresponding section.



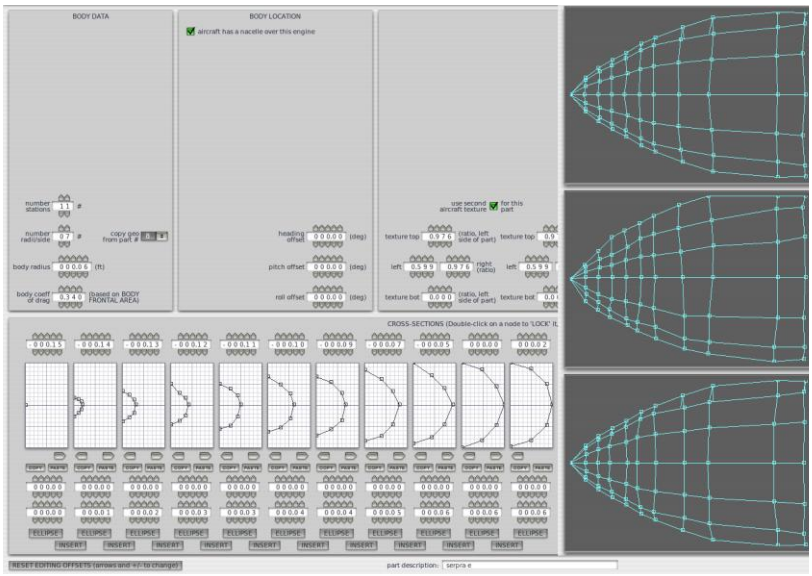
longitudinal, lateral, and vertical object attach arm | heading and pitch offsets (deg)

clear	0 0 0.0 8	0 0 2.3 1	0 0 0.0 7	aircraft reference	0 0 0	0 0 0	Nav Right	□ Hide						
clear	0 0 0.0 8	- 0 0 2.3 1	0 0 0.0 7	aircraft reference	0 0 0	0 0 0	Nav Left	□ Hide						
clear	0 0 0.0 1	- 0 0 1.3 5	0 0 0.0 4	aircraft reference	0 0 0	0 0 0	Landing	□ Hide	# 0 0	length	0 3 0	6 0	width	
										rgb	1.0 0	1.0 0	1.0 0	
clear	0 0 0.0 1	- 0 0 1.3 9	0 0 0.0 4	aircraft reference	0 0 0	0 0 0	Landing	□ Hide	# 0 0	length	0 3 0	6 0	width	
										rgb	1.0 0	1.0 0	1.0 0	
clear	0 0 2.3 9	0 0 0.0 0	0 0 0.2 8	aircraft reference	0 0 0	0 0 0	Nav Tail	□ Hide						

Figure 3.107 Plane Maker settings for modelling external lightings of the FMS 182 model

The locations of all external LED lights were defined based on the measurements in the Exterior Lights section under Systems tab.

3.3.7 Propeller Spinner Modelling


BODY DATA

number stations 1 1 #

number radii/side 0 7 # copy geo from part #

body radius 0 0 0.0 6 (ft)

body coeff of drag 0.3 4 0 (based on BODY FRONTAL AREA)

BODY LOCATION

☑ aircraft has a nacelle over this engine

heading offset 0 0 0.0 0 (deg)

pitch offset 0 0 0.0 0 (deg)

roll offset 0 0 0.0 0 (deg)

use second aircraft texture ☑ for this part

texture top 0.9 7 6 (ratio, left side of part) texture top 0.9

left 0.5 9 9 0.9 7 6 right (ratio) left 0.5 9 9

texture bot 0.0 0 0 (ratio, left side of part) texture bot 0.0

CROSS-SECTIONS (Double-click on a node to 'LOCK' it,

- 0 0 0.1 5	- 0 0 0.1 4	- 0 0 0.1 3	- 0 0 0.1 2	- 0 0 0.1 1	- 0 0 0.1 0	- 0 0 0.0 9	- 0 0 0.0 7	- 0 0 0.0 5	0 0 0.0 0	0 0 0.0 2
0 0 0.0 0	0 0 0.0 0	0 0 0.0 0	0 0 0.0 0	0 0 0.0 0	0 0 0.0 0	0 0 0.0 0	0 0 0.0 0	0 0 0.0 0	0 0 0.0 0	0 0 0.0 0
0 0 0.0 0	0 0 0.0 1	0 0 0.0 2	0 0 0.0 3	0 0 0.0 3	0 0 0.0 4	0 0 0.0 4	0 0 0.0 5	0 0 0.0 6	0 0 0.0 6	0 0 0.0 6

part description: serpra e

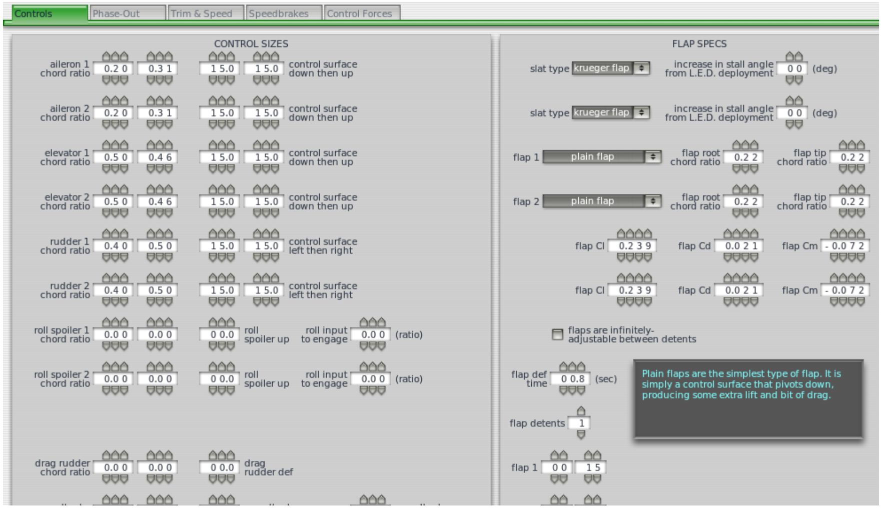
Figure 3.108 Propeller spinner modelling for the FMS 182 in Plane Maker

The same method used for fuselage modelling was utilised to create the spinner and position it appropriately within the modelling space of the aircraft.

3.3.8 Wheel Fairings Modelling

The wheel fairings were also modelled in by following the same method utilised by the fuselage and spinner modelling. As such the details of it have been documented only in the Appendix (Chapter 7.5).

3.3.9 Control geometry modelling



CONTROL SIZES

aileron 1 chord ratio 0.20 0.31 15.0 15.0 control surface down then up
aileron 2 chord ratio 0.20 0.31 15.0 15.0 control surface down then up
elevator 1 chord ratio 0.50 0.46 15.0 15.0 control surface down then up
elevator 2 chord ratio 0.50 0.46 15.0 15.0 control surface down then up
rudder 1 chord ratio 0.40 0.50 15.0 15.0 control surface left then right
rudder 2 chord ratio 0.40 0.50 15.0 15.0 control surface left then right
roll spoiler 1 chord ratio 0.00 0.00 00.0 roll spoiler up roll input to engage 0.00 (ratio)
roll spoiler 2 chord ratio 0.00 0.00 00.0 roll spoiler up roll input to engage 0.00 (ratio)
drag rudder chord ratio 0.00 0.00 00.0 drag rudder def

FLAP SPECS

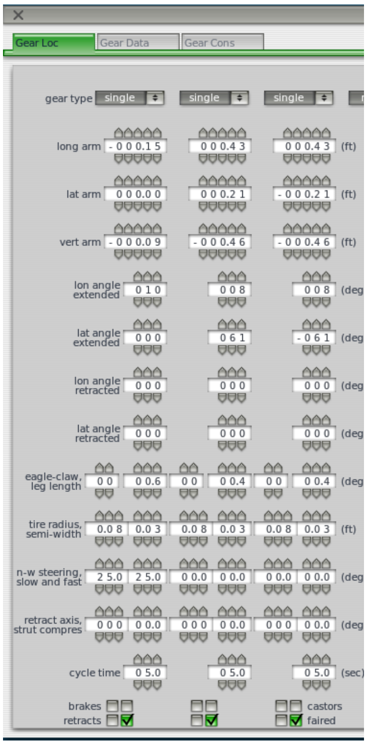
slat type krueger flap increase in stall angle from L.E.D. deployment 00 (deg)
slat type krueger flap increase in stall angle from L.E.D. deployment 00 (deg)
flap 1 plain flap flap root chord ratio 0.22 flap tip chord ratio 0.22
flap 2 plain flap flap root chord ratio 0.22 flap tip chord ratio 0.22
flap Cl 0.239 flap Cd 0.021 flap Cm -0.072
flap Cl 0.239 flap Cd 0.021 flap Cm -0.072
□ flaps are infinitely-adjustable between detents
flap def time 00.8 (sec)
flap detents 1
flap 1 00 15

Plain flaps are the simplest type of flap. It is simply a control surface that pivots down, producing some extra lift and bit of drag.

Figure 3.109 Control systems settings for FMS 182 in Plane Maker

The dimensions of the control surfaces were entered in the form of chord ratios and their max deflections in degrees were assigned. “aileron 1”, “elevator 1” etc. all correspond to the Element Spec section of relevant aerofoils where their locations were defined.

3.3.10 Landing Gear modelling

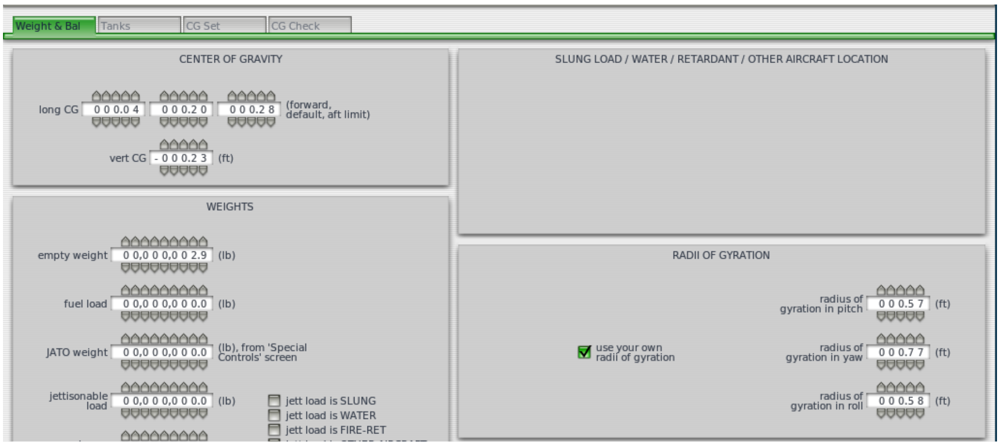


gear type single single single
long arm -000.15 000.43 000.43 (ft)
lat arm 000.00 000.21 -000.21 (ft)
vert arm -000.09 -000.46 -000.46 (ft)
lon angle extended 010 008 008 (deg
lat angle extended 000 061 -061 (deg
lon angle retracted 000 000 000 (deg
lat angle retracted 000 000 000 (deg
eagle-claw, leg length 00 00.6 00 00.4 00 00.4 (deg
tire radius, semi-width 0.08 0.03 0.08 0.03 0.08 0.03 (ft)
n-w steering, slow and fast 25.0 25.0 00.0 00.0 00.0 00.0 (deg
retract axis, strut compres 000 00.0 000 00.0 000 00.0 (deg
cycle time 05.0 05.0 05.0 (sec)
brakes □ □ □ □ □ □ castors
retracts □ ☑ □ ☑ □ ☑ faired

Figure 3.110 Landing gear setting for the FMS 182 in Plane Maker

The landing gear type and location were specified under the Gear Location tab. The boxes for fairing had to be ticked as individual fairings were modelled for these landing gears. All the data entered here are based on measurements of the actual target airplane (converted and sometimes rounded to different units for X-Plane).

3.3.11 Inertial modelling



CENTER OF GRAVITY

long CG 0 0 0.0 4 | 0 0 0.2 0 | 0 0 0.2 8 (forward, default, aft limit)

vert CG - 0 0 0.2 3 (ft)

SLUNG LOAD / WATER / RETARDANT / OTHER AIRCRAFT LOCATION

WEIGHTS

empty weight 0 0,0 0 0,0 0 2.9 (lb)

fuel load 0 0,0 0 0,0 0 0.0 (lb)

JATO weight 0 0,0 0 0,0 0 0.0 (lb), from 'Special Controls' screen

jettisonable load 0 0,0 0 0,0 0 0.0 (lb)

□ jett load is SLUNG
□ jett load is WATER
□ jett load is FIRE-RET

RADII OF GYRATION

☑ use your own radii of gyration

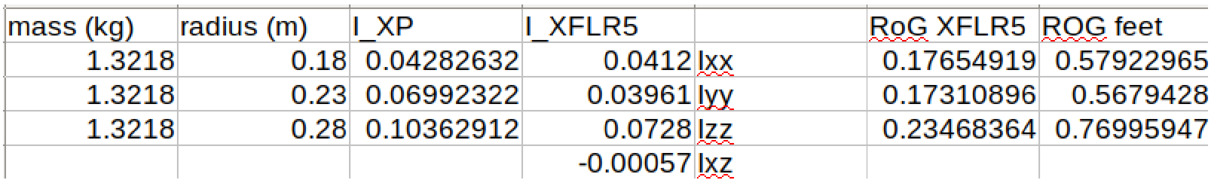
radius of gyration in pitch 0 0 0.5 7 (ft)

radius of gyration in yaw 0 0 0.7 7 (ft)

radius of gyration in roll 0 0 0.5 8 (ft)

Figure 3.111 Inertial settings in Plane Maker for the FMS 182

The method by which X-Plane determines the inertial tendencies of the overall aircraft is based on assuming a uniform density material that is evenly distributed throughout the area/volume of the aircraft. This approximation results in far less realistic inertial responses. As an attempt at improving this, the inertial tensor obtained from XFLR modelling which utilised point masses based on actual measured distances and weights was used.



mass (kg)	radius (m)	I_XP	I_XFLR5		RoG XFLR5	ROG feet
1.3218	0.18	0.04282632	0.0412	Ixx	0.17654919	0.57922965
1.3218	0.23	0.06992322	0.03961	Iyy	0.17310896	0.5679428
1.3218	0.28	0.10362912	0.0728	Izz	0.23468364	0.76995947
			-0.00057	Ixz		

Figure 3.112 Inertial tensor calculations for X-Plane based on XFLR5 prediction

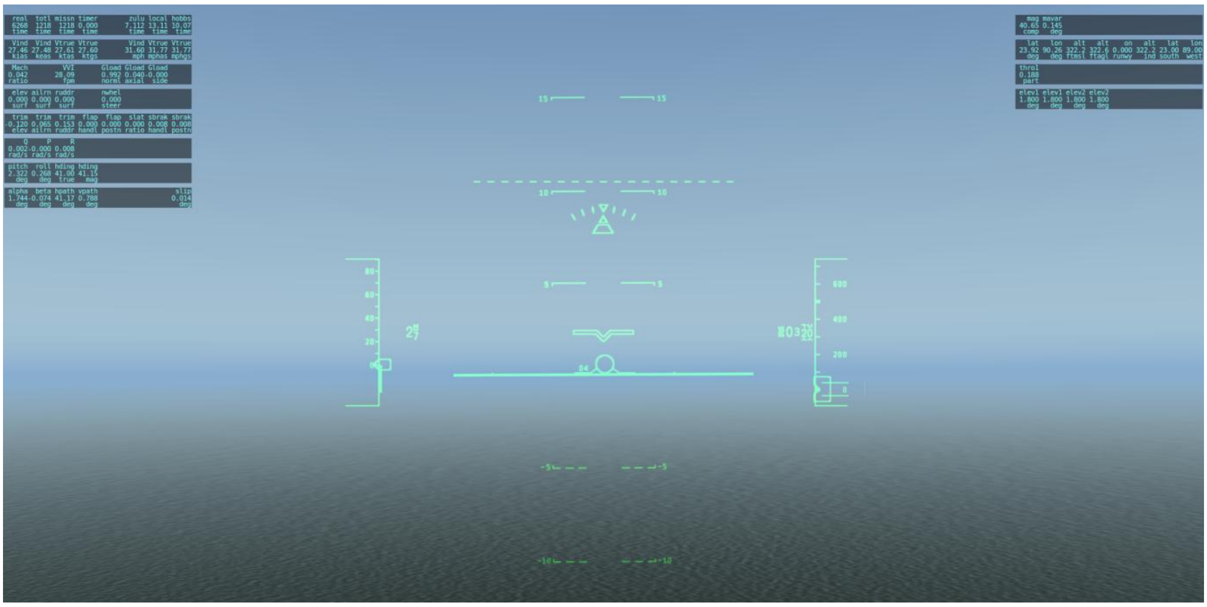
The Centre of Gravity values obtained from XFLR5 was used here along with the Radii of Gyration which required conversion from the Inertial Matrix provided by XFLR5 to the Radii of Gyration (in feet) required by X-Plane. In the above picture, I_XFLR5 is the original inertial tensor obtained from XFLR5. It was first converted from that to the Radii of Gyration matrix (RoG XFLR5) and the units subsequently converted from meters to feet to match that of X-PLANE. It should be noted here that the entry fields in

Plane Maker does not follow the convention of Roll-Pitch-Yaw and instead it requires the values be entered for Pitch-Yaw-Roll.

3.3.12 Texture Modelling

For accurate visual/aesthetic modelling, texture modelling work was carried out and the details can be found in the Appendix (Chapter 7.5).

3.3.13 X-Plane Flight Tests Methods



```
  real  totl missn timer        zulu local hobbs
  6268  1218  1218 0.000       7.112 13.11 10.07
  time  time  time  time        time  time  time

  Vind  Vind Vtrue Vtrue        Vind Vtrue Vtrue
 27.46 27.48 27.61 27.60       31.60 31.77 31.77
  kias  keas  ktas  ktgs         mph mphas mphgs

  Mach         VVI       Gload Gload Gload
 0.042       28.09       0.992 0.040-0.000
 ratio         fpm       norml axial  side

  elev ailrn ruddr       nwhel
 0.000 0.000 0.000       0.000
  surf  surf  surf       steer

  trim  trim  trim  flap  flap  slat sbrak sbrak
-0.120 0.065 0.153 0.000 0.000 0.000 0.008 0.008
  elev ailrn ruddr handl postn ratio handl postn

     Q     P     R
 0.002-0.000 0.008
 rad/s rad/s rad/s

 pitch  roll hding hding
 2.322 0.268 41.00 41.15
   deg   deg  true   mag

 alpha  beta hpath vpath                    slip
 1.744-0.074 41.17 0.788                   0.014
   deg   deg   deg   deg                     deg
```

```
   mag mavar
 40.65 0.145
  comp   deg

   lat   lon   alt   alt    on   alt   lat   lon
 23.92 90.26 322.2 322.6 0.000 322.2 23.00 89.00
   deg   deg ftmsl ftagl runwy   ind south  west

 throl
 0.188
  part

 elev1 elev1 elev2 elev2
 1.800 1.800 1.800 1.800
   deg   deg   deg   deg
```

Figure 3.113 Screenshot from flight test of the FMS 182 conducted in X-Plane

As a demonstration of the advantages and functionality of the finished model, various flights tests were conducted. The following were the test conditions for all these tests:

- Sea level altitude
- Mean sea level atmospheric conditions
- Wind and gust set to zero/minimal.
- Aircraft weight kept constant at its maximum weight.
- The feature "Save Situation" was used to reset the aircraft and environment to identical situations every test.
- The procedure followed for the excitation of individual dynamic modes of the airplane required the aircraft to be in trimmed level and steady flight. This was followed by abrupt control input to the elevator and rudder in separate tests.

Following the flight data extraction and processing methods discussed in the Parameter Identification methods section, attempts at determining stability derivatives from this flight data has been made (see Chapter 4.2.7). The following section details the procedure.

3.3.13.1 Importing of flight data from X-Plane:

Live flight data from X-Plane was imported into MATLAB via the use of a modified JavaScript [see Appendix]. Even though the source JavaScript [10] is technically capable of reading X-Plane's DRef messages from the memory, a series of modification to the code had to carried out to make it work the way it's best suited for the flight data processing method used. By default, the JavaScript reads a few parameters from X-Plane's memory and dumps it onto the terminal (Linux command line utility). While this is useful for live monitoring of the flight data, it needed to be changed for the data processing later on.

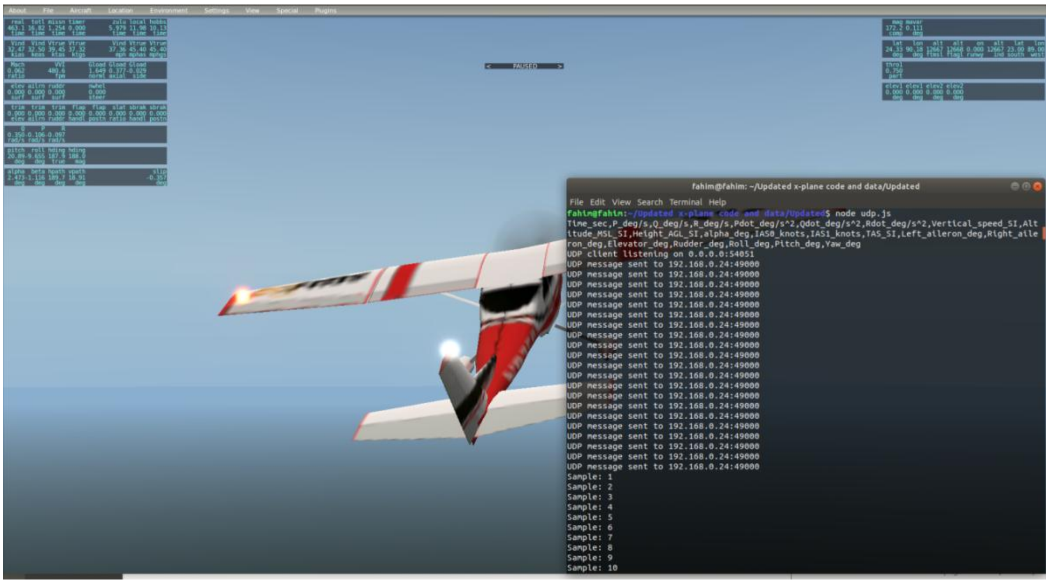
3.3.13.1.1 UDP.js

The JavaScript UDP.js was the end result of the modifications stated in the previous section. The full code is provided in the Appendix (Chapter 7.2). The source code in the Appendix should be consulted to understand the implementation of these modifications. The following modifications and extensions were carried out:

- The IP address changed to match the host IP address.
- The sampling rate was changed from 5 to 100 for maximum data fidelity.
- "Fs" introduced in the code is the file system handler used to write the csv file.
- The data sample counter was introduced (for debugging purpose only). It is used to check live data import using the terminal.
- The original function of the JavaScript was to import the data stream from X-Plane into the terminal. That has been altered such that we now write the flight data into a csv file for later processing and only output the sample number to the screen to let the user know it is working.
- "Generate file name" creates a filename and checks if that file already exists. If it does, it increments the counter (fnameIndex)and checks again. It outputs a csv

file using the fnameIndex along with the fnamePrefix. So the file name follows the convention of fnamePrefix followed by index number and file format (.csv).

- “const drefNames” were used to list all the parameters from X-Plane’s memory that we wish to import. This is where every parameter of interest from X-Plane’s working memory is specified.
- “Concatenate into CSV header” takes all the value in the array and puts them into a comma delimited single line. This is what is used as the headers for all the imported data. Without this, it is difficult to keep track of multiple different parameters and is easy to run into confusion.
- “console.log(nameString)” was added to check if the process was successful.
- “fs.writeFile” creates the actual data file and writes in the imported headers. If the file already exists, it will get overwritten as the flag w was used.
- “const messages” needed to be modified to define every single dref that is needed. “const messages” link to all X-Plane dataref was also included within the code for identification of the particular datarefs for any future use. All the parameters were listed (Flight time, vertical speed etc.)
- Under "Message received", a value counter was added. As was the message string on line 165 including the code to write the actual csv file line by line without erasing the previous data set. For the first entry, there must not be a comma while the subsequent entries need to be delimited using comma. That is done in line 177-178. In line 187, the same method (fs.writeFile) is used with a different flag (a) to append the message string with a new line of data. If this modification isn’t done, the final output flight data csv file will be limited to only one row of flight data (the latest one). The “/n” is used to force it to add a new line and print in the new line. Instead of a continuous line of comma separated values.


fahim@fahim: ~/Updated x-plane code and data/Updated
File Edit View Search Terminal Help

```
fahim@fahim:~/Updated x-plane code and data/Updated$ node udp.js
Time_sec,P_deg/s,Q_deg/s,R_deg/s,Pdot_deg/s^2,Qdot_deg/s^2,Rdot_deg/s^2,Vertical_speed_SI,Alt
itude_MSL_SI,Height_AGL_SI,alpha_deg,IAS0_knots,IAS1_knots,TAS_SI,Left_aileron_deg,Right_aile
ron_deg,Elevator_deg,Rudder_deg,Roll_deg,Pitch_deg,Yaw_deg
UDP client listening on 0.0.0.0:54051
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
UDP message sent to 192.168.0.24:49000
Sample: 1
Sample: 2
Sample: 3
Sample: 4
Sample: 5
Sample: 6
Sample: 7
Sample: 8
Sample: 9
Sample: 10
```

Figure 3.114 Screenshot showing the JavaScript in action with FMS 182 X-Plane simulation running

- Line 189-190 is used for debugging via the sample counter by using terminal outputs. All it does is output the current sample number to terminal (see above image) so that the user knows it has been processed. The CSV file has been specifically formatted for use with MATLAB's "import table" function. This way the built in functionality of MATLAB can be utilised to a greater degree while reducing errors and confusion from mishandling of the dataset in the later stages.

3.3.13.1.2 MATLAB import process of flight data

In order for the flight data import process to work, it is required to navigate to the directory containing the JavaScript and MATLAB functions written for the task. The function "readtable" is utilised to read in the data table. See the corresponding results section for more details. A series of flight tests were conducted and the subsequent dataset were all imported into MATLAB by following the same method.

3.3.13.2 X-Plane Parameter Identification

For the parameter identification, old MATLAB code developed at Cranfield University [11] was used. Although usable, the code itself is very old and has such it was necessary to rewrite and modify it to work better with current/more recent versions of MATLAB. The full set of modified code is made available in the Appendix (Chapter 7.2). As part of the extension to the code, a MATLAB script (runScript.m) was written to make all the discrete pieces of the code to work together for flight data processing. This is the script that controls the entire process instead of the user having to manually go through every individual MATLAB code like it was originally designed. The first 10 lines of runScript.m have the control parameters such as switches for turning on/off plots and calculations, definitions of which dataset to process and what variable to plot for the initial diagnostics. The most important parameter is “cutoffSample”. The flight data always contains an initial idle period where nothing happens. This idle period is used to determine the neutral points and zero values to match the requirements of the PI code [4][see Appendix for details]. The sample number at which the idle period ends and the test begins is determined using the checkPlot switch (line 2 in the screenshot). This is what informs the rest of the code to determine it’s starting point. This initialisation process is manual and important as the rest of the code depends on a current cut-off point for reliable operation. Once that sample number has been determined, the variables checkPlot is set to false, run calc is set to true to enable the actual solver to execute the rest of the calculation.

3.3.13.2.1 LoadUdpData.m

The MATLAB script LoadUdpData.m (see Appendix Chapter 7.2 for full code) was written to aid in the loading process of the flight data into MATLAB’s workspace. This custom function can be called with or without an argument to load all csv files into MATLAB’s workspace. It will either load it from the current directory or the one that’s specified in the input argument. This function creates the dataset structure in memory. This structure contains tables within it which are organised using the file names. This allows for more flexibility and easier handling of large amount of flight data from various sources and simulation runs.

To load the actual raw flight data csv files, loadUdpData calls another custom MATLAB function called udpDumpImport.m. This function was written because the built-in MATLAB function “readtable” shown previously proved to be unreliable at times in the workstations running MATLAB. To avoid errors and various other potential failures, this custom function was written to do the same thing albeit in a less efficient way. The advantage is that this custom code does not rely on the unreliable dependencies. The first action of this function is to import the actual CSV data using the built in “csvread” function. It skips the first line (as that is the header in ascii that csv read cannot handle). The rest of the function processes the header into individual variable names. It creates a blank table and assigns each column of the csv data to a variable name within the table pulled from the header. The next step in the process is to clean up and resample the data to a consistent sampling rate which is compatible with the Parameter Estimation code [11]. This is done by calling the custom function reprocessDataSet.m.

3.3.13.2.2 reprocessDataSet.m

Appendix Chapter 7.2 should be consulted for the full code. The code starts with it’s own control variables. The most important of which being resamplingFrequency and resampleMethod where the user can set the desired output. The method allows a choice of 3 different algorithms. For the available flight test data, the preferred resampling rate was 200 hz and the method used was linear. Oversampling like this prevents aliasing issues during the calculation process further down the line. The first step in the process is to remove duplicate data points using the unique function. Duplicate entries are created when the simulation is paused by the user, but the JavaScript code continues to execute from the simulator. The PI[11] code assumes no duplicates so they need to be removed. Next, simulation time is reset to zero for practicality. The solver is able to cope with a non-zero starting time (which invariably results from later data trimming and the cut-off discussed above), however when selecting the cut-off sample and reviewing plots in general, it is better for the end user that the plots begin at or near t=0.

After resetting the time, data in each set are resampled using MATLAB’s built-in resample function, directed per the control variables. The resampling process introduces additional noise at the beginning and end of the data set (see corresponding results section) due to inherent limitations in the algorithm. Fortunately at 200Hz this results in a very short time period being affected, which can be safely trimmed away. The function returns the resampled and trimmed data for further processing.

Once these functions are called by “runScript.m”, the next section of the “runScript.m” allows the end user to check the data fidelity using plots. It is mainly used to determine the cut-off sample. This is followed by running the actual stability analysis code [11].

The stability analysis code / PI code did not come with a driving script and for that the following script was written.

3.3.13.2.3 ComputeLSS.m

This script was written to work in unison with the modified and extended PI code[11] such that on execution all the necessary computations are carried out automatically and the results organised and stored in MATLAB workspace. See the corresponding results section for the outputs and Chapter 7.2 for the full code.

3.3.13.2.4 ls_sp & ls_long

The parameter estimation MATLAB code[11] ls_sp considers the following state space short period description (Eq 3.2-3.3):

$$\begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} z_w & z_q \\ m_w & m_q \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} z_\eta \\ m_\eta \end{bmatrix} \eta \qquad \text{[Eq. 3.2]}$$

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_u & x_w & x_q & x_u \\ z_u & z_w & z_q & z_\theta \\ m_u & m_w & m_q & m_\theta \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} x_\eta \\ z_\eta \\ m_\eta \\ m_\theta \end{bmatrix} \eta \qquad \text{[Eq. 3.3]}$$

Details on these models are documented in the literature review section. The parameter estimation code were rewritten to work with current versions of MATLAB and adjusted wherever necessary for smooth processing.

3.4 Simulink Methods

The MATLAB/Simulink modelling and simulation approach is based on standard published sets of equations and formulations used for the kinematics and dynamics modelling of fixed-wing aircraft [12-24]. For the actual implementation of these equations in the form of a simulation, Beard & McLain [13] has been followed. This is due to the approach relying on creating custom code and Simulink blocks (Simulink S-Function and MATLAB Function blocks). Beard & McLain [13] presents skeletal template files for their simulation to be constructed and developed by interested parties. However, this research involved a range of modifications to this approach. As a result, these templates were modified appropriately and a unique six-degrees of freedom simulation of the FMS 182 was constructed which benefits from:

- Utilization of the underlying mathematical modelling referred to earlier.
- The customization of coding and utilization of standard Simulink blocks.
- Not requiring the Aerospace Blockset [25].
- Modularity

Based on the previously discussed published literature, the following sections describe the mathematical considerations made for the modules of the constructed simulation.

3.4.1 Modelling Aerofoil Stall

The lift produced by the aerofoil depends on the lift coefficient (C_L) of that aerofoil. In case of the linear model, the lift coefficient (Eq. 3.4) is represented as follows.

$$C_L(\alpha) = C_{L_0} + C_{L_\alpha}\alpha \qquad \text{[Eq. 3.4]}$$

We are able to approximate C_{L_α}as shown in Equation 3.5 [13]:

$$C_{L_\alpha} = \frac{\pi AR}{1 + \sqrt{1 + (AR/2)^2}} \qquad \text{[Eq. 3.5]}$$

This coefficient is not a constant and is directly linked to the angle of attack of the aerofoil (α). While increasing α can increase the lift, this relationship is only true within a certain range of α. Beyond that range, the output from the equations cannot be trusted. This behaviour is modelled by modifying the lift and drag equations (in the stability frame) as shown in Equations 3.6-3.7) [13]:

$$F_{lift} = \frac{1}{2}\rho {V_a}^2 S\left[C_L(\alpha) + C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e\right] \qquad \text{[Eq. 3.6]}$$

$$F_{drag} = \frac{1}{2}\rho {V_a}^2 S\left[C_D(\alpha) + C_{D_q}\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e\right] \qquad \text{[Eq.3.7]}$$

To include the effects of stall on C_L, the method of blending the linear term (Eq. 3.8) with a flat-plate term [13] was followed.

$$C_{L(flat\,plate)} = 2sign(\alpha)\sin^2\alpha\cos\alpha \qquad \text{[Eq.3.8]}$$

The realism was captured by the use of the following equation from Beard and McLain [13], which is used throughout the simulation code (e.g. See next section).

$$C_L(\alpha) = \left(1 - \sigma(\alpha)\right)\left[C_{L_0} + C_{L_\alpha}\alpha\right] + \sigma(\alpha)[2sign(\alpha)\sin^2\alpha\cos\alpha] \qquad \text{[Eq. 3.8]}$$

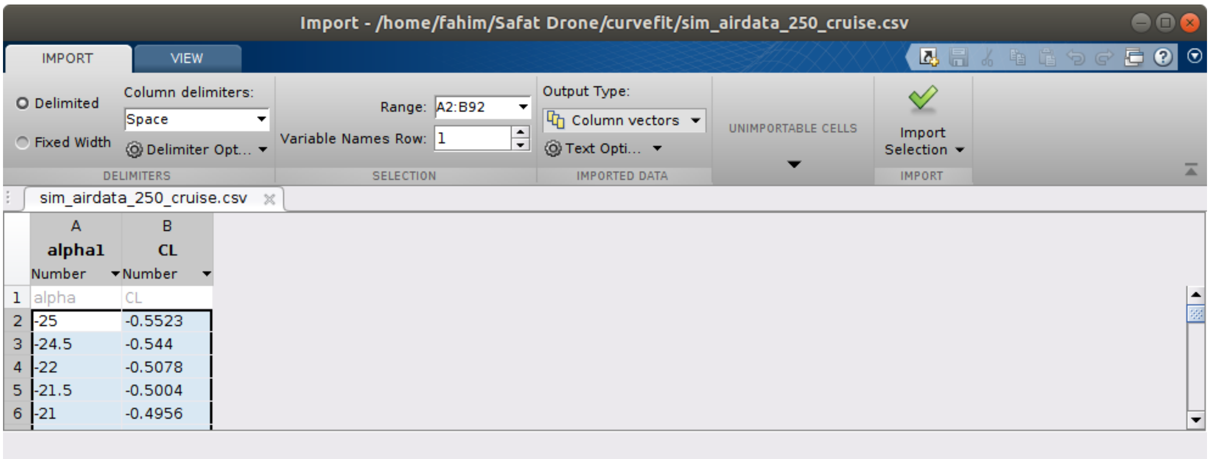
The sigmoid function in Equation 3.8 has the cut-off point at ±α and a positive and constant transition rate (M) and can be written as Equation 3.9 [13]:

$$\sigma(\alpha) = \frac{1 + e^{-M(\alpha-\alpha_0)} + e^{M(\alpha+\alpha_0)}}{\left(1 + e^{-M(\alpha-\alpha_0)}\right)\left(1 + e^{M(\alpha+\alpha_0)}\right)} \qquad \text{[Eq. 3.9]}$$

Both methods have been tested during the construction process of the Simulink simulation and the non-linear one was used and even developed to include additional advantages as described in the next subsection. The MATLAB code written to test the output from these models have been documented in the Appendix (Chapter 7.3).

3.4.1.1 Extension of the sigmoid function utility

As shown previously, the use of the sigmoid function greatly enhances the non-linear modelling of the aerodynamic behaviour of the modelled aircraft. However, it can be difficult to adjust the function variables such that it accurately reflects the aerodynamic characteristics of the actual platform being modelled. To improve this feature, the available aerodynamic data from XFLR5 has been used along with MATLAB to develop a curve-fitting method which is capable of handling XFLR5 data and processing it to output the appropriate values for the individual variables that define the sigmoid function and therefore the aerodynamic behaviour of the aircraft. This is accomplished by a set of MATLAB code comprising of two custom functions, one script and a .sfit file.



Import - /home/fahim/Safat Drone/curvefit/sim_airdata_250_cruise.csv

● Delimited

○ Fixed Width

Column delimiters: Space

Delimiter Opt...

Range: A2:B92

Variable Names Row: 1

Output Type: Column vectors

Text Opti...

sim_airdata_250_cruise.csv

	A	B
	alpha1	CL
	Number	Number
1	alpha	CL
2	-25	-0.5523
3	-24.5	-0.544
4	-22	-0.5078
5	-21.5	-0.5004
6	-21	-0.4956

Figure 3.115 Loading data from XFLR5 on to MATLAB as Column Vectors

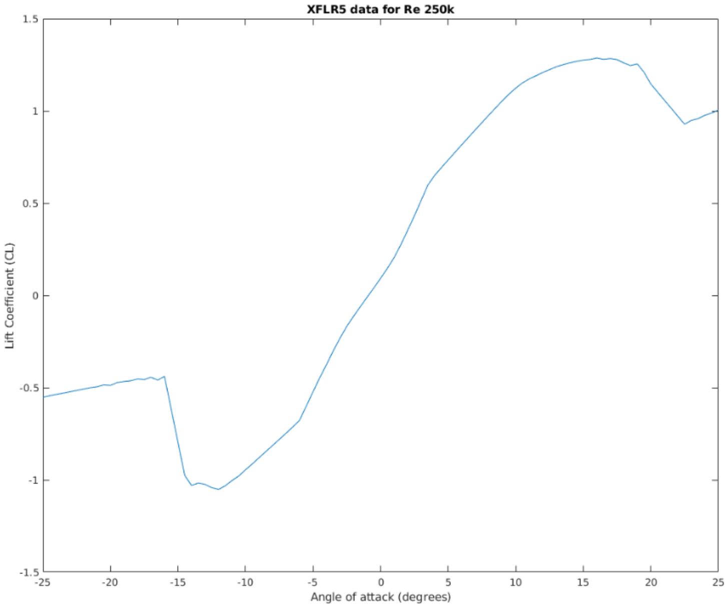
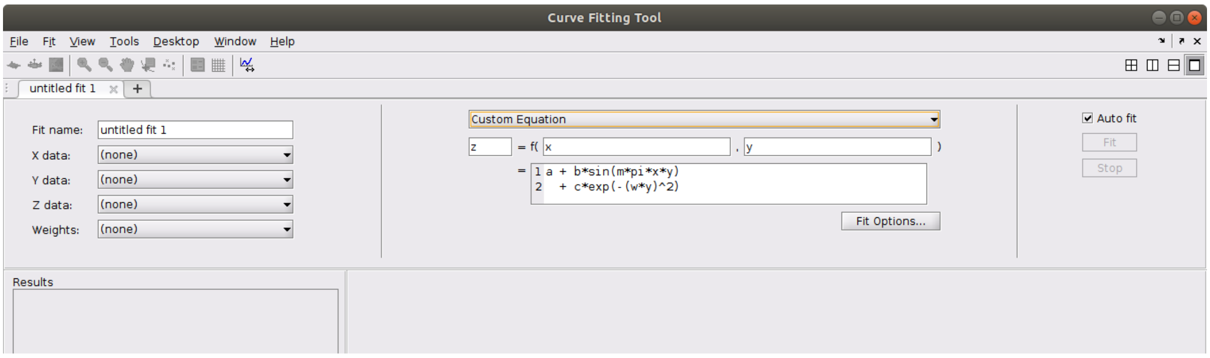


Figure 3.116 XFLR5 raw data plotted in MATLAB

The process starts with loading the available XFLR5 data from the csv file in the form of column vectors on to MATLAB workspace. The output type in the import process must be Column vectors for the rest of the process to work. The graph above depicts the raw data from XFLR5. Once the data is in the workspace through the generatecurve.m script, the values for the angle of attack are converted to radians, aspect ratio is set and CL is calculated based on that. This is followed by calling the next custom function "fitSigmaAlpha.m".



Fit name: untitled fit 1
X data: (none)
Y data: (none)
Z data: (none)
Weights: (none)

Custom Equation

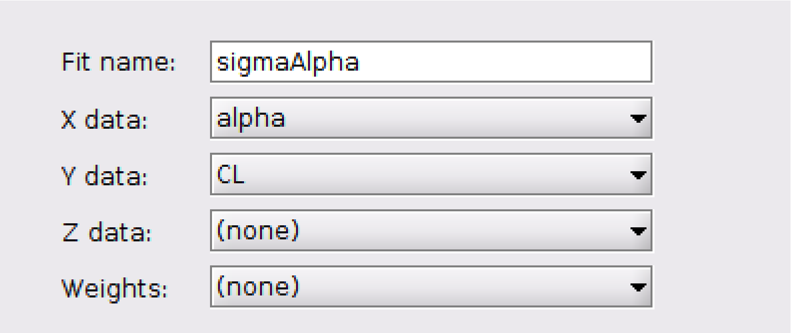
z = f(x , y)

```
= a + b*sin(m*pi*x*y)
    + c*exp(-(w*y)^2)
```

☑ Auto fit

Figure 3.117 Custom equations in curve fitting

The curve fitting tool in MATLAB automatically generates a function of this type but it needed to be modified. As the initialising setting, the variables alpha and CL were provide to the curve fitting tool as X and Y data respectively.



Fit name: sigmaAlpha

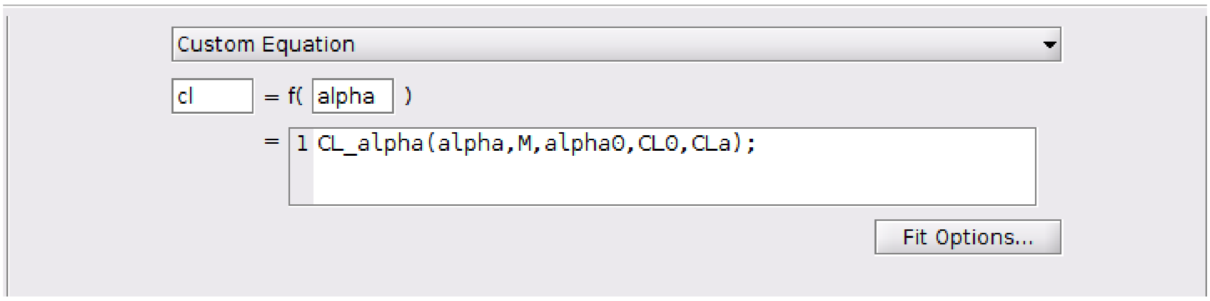
X data: alpha

Y data: CL

Z data: (none)

Weights: (none)

Figure 3.118 Settings for curve fitting



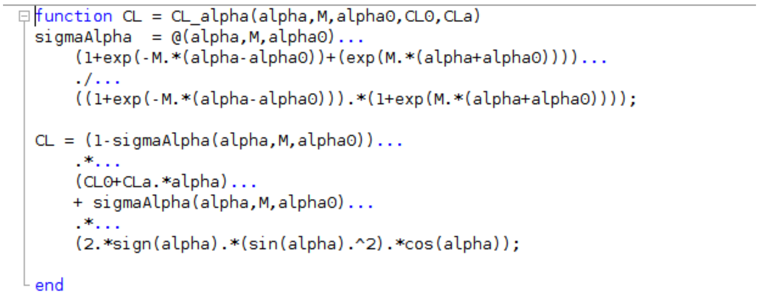
Custom Equation

cl = f(alpha)

= CL_alpha(alpha,M,alpha0,CL0,CLa);

Figure 3.119 Custom function

The custom function CL_alpha.m outputs all the necessary CL values for this process.



```
function CL = CL_alpha(alpha,M,alpha0,CL0,CLa)
sigmaAlpha  = @(alpha,M,alpha0)...
    (1+exp(-M.*(alpha-alpha0))+(exp(M.*(alpha+alpha0))))...
    ./...
    ((1+exp(-M.*(alpha-alpha0))).*(1+exp(M.*(alpha+alpha0))));

CL = (1-sigmaAlpha(alpha,M,alpha0))...
    .*...
    (CL0+CLa.*alpha)...
    + sigmaAlpha(alpha,M,alpha0)...
    .*...
    (2.*sign(alpha).*(sin(alpha).^2).*cos(alpha));

end
```

Figure 3.120 CL_alpha.m function for sigmoid function

As seen in the above screenshot, this is done by transcribing the previously detailed sigmoid function equations into MATLAB code such that relevant sets of values for CL can be computed. This function is called by the "fitSigmaAlpha.m" to identify the best fit. Conditional statements were included in to the function to detect potential mismatch of units of alpha and automatically correct them (the rest of the code base works with radians). The custom function "CL_alpha.m" was also incorporated into fitSigmaAlpha.m in the form of a nested function such that it becomes an all-in-one function for this job. The results are documented in the corresponding results section.

3.4.2 Modelling of drag

A similar approach has been taken to model the drag acting on the FMS 182. Two types of drags are considered, parasitic and induced. Within the expected operational envelope of the FMS 182 (small α), the induced drag acting on it can be shown to be proportional to the square of the lift force acting on it [13]. The parasitic drag, which results from the sheer stress acting on the aerofoil due to the airflow over it, is however taken to be a constant. Following the convention in the literature [13], we are already using $C_{D_0}$to represent the drag coefficient when α = 0, therefore we don't use the same label for the parasitic drag coefficient and instead label that C_{D_p}. Therefore the equation for the drag coefficient (Eq. 3.10) is modelled as follows.

$$C_D(\alpha) = C_{D_0} + C_{D_\alpha}\alpha \qquad \text{[Eq. 3.10]}$$

The above model was used for testing linear approximations and the equation (Eq. 3.11) below was used for the non-linear approximation (e = Oswald's efficiency factor [13]).

$$C_D(\alpha) = C_{D_p} + \frac{\left(C_{L_0} + C_{L_\alpha}\alpha\right)^2}{\pi e AR} \qquad \text{[Eq. 3.11]}$$

3.4.3 Modelling of Cm

In the absence of wind tunnel testing or data obtained from actual flight tests, the pitching moment acting on the FMS 182 was approximated (Eq. 3.12) with the following linear model [13]:

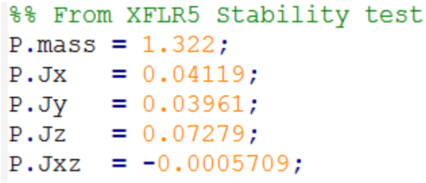
$$C_m(\alpha) = C_{m_0} + C_{m_\alpha}\alpha \qquad \text{[Eq. 3.12]}$$

The variables were extracted from XFLR5 stability test results and the linear model is presented in the results section.

3.4.4 Simulation Initialisation

The developed Simulink model of the FMS 182 relies on an initialisation process which loads all important constant parameters for the specific aircraft. As the separate modules of the simulation begin to execute their functionality, they use these parameters as inputs for all their relevant calculations. This model initialisation function is the “nParam.m” file. For this file, the same convention from the template was used. All parameters are recorded in the format of P.x where x is the parameter.

For example, in this convention the constant value of 9.81 meters per second squared is stored as P.g = 9.81.



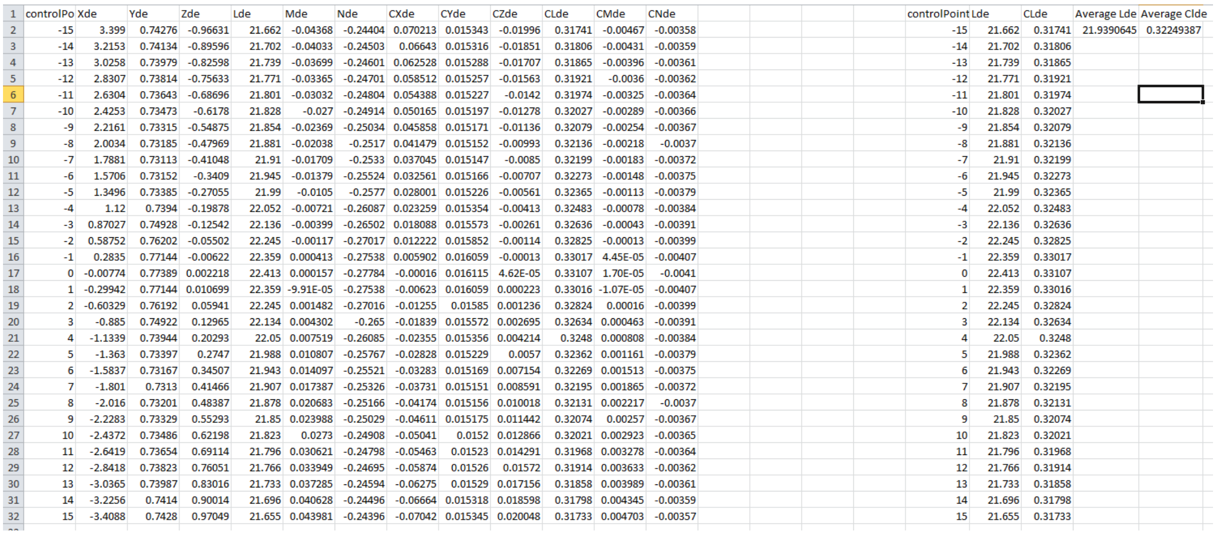
```
%% From XFLR5 Stability test
P.mass = 1.322;
P.Jx   = 0.04119;
P.Jy   = 0.03961;
P.Jz   = 0.07279;
P.Jxz  = -0.0005709;
```

Figure 3.121 Inertial tensor from the XFLR5 estimations

This method was used to manually import the data generated by XFLR5 and X-Plane simulations into Simulink to create a unique parameters file that is usable with this simulation. The above screenshot is an example of convention mismatch which complicates the process since XFLR5 uses the “I” matrix instead of “J” and X-Plane uses Radii of Gyration). However, all such variable naming conventions were carefully noted and examined to create the nParam file whose contents are made available in the results section.

3.4.5 Stability & Control derivatives data

Many of the key stability and control derivatives used for the simulation were obtained from the XFLR5 and X-Plane flight data.



1	controlPo	Xde	Yde	Zde	Lde	Mde	Nde	CXde	CYde	CZde	CLde	CMde	CNde
2	-15	3.399	0.74276	-0.96631	21.662	-0.04368	-0.24404	0.070213	0.015343	-0.01996	0.31741	-0.00467	-0.00358
3	-14	3.2153	0.74134	-0.89596	21.702	-0.04033	-0.24503	0.06643	0.015316	-0.01851	0.31806	-0.00431	-0.00359
4	-13	3.0258	0.73979	-0.82598	21.739	-0.03699	-0.24601	0.062528	0.015288	-0.01707	0.31865	-0.00396	-0.00361
5	-12	2.8307	0.73814	-0.75633	21.771	-0.03365	-0.24701	0.058512	0.015257	-0.01563	0.31921	-0.0036	-0.00362
6	-11	2.6304	0.73643	-0.68696	21.801	-0.03032	-0.24804	0.054388	0.015227	-0.0142	0.31974	-0.00325	-0.00364
7	-10	2.4253	0.73473	-0.6178	21.828	-0.027	-0.24914	0.050165	0.015197	-0.01278	0.32027	-0.00289	-0.00366
8	-9	2.2161	0.73315	-0.54875	21.854	-0.02369	-0.25034	0.045858	0.015171	-0.01136	0.32079	-0.00254	-0.00367
9	-8	2.0034	0.73185	-0.47969	21.881	-0.02038	-0.2517	0.041479	0.015152	-0.00993	0.32136	-0.00218	-0.0037
10	-7	1.7881	0.73113	-0.41048	21.91	-0.01709	-0.2533	0.037045	0.015147	-0.0085	0.32199	-0.00183	-0.00372
11	-6	1.5706	0.73152	-0.3409	21.945	-0.01379	-0.25524	0.032561	0.015166	-0.00707	0.32273	-0.00148	-0.00375
12	-5	1.3496	0.73385	-0.27055	21.99	-0.0105	-0.2577	0.028001	0.015226	-0.00561	0.32365	-0.00113	-0.00379
13	-4	1.12	0.7394	-0.19878	22.052	-0.00721	-0.26087	0.023259	0.015354	-0.00413	0.32483	-0.00078	-0.00384
14	-3	0.87027	0.74928	-0.12542	22.136	-0.00399	-0.26502	0.018088	0.015573	-0.00261	0.32636	-0.00043	-0.00391
15	-2	0.58752	0.76202	-0.05502	22.245	-0.00117	-0.27017	0.012222	0.015852	-0.00114	0.32825	-0.00013	-0.00399
16	-1	0.2835	0.77144	-0.00622	22.359	0.000413	-0.27538	0.005902	0.016059	-0.00013	0.33017	4.45E-05	-0.00407
17	0	-0.00774	0.77389	0.002218	22.413	0.000157	-0.27784	-0.00016	0.016115	4.62E-05	0.33107	1.70E-05	-0.0041
18	1	-0.29942	0.77144	0.010699	22.359	-9.91E-05	-0.27538	-0.00623	0.016059	0.000223	0.33016	-1.07E-05	-0.00407
19	2	-0.60329	0.76192	0.05941	22.245	0.001482	-0.27016	-0.01255	0.01585	0.001236	0.32824	0.00016	-0.00399
20	3	-0.885	0.74922	0.12965	22.134	0.004302	-0.265	-0.01839	0.015572	0.002695	0.32634	0.000463	-0.00391
21	4	-1.1339	0.73944	0.20293	22.05	0.007519	-0.26085	-0.02355	0.015356	0.004214	0.3248	0.000808	-0.00384
22	5	-1.363	0.73397	0.2747	21.988	0.010807	-0.25767	-0.02828	0.015229	0.0057	0.32362	0.001161	-0.00379
23	6	-1.5837	0.73167	0.34507	21.943	0.014097	-0.25521	-0.03283	0.015169	0.007154	0.32269	0.001513	-0.00375
24	7	-1.801	0.7313	0.41466	21.907	0.017387	-0.25326	-0.03731	0.015151	0.008591	0.32195	0.001865	-0.00372
25	8	-2.016	0.73201	0.48387	21.878	0.020683	-0.25166	-0.04174	0.015156	0.010018	0.32131	0.002217	-0.0037
26	9	-2.2283	0.73329	0.55293	21.85	0.023988	-0.25029	-0.04611	0.015175	0.011442	0.32074	0.00257	-0.00367
27	10	-2.4372	0.73486	0.62198	21.823	0.0273	-0.24908	-0.05041	0.0152	0.012866	0.32021	0.002923	-0.00365
28	11	-2.6419	0.73654	0.69114	21.796	0.030621	-0.24798	-0.05463	0.01523	0.014291	0.31968	0.003278	-0.00364
29	12	-2.8418	0.73823	0.76051	21.766	0.033949	-0.24695	-0.05874	0.01526	0.01572	0.31914	0.003633	-0.00362
30	13	-3.0365	0.73987	0.83016	21.733	0.037285	-0.24594	-0.06275	0.01529	0.017156	0.31858	0.003989	-0.00361
31	14	-3.2256	0.7414	0.90014	21.696	0.040628	-0.24496	-0.06664	0.015318	0.018598	0.31798	0.004345	-0.00359
32	15	-3.4088	0.7428	0.97049	21.655	0.043981	-0.24396	-0.07042	0.015345	0.020048	0.31733	0.004703	-0.00357

controlPoint	Lde	CLde	Average Lde	Average Clde
-15	21.662	0.31741	21.9390645	0.32249387
-14	21.702	0.31806		
-13	21.739	0.31865		
-12	21.771	0.31921		
-11	21.801	0.31974		
-10	21.828	0.32027		
-9	21.854	0.32079		
-8	21.881	0.32136		
-7	21.91	0.32199		
-6	21.945	0.32273		
-5	21.99	0.32365		
-4	22.052	0.32483		
-3	22.136	0.32636		
-2	22.245	0.32825		
-1	22.359	0.33017		
0	22.413	0.33107		
1	22.359	0.33016		
2	22.245	0.32824		
3	22.134	0.32634		
4	22.05	0.3248		
5	21.988	0.32362		
6	21.943	0.32269		
7	21.907	0.32195		
8	21.878	0.32131		
9	21.85	0.32074		
10	21.823	0.32021		
11	21.796	0.31968		
12	21.766	0.31914		
13	21.733	0.31858		
14	21.696	0.31798		
15	21.655	0.31733		

Figure 3.122 XFLR5 dataset from control tests being processed

The general method used was to import the XFLR5 and X-Plane data onto a spread sheet and isolate the values of interest across the range of control points (deflection angle of the control surface). The sources of these data are as follows:

- T2/T7-VLM2 test results from XFLR5 (FMS 182b Elevator Up)
- T2/T7-VLM2 test results from XFLR5 (FMS 182b Elevator Down)
- T7-VLM2 test results from XFLR5 (FMS 182b Rudder Up)
- T7-VLM2 test results from XFLR5 (FMS 182b Rudder Down)
- T7-VLM2 test results from XFLR5 (FMS 182b Aileron Up)
- T7-VLM2 test results from XFLR5 (FMS 182b Aileron down)
- Processed flight data from X-Plane flight tests.

The control derivatives for the elevator responses had to be estimated from T2 tests (at 6 degrees alpha)as the other XFLR5 tests consider a zero pitching moment situation which interferes with the process of determining the derivatives based on elevator input. Unfortunately the method requires manipulation of huge amount of datasets. As such, the key ones are being included in the Appendix (Chapter 7.4) and the rest of the bulk of the dataset produced will be uploaded online and linked within the Appendix.

3.4.6 Visualisation Methods

Simulations of fixed-wing UAVs can benefit from the addition of visual outputs to make the results more human-friendly. While it is entirely possible to construct simulations where the only outputs are either numeric values or those values plotted on graphs. While there is nothing innately wrong with that simplicity, the lack of any visual simulation of the aircraft can result in both added diagnostic difficulties and much needed sanity checks when errors or bugs produce erroneous outputs.

For the purpose of constructing the MATLAB/Simulink simulation of the FMS 182, an interpreted MATLAB function (drawVehicle.m) has been utilised. This can be obtained from the skeletal/template files [see Chapter 7.3]. This function, when populated with appropriate code can form the basis of the visual simulation for the aircraft. In every cycle, it updates the 3 dimensional graphing space with the present orientation of the vehicle. The 6 parameters of interest to represent the aircraft in 3D space are as follows:

- Inertial North Position
- Inertial East Position
- Inertial Down Position

- Roll Angle
- Pitch Angle
- Yay Angle

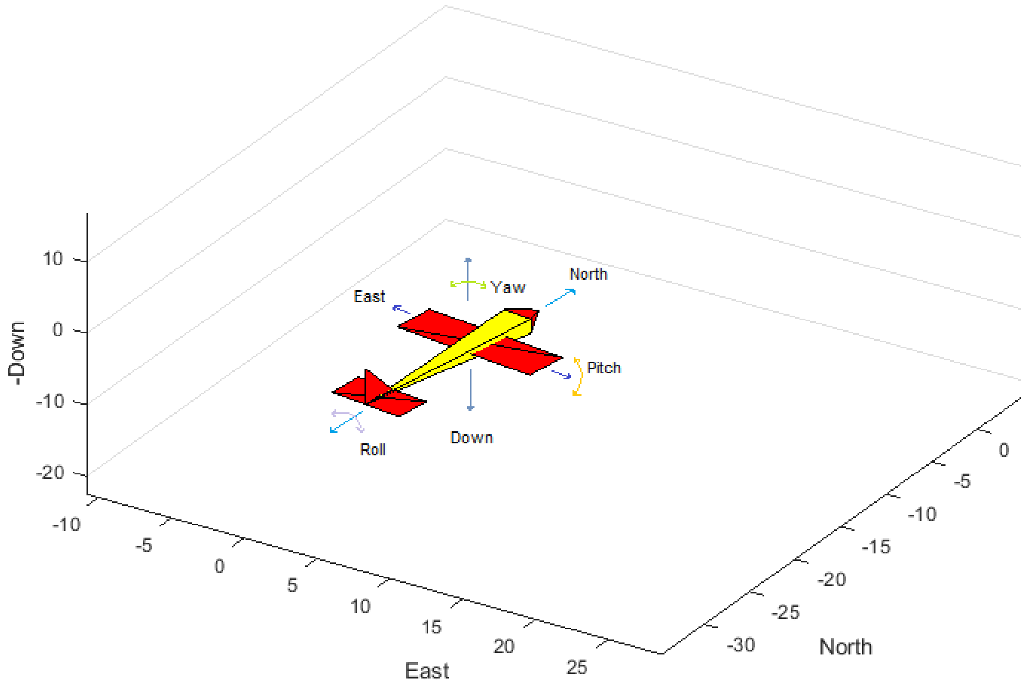


Figure 3.123 A simple visualisation based on the vertices and faces method

The first three parameters are the 3 translational degrees of freedom (surge, sway & heave) and the last three parameters are the 3 rotational degrees of freedom (roll, pitch & yaw). In the image, North, East & Down corresponds to Surge, Sway and Heave. Combined, they go on to describe and capture the motion of the aircraft in all six degree-of-freedom (6DOF) [27]. The appropriate values for these 6 parameters are computed by the kinematics and rigid body dynamics [13][28-30] worked out by the module of the simulation responsible for the dynamics calculations. As such, we may provide them as inputs to the “drawVehicle” function for it to render the current orientation of the vehicle.

The way the basic template file attempts to do it by defining the aircraft in terms of vertices and faces. This principle has been exploited to allow for much more complex geometric designs than the type in shown in the above image [Figure 3.123].

3.4.6.1 Vertices & Faces Method

The patch command [31] in MATLAB allows for the construction of 2D surfaces that can be put together to generate any 3D object. This is done by expressing the target object as a combined set of vertices and faces. Every vertex is a point in 3D space that determines its location and every face is a polygon created from these vertices. With these points and defined surfaces, MATLAB allows the rendering of 3D objects [31]. The following example of a pyramid shape can be considered as a demonstration of this technique.

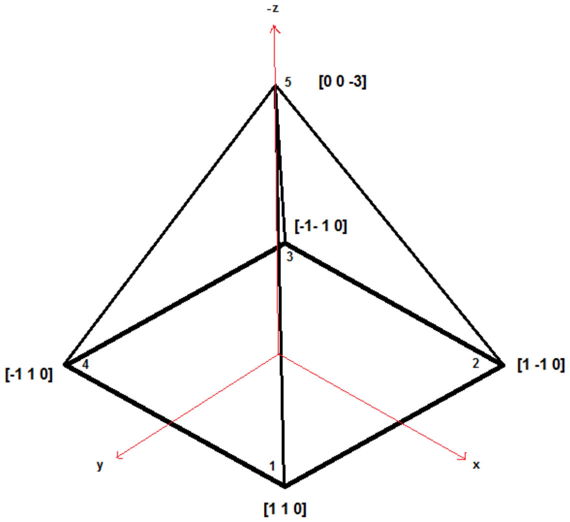


Figure 3.124 Example pyramid in 3D Cartesian space

It can be seen that the shape has 5 points of intersection of lines with the coordinates stated beside them (in the [x,y,z] format). The method starts by defining these vertices and faces in MATLAB as shown in Appendix (Chapter 7.3).To aid with the visualisation, the surfaces can be coloured following MATLAB’s predefined conventions [31] and convert the colour information into a matrix (FaceVertexCData) as follows. This specific code (Chapter 7.3) has some redundancy due to the fact that MATLAB has 7 predefined colours that can be used without defining the RGB triplets [31]. But the advantage of doing it this way is that it illustrates how more complex shadings can be arrived at by supplying the FaceVertexCData with a matrix containing the colour information. Once this is done, we can put together all the defined surfaces as follows. The matrices V and F supply it with the necessary information for the generation of the surfaces that are to be put together the matrix “colours” provide the information on the colour of individual surfaces. The resulting output is a 3D shape resembling the original drawing.

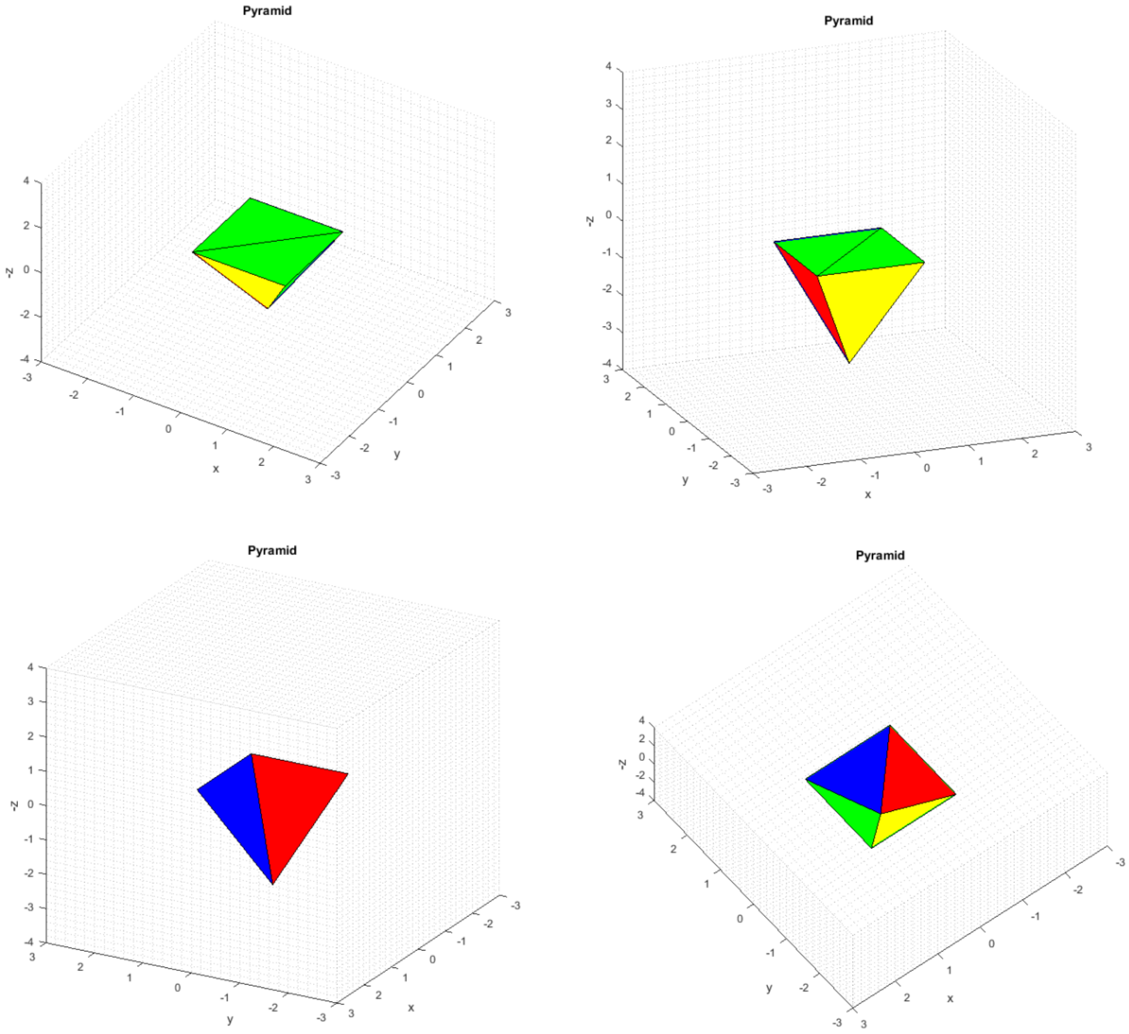


Figure 3.125 Rendering of the 3D pyramid shape

This principle has been used to import complex aircraft geometry right into the simulation thereby enhancing its capabilities.

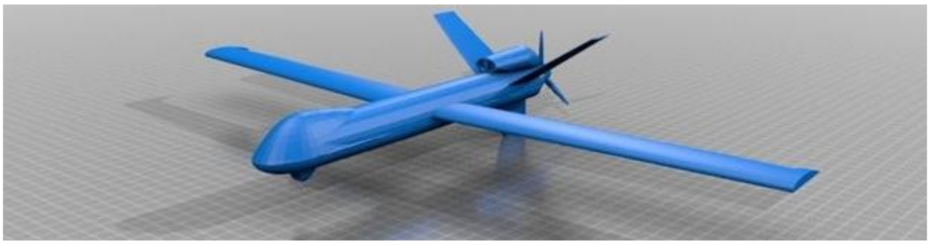
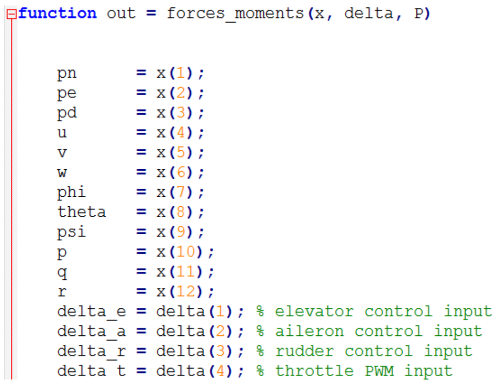


Figure 3.126 MQ-9 Reaper 3D model [32]

There are freely available CAD models [32] that were utilised for this purpose. The results are shown in the corresponding results section. These models generally are provided in STL files file format [33] which form the basis of the geometric data needed for the visual rendering in simulation space. For testing the limits of such rendering, a CAD model of the MQ-9 Reaper [32] (pictured above) was obtained and utilised. The function stlread [34] was used to decompose the STL file into the vertexes and faces (V & F) matrices. Once these matrices are available, the exact same process of using the patch function can be used to generate the 3D visual representation of this aircraft within the 3D plot as shown in the corresponding results section (Chapter 4.3.5) For general visualisation of the simulation during tests, this visualisation method is handled by the template file drawVehicle.m. Modifications were needed to make it more modular and accommodate the type of complex visualisation stated here. A standalone test model including control input and kinematics block has been uploaded online and the web link shared in the Appendix.

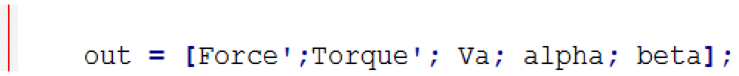
3.4.7 Modelling of Forces & Moments

The modelling of all the forces and moments acting on the FMS 182 has been done via the use of interpreted MATLAB function blocks in Simulink [36]. The block intakes the input from the initialisation function discussed earlier, the control inputs from Simulink (elevator, rudder, aileron and throttle) and the state matrix x.



```
function out = forces_moments(x, delta, P)


    pn      = x(1);
    pe      = x(2);
    pd      = x(3);
    u       = x(4);
    v       = x(5);
    w       = x(6);
    phi     = x(7);
    theta   = x(8);
    psi     = x(9);
    p       = x(10);
    q       = x(11);
    r       = x(12);
    delta_e = delta(1); % elevator control input
    delta_a = delta(2); % aileron control input
    delta_r = delta(3); % rudder control input
    delta t = delta(4); % throttle PWM input
```

Figure 3.127 Input to the forces_moments function



```
    out = [Force';Torque'; Va; alpha; beta];
```

Figure 3.128 Output from the forces_moments function

As shown in Figure 3.127-3.128, the imported inputs from the structure P are assigned to the state matrix and the control inputs (eg. Delta_e) are assigned to delta(). A serious of calculations are performed in this block to finally produce the output for the Force, Moments, Airspeed, angle of attack and side slip angle. All the mathematical models used here are taken from literature (see Literature Review). The equations used for the calculations are as follows.

Longitudinal forces and moments were modelled based on the following sets of equations (Eq. 3.13) [13]:

$$\begin{bmatrix} f_x \\ f_z \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} -F_{drag} \\ -F_{lift} \end{bmatrix}$$

$$= \frac{1}{2}\rho {V_a}^2 S \begin{bmatrix} [-C_D(\alpha)\cos\alpha + C_L(\alpha)\sin\alpha] + \left[-C_{D_q}(\alpha)\cos\alpha + C_{L_q}(\alpha)\sin\alpha\right]\frac{c}{2V_a}q + \left[-C_{D_{\delta_e}}(\alpha)\cos\alpha + C_{L_{\delta_e}}(\alpha)\sin\alpha\right]\delta_e \\ [-C_D(\alpha)\cos\alpha - C_L(\alpha)\sin\alpha] + \left[-C_{D_q}(\alpha)\cos\alpha - C_{L_q}(\alpha)\sin\alpha\right]\frac{c}{2V_a}q + \left[-C_{D_{\delta_e}}(\alpha)\cos\alpha - C_{L_{\delta_e}}(\alpha)\sin\alpha\right]\delta_e \end{bmatrix}$$

[Eq. 3.13]

Lateral forces and moments were modelled based on the following sets of equations (Eq. 3.14-3.16)[13]:

$$f_y = \frac{1}{2}\rho {V_a}^2 S\left[C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a + C_{Y_r}\delta_r\right] \quad \text{[Eq. 3.14]}$$

$$l = \frac{1}{2}\rho {V_a}^2 Sb\left[C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_r}\delta_r\right] \quad \text{[Eq. 3.15]}$$

$$n = \frac{1}{2}\rho {V_a}^2 Sb\left[C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a + C_{n_r}\delta_r\right] \quad \text{[Eq. 3.16]}$$

The propulsive forces were modelled based on Equation 3.17 [13]:

$$f_p = \frac{1}{2}\rho S_{prop} C_{prop} \begin{bmatrix} [k_{motor}\delta_t]^2 - {V_a}^2 \\ 0 \\ 0 \end{bmatrix} \quad \text{[Eq. 3.17]}$$

The moment caused by the spinning propeller was modelled based on the following equations (Eq. 3.18-3.19) [13]:

$$T_p = -k_{T_p}(k_\Omega \delta_t)^2 \quad \text{[Eq. 3.18]}$$

$$m_p = \begin{bmatrix} -k_{T_p}(k_\Omega \delta_t)^2 \\ 0 \\ 0 \end{bmatrix} \quad \text{[Eq. 3.19]}$$

The force of gravity (in body frame) was modelled based on Equation 3.20 [13]:

$$f_g^b = \begin{bmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{bmatrix} \quad \text{[Eq. 3.20]}$$

Summation of all forces acting on the FMS 182 was done by separately calculating these matrices within the interpreted MATLAB function and summing them up.

As shown in the relevant MATLAB code (Chapter 7.3), these equations have to be transcribed to MATLAB conventions following all the correct parameter referencing. Because these equations are rather large and can be cumbersome to manage while debugging, they are split into pieces that are calculated separately but they are all summed together in the end to pass to the block's output. The exact same method has been used for the moments calculations based on the stated equations. The airspeed vector (in all three axis) has been modelled based on the Equation 3.21 [13]:

$$V_a^b = \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = \begin{bmatrix} u - u_w \\ v - v_w \\ w - w_w \end{bmatrix} \qquad \text{[Eq. 3.21]}$$

The magnitudes of V_a, α and β have been derived from it and the equations (Eq. 3.22-3.24) used are as follows [13]:

$$V_a = \sqrt{{u_r}^2 + {v_r}^2 + {w_r}^2} \qquad \text{[Eq. 3.22]}$$

$$\alpha = \tan^{-1}\left[\frac{w_r}{u_r}\right] \qquad \text{[Eq. 3.23]}$$

$$\beta = \sin^{-1}\left[\frac{v_r}{\sqrt{{u_r}^2 + {v_r}^2 + {w_r}^2}}\right] \qquad \text{[Eq. 3.24]}$$

The same process of transcribing them into MATLAB format and passing the final results to the output signal has been followed for them.

3.4.8 Dynamics & Kinematics block

At any given moment during the operational flight envelope of the FMS 182, there is key information about certain states that we wish to know. As such, based on Beard and McLain [13], the following summary is provided.

State	Symbol	Description	Symbol (Derivatives)
Position (North)	p_n	Inertial North position in the inertial frame	$\dot{p}_n$
Position (East)	p_e	Inertial East position in the inertial frame	$\dot{p}_e$
Position (Down)	p_d	Inertial down (-altitude) position in the inertial frame	$\dot{p}_d$
Linear velocity (u)	u	Linear velocity component measured in the i-axis of the body frame.	$\dot{u}$
Linear velocity (v)	v	Linear velocity component measured in the j-axis of the body frame.	$\dot{v}$
Linear velocity (w)	w	Linear velocity component measured in the k-axis of the body frame.	$\dot{w}$
Roll	ϕ	Roll angle (vehicle-2 frame)	$\dot{\phi}$
Pitch	θ	Pitch angle (vehicle-1 frame)	$\dot{\theta}$
Yaw	ψ	Yaw angle (vehicle frame)	$\dot{\psi}$
Roll rate	p	Roll rate measured in the i-axis of the body frame.	$\dot{p}$
Pitch rate	q	Pitch rate measured in the j-axis of the body frame.	$\dot{q}$
Yaw rate	r	Yaw rate measured in the k-axis of the body frame.	$\dot{r}$

Table 3-12 12 state parameters and their derivations

The Forces and Moments equation and representations considered previously need to work in unison with appropriate kinematics and dynamics modelling to create the appropriate simulation of the FMS 182. The table above lists the key important state parameters. Within any given coordinate frames, these parameters can be calculated from rotations and translations [13]. Based on the formulation discussed in the literature review, the following expressions (Eq. 3.25) to calculate these parameters have been used.

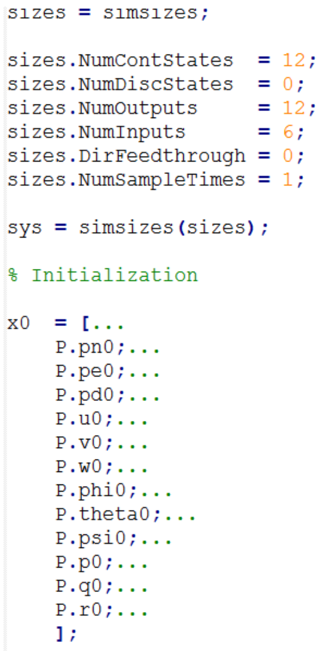
$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_{v2}^{b}(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{v2}^{b}(\phi) R_{v1}^{v2}(\phi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

[Eq. 3.25]

Both interpreted and s-function blocks can be used to implement them in Simulink following the exact same procedure used for the Forces & Moments block. For the dynamics block, all the derivatives are computed based on the transcribed equations while using variable names consistent with the templates (see Chapter 3.4, first paragraph). However, instead of a regular output, all the calculated values are passed on to the matrix “sys” because of the use of s-function blocks [35]. Even though s-functions provide advantages like described in the literature review section, it does require a different format of function block creation compared to the previous block.



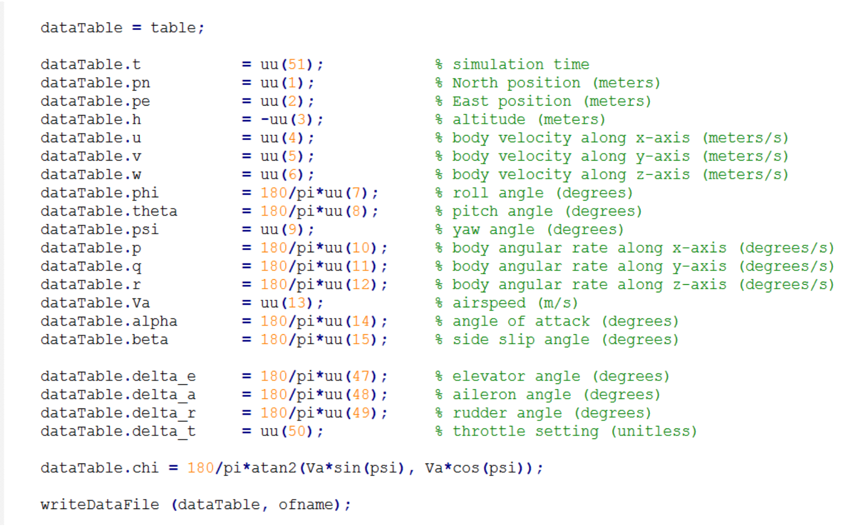
```
sizes = simsizes;

sizes.NumContStates  = 12;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 12;
sizes.NumInputs      = 6;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);

% Initialization

x0  = [...
    P.pn0;...
    P.pe0;...
    P.pd0;...
    P.u0;...
    P.v0;...
    P.w0;...
    P.phi0;...
    P.theta0;...
    P.psi0;...
    P.p0;...
    P.q0;...
    P.r0;...
    ];
```

Figure 3.129 s-function configuration from templates

As shown in Figure 3.129, the vector x0 has to be initialized and various other parameters configured based on how many parameters the block is handling from input to output. Most of it is auto generated by Simulink template that can be used by loading “msfuntmpl_basic.m” or “msfuntmpl.m” from the command window. In our case the the values for Forces and three values for Moments were the inputs to the block so the “sizes.Inputs” parameter needed to be set to 6 while the sates and outputs (sizes.NumContStates & sizes.NumOutputs) were set to 12 to represent the 12 state parameters this block outputs.

3.4.9 Data output

A function from the template (“plotstatevariables”) to create live plots of the state information (see Appendix Chapter 7.3) of interest was modified and extended to meet the requirement of this project. The function uses simple MATLAB functions (e.g. subplot) along with persistent variable definitions and handles to create the live drawing. This code was modified with the aim to extend its capabilities and turn it into a flight data recorder.



```
dataTable = table;

dataTable.t          = uu(51);          % simulation time
dataTable.pn         = uu(1);           % North position (meters)
dataTable.pe         = uu(2);           % East position (meters)
dataTable.h          = -uu(3);          % altitude (meters)
dataTable.u          = uu(4);           % body velocity along x-axis (meters/s)
dataTable.v          = uu(5);           % body velocity along y-axis (meters/s)
dataTable.w          = uu(6);           % body velocity along z-axis (meters/s)
dataTable.phi        = 180/pi*uu(7);    % roll angle (degrees)
dataTable.theta      = 180/pi*uu(8);    % pitch angle (degrees)
dataTable.psi        = uu(9);           % yaw angle (degrees)
dataTable.p          = 180/pi*uu(10);   % body angular rate along x-axis (degrees/s)
dataTable.q          = 180/pi*uu(11);   % body angular rate along y-axis (degrees/s)
dataTable.r          = 180/pi*uu(12);   % body angular rate along z-axis (degrees/s)
dataTable.Va         = uu(13);          % airspeed (m/s)
dataTable.alpha      = 180/pi*uu(14);   % angle of attack (degrees)
dataTable.beta       = 180/pi*uu(15);   % side slip angle (degrees)

dataTable.delta_e    = 180/pi*uu(47);   % elevator angle (degrees)
dataTable.delta_a    = 180/pi*uu(48);   % aileron angle (degrees)
dataTable.delta_r    = 180/pi*uu(49);   % rudder angle (degrees)
dataTable.delta_t    = uu(50);          % throttle setting (unitless)

dataTable.chi = 180/pi*atan2(Va*sin(psi), Va*cos(psi));

writeDataFile (dataTable, ofname);
```

Figure 3.130 Modifications to allow for flight test data to be recorded

Within the loop of the graphing function, a table was generated and organised and stored in memory. The line “ofname = sprintf('Flight test at t = %s.csv',datetime);” was added right before the plotting commands and this translates to the creation of a easily accessible CSV file named based on the unique date and time the simulation commenced. The results are shared in the corresponding results section.

3.5 References

[1] FMS SkyTrainer 182 RC Plane, https://www.motionrc.com/products/fms-sky-trainer-182-v2-5ch-red-1410mm-55-5-wingspan-pnp-fms007-atred , retrieved 10/10/2019

[2] Cessna Skylane, https://cessna.txtav.com/en/piston/cessna-skylane , retrieved 25/06/2021

[3] XFLR5, http://www.xflr5.tech, retrieved 23/05/2020

[4] Developers, X-Plane, https://developer.x-plane.com/docs/aircraft/, retrieved 17/05/2021

[5] Allen, Bob, NACA airfoils, Allen, Bob. "NACA Airfoils". *nasa.gov*. NASA. Retrieved 27 July 2020.

[6] Vural, Murat, and L. M. Nicolai. "Estimating R/C model aerodynamics and performance." *Illinois Institute of Technology* (2009).

[7] Scorpion_calc, Scorpion Systems, https://www.scorpionsystem.com/, retrieved 03/09/2021

[8] Rotaro Digital Tachometer, https://www.rheintacho.de/en/products/digital-hand-tachometers/rotaro/rotaro/, retrieved 03/09/2021

[9] GNU Image Manipulation Program, https://www.gimp.org/, retrieved 03/09/2021

[10] Node Js Javascript, https://github.com/viralpickaxe/xplane-udp, retrieved 03/09/2020

[11] G. J. Mullen, Aircraft Parameter Identification Using Matlab, Volume 11 of Cranfield College of Aeronautics report, Cranfield University, 2000.

[12] R. C. Nelson, Flight Stability and Automatic Control, 2nd Edition, Singapore, McGraw-Hill, 1998, ch.3-5.

[13] R. W. Beard and T. W. McLain. Small Unmanned Aircraft: Theory and Practice. Princeton University Press, USA 2012.

[14] Rotation Matrix Overview: M.W. Spong and M. Vidyasagar, Robot Dynamics and Control. New York: John Wiley & Sons, Inc., 1989

[15] Stevens, B. L., and Lewis, F. L., Aircraft Control And Simulation, 2nd edition, John Wiley & Sons, Hoboken, NJ, 2003.

[16] Phillips, W. F., Mechanics of Flight, John Wiley & Sons, Hoboken, NJ, 2004.

[17] M.V. Cook, Flight Dynamics Principles. New York: John Wiley & Sons, 1997

[18] T. R. Yechout, S. L. Morris, D. E. Bossert, and W. F. Hallgren, Introduction to Aircraft Flight Mechanics. AIAA Education Series, American Institute of Aeronautics and Astronautics, 2003.

[19] H. Goldstein, Classical Mechanics. Cambridge, MA: Addison-Wesley, 1951.

[20] A. V. Rao, Dynamics of Particles and Rigid Bodies: A Systematic Approach. Cambridge: Cambridge University Press, 2006.

[21] J. B. Marion, Classical Dynamics of Particles and Systems. New York: Academic

Press, 2nd edition, 1970.

[22] W. E. Wiesel, Spaceflight Dynamics. New York: McGraw Hill, 2nd ed., 1997.

[23] J. R. Wertz, ed., Spacecraft Attitude Determination and Control. Dordrecht, Neth.: Kluwer Academic Publishers, 1978.

[24] R. F. Stengel, Flight Dynamics. Princeton, NJ: Princeton University Press, 2004.

[25] Aerospace Blockset for Matlab, https://uk.mathworks.com/products/aeroblks.html, retrieved: 02/10/2017

[26] Raymer, Daniel P. (2006). Aircraft Design: A Conceptual Approach, Fourth edition. AIAA Education Series. ISBN 1-56347-829-3

[27] Batallé, Jordi (12 February 2013). "An Introduction to Positional Tracking and Degrees of Freedom (DOF)". Road to VR.

[28] H. Goldstein, Classical Mechanics. Cambridge, MA: Addison-Wesley, 1951.

[29] A. V. Rao, Dynamics of Particles and Rigid Bodies: A Systematic Approach. Cambridge: Cambridge University Press, 2006.

[30] J. B. Marion, Classical Dynamics of Particles and Systems. New York: Academic Press, 2nd edition, 1970.

[31] Patch Function & VertexCData, https://uk.mathworks.com/help/matlab/ref/patch.html, last accessed 17/05/2021.

[32] MQ-9 Reaper 3D model by usafitz, https://www.thingiverse.com/thing:1307919/files, last accessed 17/05/2021

[33] STL files, SolidWorks, Dassault Systemes, https://help.solidworks.com/2018/english/SolidWorks/sldworks/c_stl_files.htm?id=8c45b6fab2de457dbbededfade3a27ed#Pg0, last accessed 17/05/2021

[34] Francis Esmonde-White (2021). Binary STL file reader (https://www.mathworks.com/matlabcentral/fileexchange/29906-binary-stl-file-reader), MATLAB Central File Exchange. Retrieved 17/05/2021.

[35] S-Function blocks, MathWorks, https://www.mathworks.com/help/simulink/sfg/what-is-an-s-function.html. Retrieved 02/05/2020

[36] Custom MATLAB Function block, Mathworks, https://www.mathworks.com/help/simulink/ug/creating-an-example-model-that-uses-a-matlab-function-block.html, retrieved 02/05/2020

4 Results

4.1 XFLR5 Results

4.1.1 Aerofoil Creation Results

The following polars were obtained for the aerofoils modelled and analysed. Only the 3 standard NACA foil polars are shown here for demonstration purpose. The variants of the following 3 aerofoils (e.g. NACA 1214 FMS 15 deg aileron down) were also used to generate the same type of polars. Those example results can be found in the Appendix (Chapter 7.6.1).

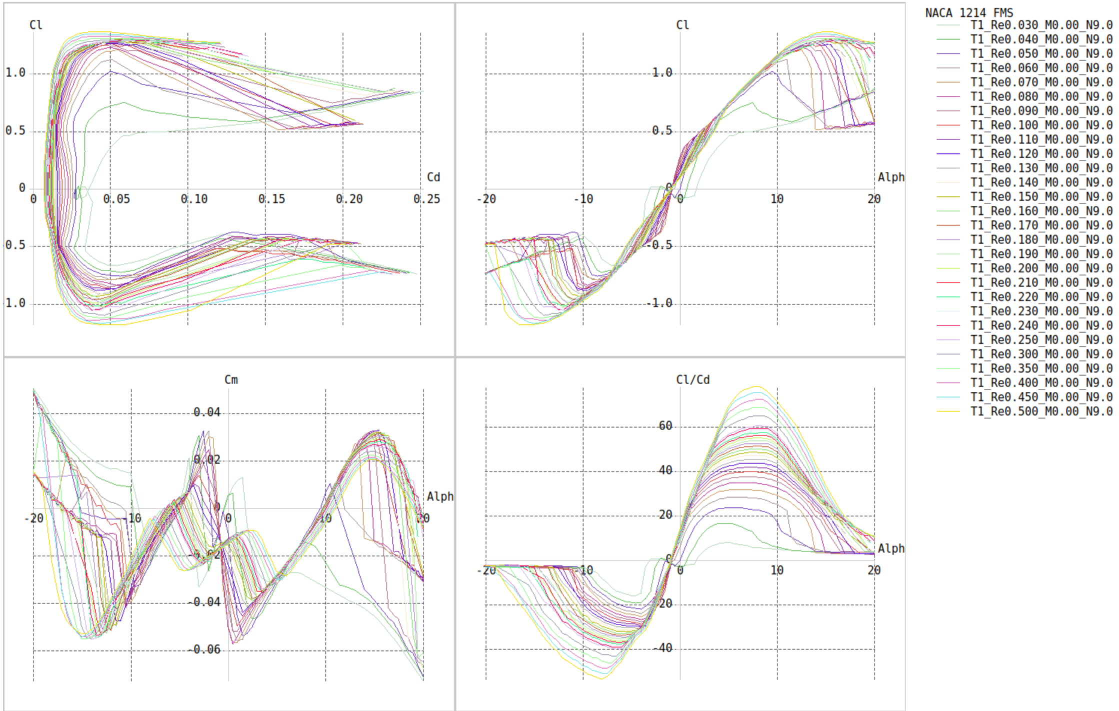


Figure 4.1 NACA 1214 FMS (Re 30k to 500k)

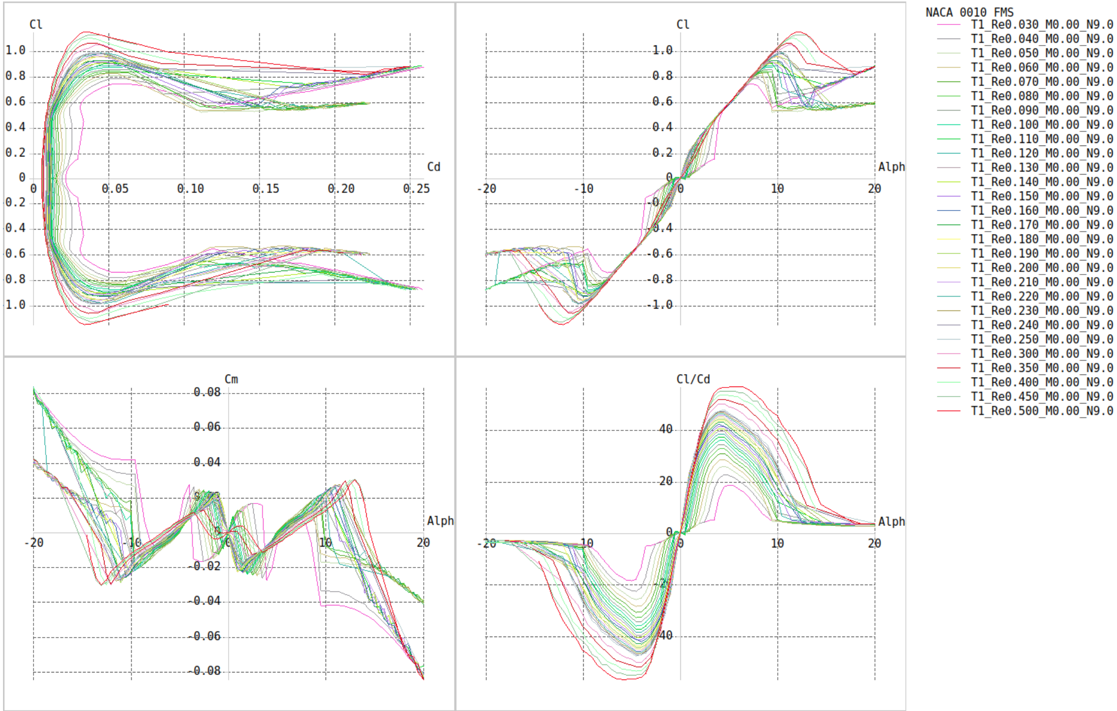


Figure 4.2 NACA 0010 FMS (Re 30k to 500k)

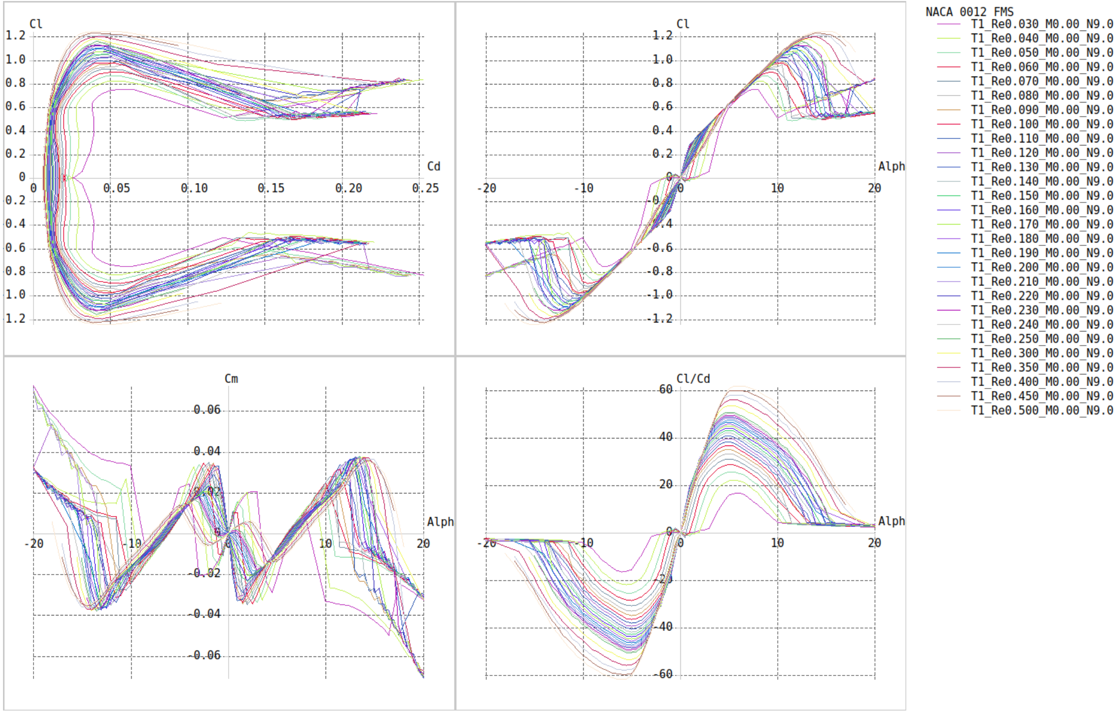


Figure 4.3 NACA 0012 FMS (Re 30k to 500k)

For virtually all the foils analysed, the greatest amount of change (turbulence related variance) was observed at low Reynolds numbers. This is expected as the transition from laminar to turbulent flow occurs at fairly low Reynolds numbers. At higher Reynolds number ranges, the outputs do not show such high degree of variance. This reduced amount of variance allowed for sparser polar density in the higher Reynolds number range for the overall analysis without affecting the precision of the interpolation method used by XFLR5.

Under some test conditions, the subsequent 3D analysis (for performance, stability and control tests) indicated local Reynolds number outside of the range of the original 2d analysis. The software interpolates between two given Reynolds numbers, and deliberately avoids extrapolating outside of this range. This is done to avoid unfounded assumptions lowering the confidence in the simulation results. As such, various other Reynolds number ranges were needed and included in the dataset to ensure the integrity of the subsequent simulations. That resulting extended dataset is too big to be represented within the body of this document (graphically) or added to the Appendix (tabular). As a result, they were compiled as CSV files and uploaded to an online repository linked in the Appendix.

4.1.2 Plane Modelling Results

Main Wing	Magnitude	Units
Wing Span	1410	mm
Area	0.27	m^2
Projected Span	1409.75	mm
Projected Area	0.27	m^2
Mean Geometric Chord	190.17	mm
Mean Aerodynamic Chord	193.18	mm
Aspect Ratio	7.41	
Taper Ratio	1.53	
Root to Tip Sweep	-1.48	degrees
VLM Panels	676	
3D Panels	1378	

Table 4-1 Main wing modelling data for FMS 182

Horizontal Stabiliser	Magnitude	Units
Wing Span	450	mm
Area	0.06	m^2
Projected Span	450	mm
Projected Area	0.06	m^2
Mean Geometric Chord	136	mm
Mean Aerodynamic Chord	138.67	mm
Aspect Ratio	3.31	
Taper Ratio	1.64	
Root to Tip Sweep	3.92	degrees
VLM Panels	98	
3D Panels	210	

Table 4-2 Horizontal Stabiliser modelling data for FMS 182

Vertical Stabiliser	Magnitude	Units
Wing Span	390	mm
Area	0.03	m^2
Projected Span	390	mm
Projected Area	0.03	m^2
Mean Geometric Chord	130	mm
Mean Aerodynamic Chord	134.96	mm
Aspect Ratio	3	
Taper Ratio	2.02	
Root to Tip Sweep	33.21	degrees
VLM Panels	49	
3D Panels	112	

Table 4-3 Vertical Stabiliser modelling data for FMS 182

4.1.2.1 Inertia:

4.1.2.1.1 Main Wing:

Empty Wing CoG	**Magnitude**	**Units**		**Empty wing Inertia (CoG Frame)**	**Magnitude**	**Units**
Wing Mass	232	grams		Ixx	0.03	kg.m^2
X_CoG	82.517	m		Iyy	0.0005148	kg.m^2
Y_CoG	0	m		Izz	0.031	kg.m^2
Z_CoG	8.063	m		Ixz	5.42E-06	kg.m^2
Wing CoG (Total)	magnitude	units		Total Inertia (CoG Frame)	magnitude	units
Total Mass	291	grams		Ixx	0.03322	kg.m^2
X_CoG	84.151	mm		Iyy	0.00054	kg.m^2
Y_CoG	1.206	mm		Izz	0.03375	kg.m^2
Z_CoG	6.428	mm		Ixz	8.48E-06	kg.m^2

Table 4-4 Inertial data for the main wing of the FMS 182

4.1.2.1.2 Horizontal Stabiliser:

Horizontal Stabiliser CoG (Total)	**Magnitude**	**Units**		**Total Inertia (CoG Frame)**	**Magnitude**	**Units**
Total Mass	29	grams		Ixx	0.00037	kg.m^2
X_CoG	640.923	mm		Iyy	0.00003	kg.m^2
Y_CoG	0	mm		Izz	0.000041	kg.m^2
Z_CoG	-118.09	mm		Ixz	-8.48E-07	kg.m^2

Table 4-5 Inertial data for the Horizontal Stabiliser of the FMS 182

4.1.2.1.3 Vertical Stabiliser:

Vertical Stabiliser CoG (Total)	**Magnitude**	**Units**		**Total Inertia (CoG Frame)**	**Magnitude**	**Units**
Total Mass	25	grams		Ixx	7.03E-05	kg.m^2
X_CoG	703.775	mm		Iyy	0.0001213	kg.m^2
Y_CoG	5.10E-05	mm		Izz	5.10E-05	kg.m^2
Z_CoG	-33.678	mm		Ixz	-4.06E-05	kg.m^2

Table 4-6 Inertial data for the Vertical Stabiliser of the FMS 182

4.1.2.1.4 Overall Aircraft Inertia:

Overall CoG (Total)	**Magnitude**	**Units**		**Total Inertia (CoG Frame)**	**Magnitude**	**Units**
Total Mass	1321.8	grams		Ixx	4.12E-02	kg.m^2
X_CoG	60.364	mm		Iyy	0.03961	kg.m^2
Y_CoG	2.00E-03	mm		Izz	7.28E-02	kg.m^2
Z_CoG	-69.153	mm		Ixz	-5.70E-04	kg.m^2

Table 4-7 Inertial data on the entire FMS 182 aircraft as a whole

4.1.3 Performance Analysis Polars

The following sections contain the results from the performance analysis conducted on the FMS 182 XFLR5 model based on the 2D and viscous analysis performed earlier. The analysis is presented in the following form:

- The polars obtained for the analysis of the model without any control surface deflection modelled (base model).
- Comparative analysis of the polars of a particular model variants using the maximum and minimum control surface deflections along with the base model.

4.1.3.1 Base Model (Clean)

The following graphs have been obtained for the base model (FMS 182b).

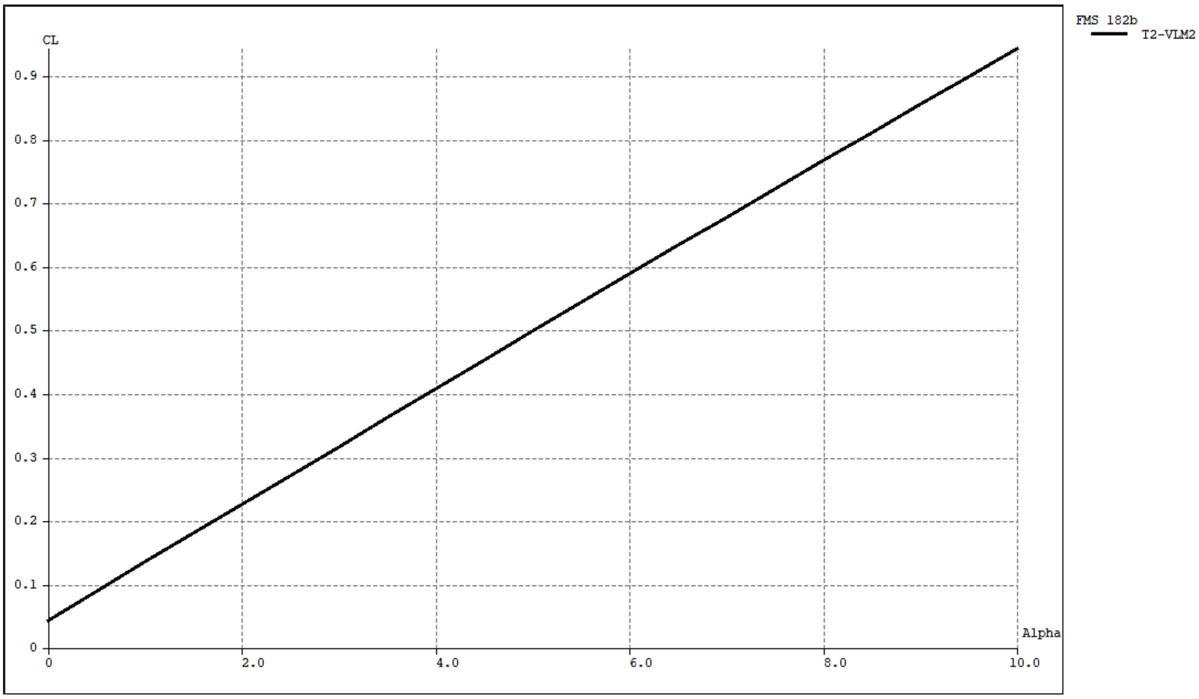


Figure 4.4 CL (y-axis) vs Alpha (x-axis) for the model

Figure 4.4 represents the lift coefficient of the entire model (as referenced to the projected wing area) against angle of attack. Technically, this includes the contribution of the horizontal stabiliser. However, that contribution is negligible due to the size of the elevator and it's angle of attack. As such, the main wing is the primary contributor to this graph. For alpha range of 0 to 10, the relationship between the lift coefficient and the angle of attack is linear as it's to be expected until the stalling point. It is also notable that at 0 degrees, the wing still produces lift (owing to the NACA 1214 aerofoil's modest camber).

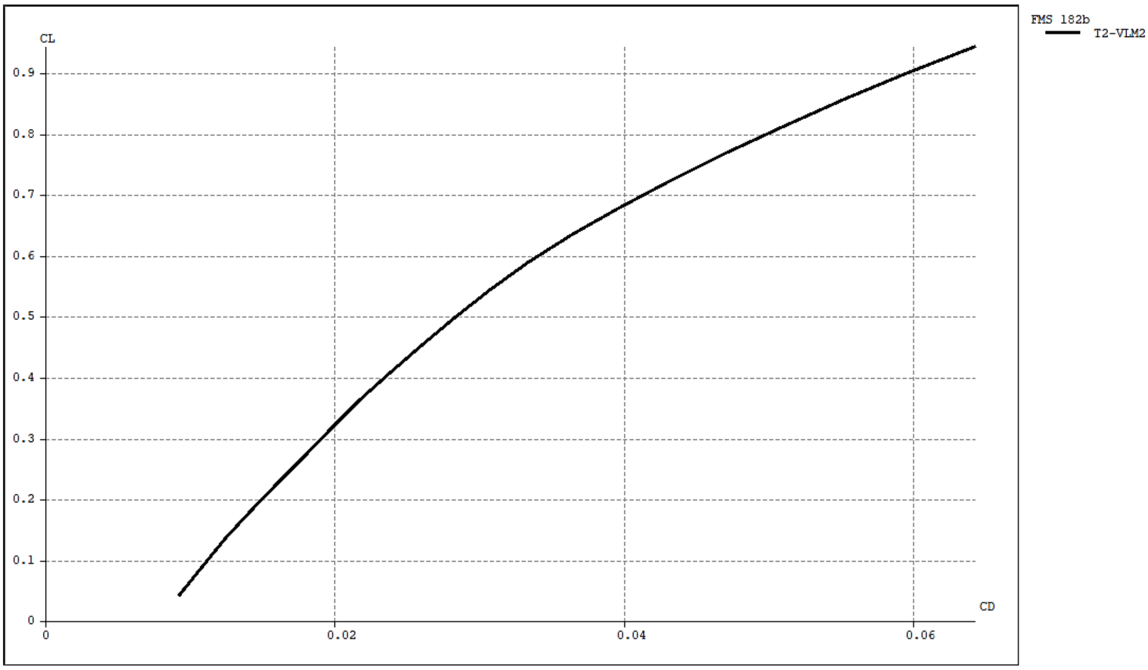


Figure 4.5 CL (y-axis) vs CD (x-axis) for the model

Figure 4.5 represents the relationship between lift and drag of the model within the constraint of straight and level flight. This is due to the fact that the T2-VLM2 test adjusts the airspeed to maintain a level flight. This graph does not include the contribution of any elevator deflection required to maintain level flight outside of the trim condition. The profile obtained is consistent with the anticipated relationship trend of CD being proportional to the square of CL. As such, it indicates that the software's prediction is consistent with expected outcomes. This is indicative of appropriate solver functioning.

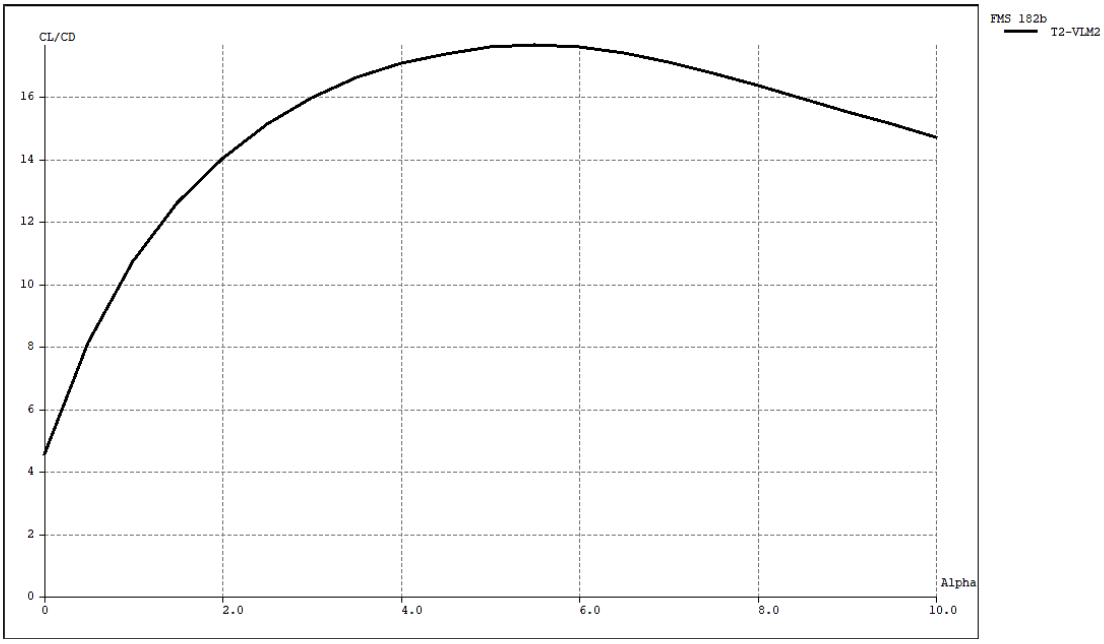


Figure 4.6 CL/CD (y-axis) against Alpha (x-axis) for the model

Figure 4.6 illustrates the aerodynamic efficiency of the model with respect to its straight and level angle of attacks. As a higher angle of attack is required at lower speed, this graph serves as an indication of the most efficient operational speed of the model (Not accounting for elevator deflection). The alpha at that peak is 5.5 degrees approximately and this equates to an efficient cruising speed of about 12 meters per second. CL/CD = 17.645.

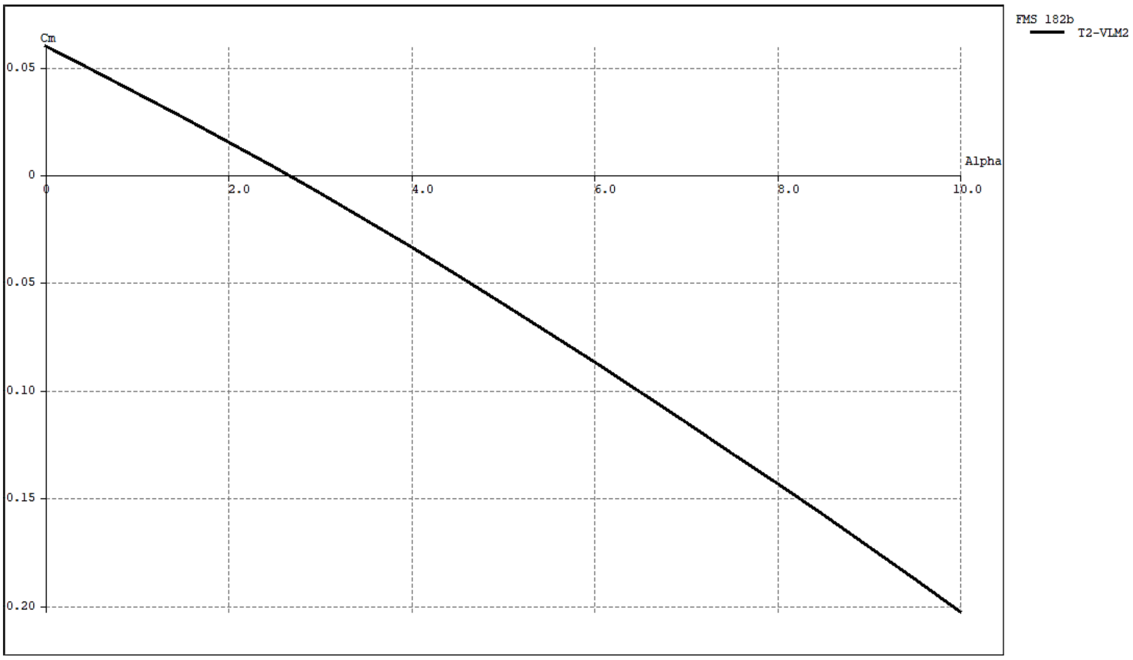


Figure 4.7 Cm (y-axis) against Alpha (x-axis) for the model

Figure 4.7 shows the relationship of the pitching moment of the model in the current trim condition for variations in the angle of attack. A negative slope is indicative of a statically stable aircraft where a perturbation in the pitch axis results in a countering moment which corrects the resulting attitude change such that the original attitude is restored. What is evident on this graph is that in the current horizontal stabiliser configuration, the model is in trim at 2.75 degrees angle of attack (where the curve crosses the x-axis). For this model, this equates to a speed between 17.05 (2.5 degrees alpha) meters per second and 15.82 meters per second (3 degrees alpha).

4.1.3.2 Flaps

The following graphs represent the performance analysis of the model with the flaps down compared to the base model.

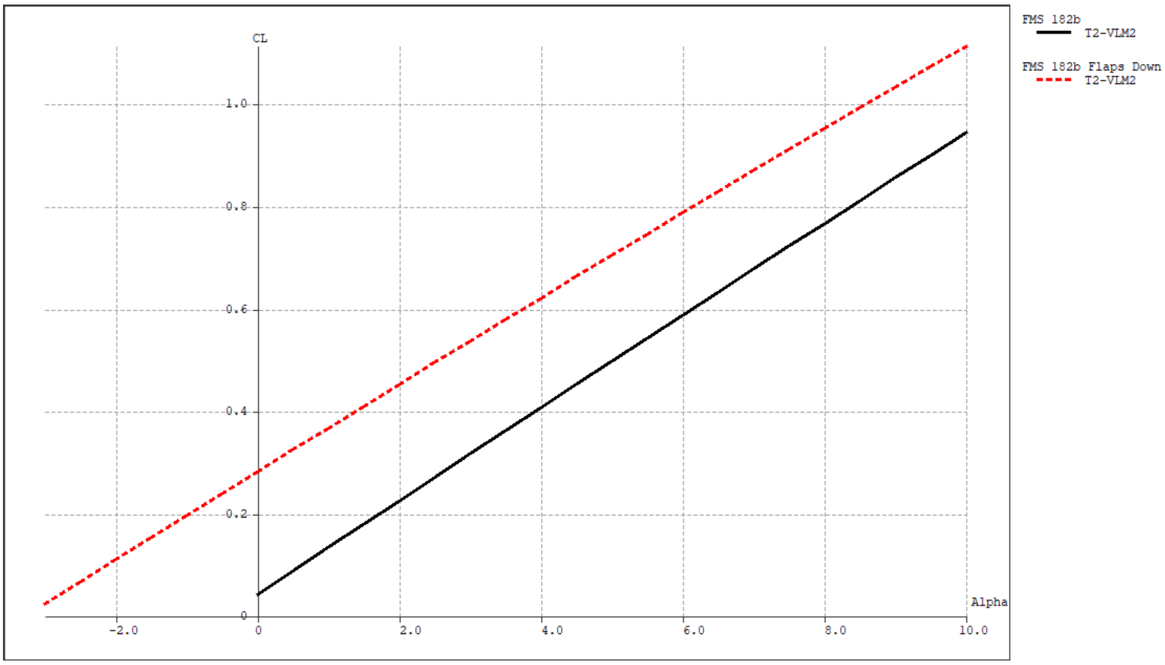


Figure 4.8 Comparison of lift performance of FMS 182b (black) and FMS 182b Flaps down (red)

The above Cl vs alpha curves (Figure 4.8) show that considerable performance gain in lift is achieved through the flap deflection as the estimated CL per unit of alpha is much higher for the flapped model. The same linear trend is observed except for the additional lift provided by the extra camber resulting from the flap deflection.

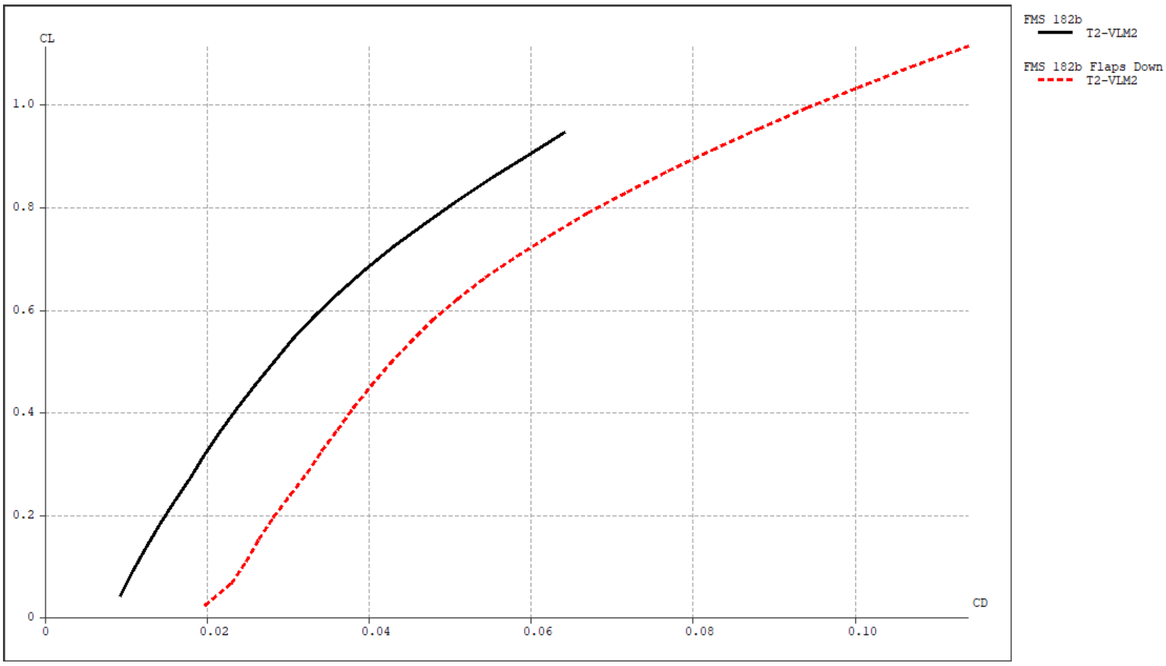


Figure 4.9 Comparison of CL against Alpha of FMS 182b (black) and FMS 182b Flaps down (red)

The same trend is followed for the lift-drag ratio except that the additional drag resulting from the flap deployment translates the curve horizontally.

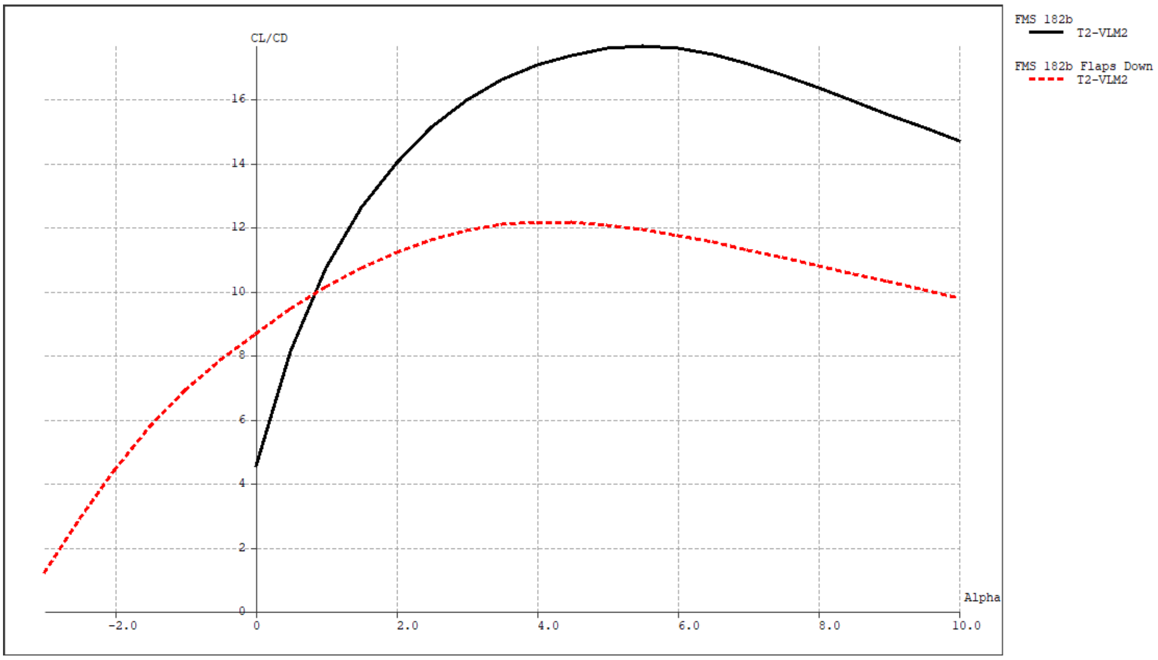


Figure 4.10 Comparison of CL/CD against Alpha of FMS 182b (black) and FMS 182b Flaps down (red)

For the comparative aerodynamic efficiency, Figure 4.10 illustrates that the deployment of the flaps will result in decreased performance (CL/CD 12.142 at alpha 4). This however is a desirable characteristic during certain phases of flight (e.g. landing) where it is necessary to slow the aircraft down as quickly as possible.

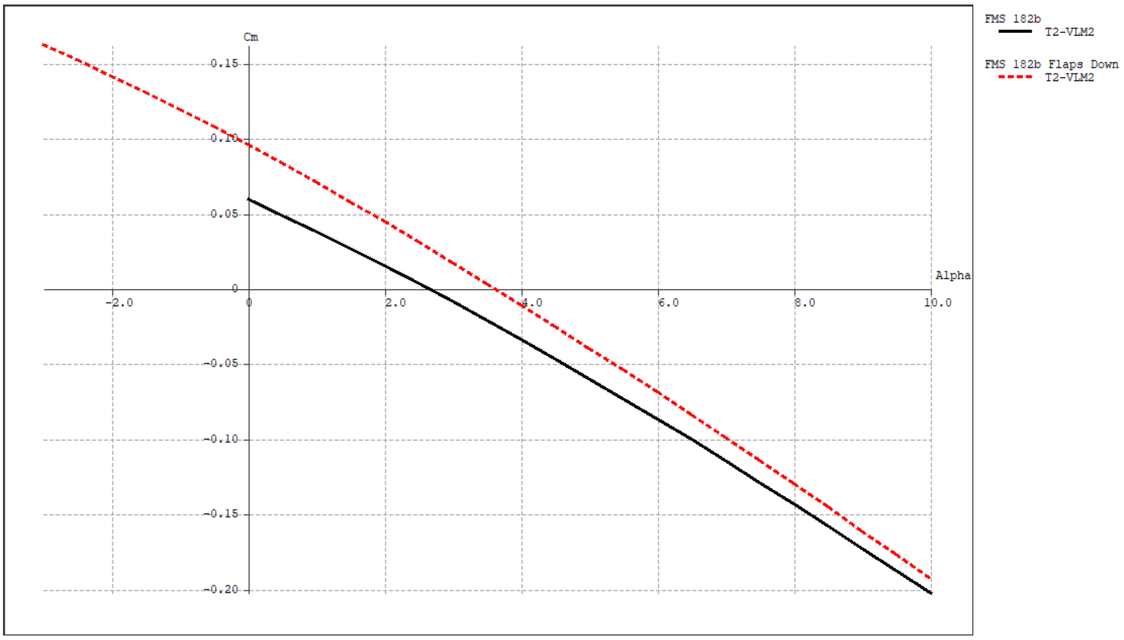


Figure 4.11 Comparison of Cm against Alpha of FMS 182b (black) and FMS 182b Flaps down (red)

As the above graph (Figure 4.11) shows, application of the flaps results in a shift of the curve for the centre of moment against alpha. The position where the CM is zero moves to alpha = 3.5 indicating that a higher angle of attack will be needed for the trim condition.

4.1.3.3 Ailerons

The following graphs represent the performance analysis of the models with aileron deflections compared to the base model.

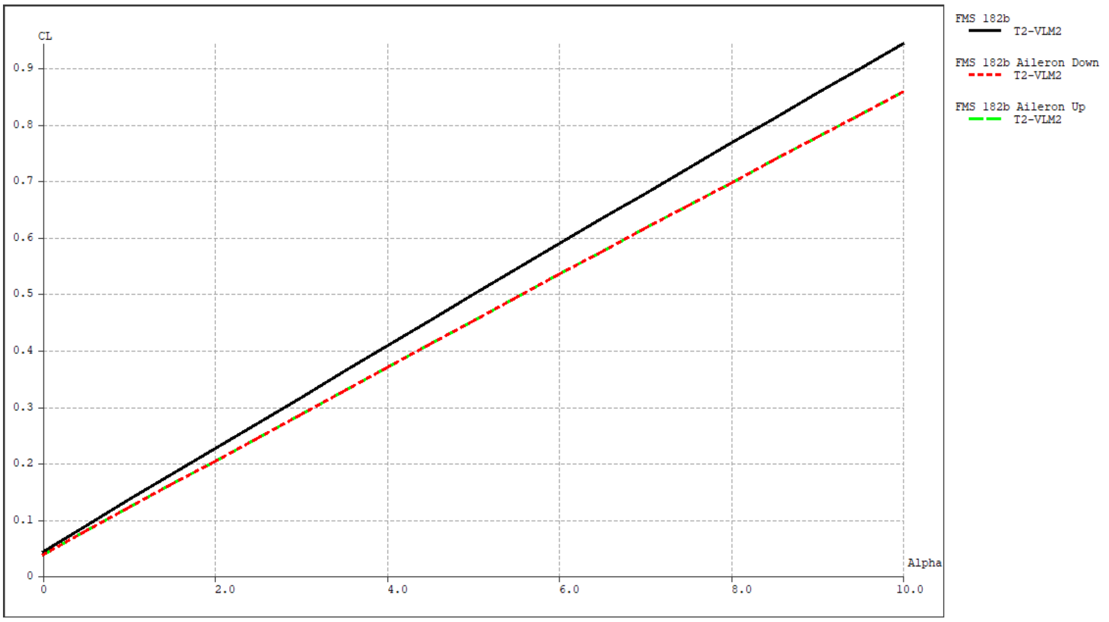


Figure 4.12 Comparison of CL against Alpha of FMS 182b (black) and Aileron Down (red)/Aileron Up (green) models

As seen in the above figure (Figure 4.12), application of the ailerons results in a reduced lift performance of the models. Both the right and the left ailerons produce equal magnitude of reduction in this performance (in their maximum deflections) and as a result their curves (red and green) overlap.

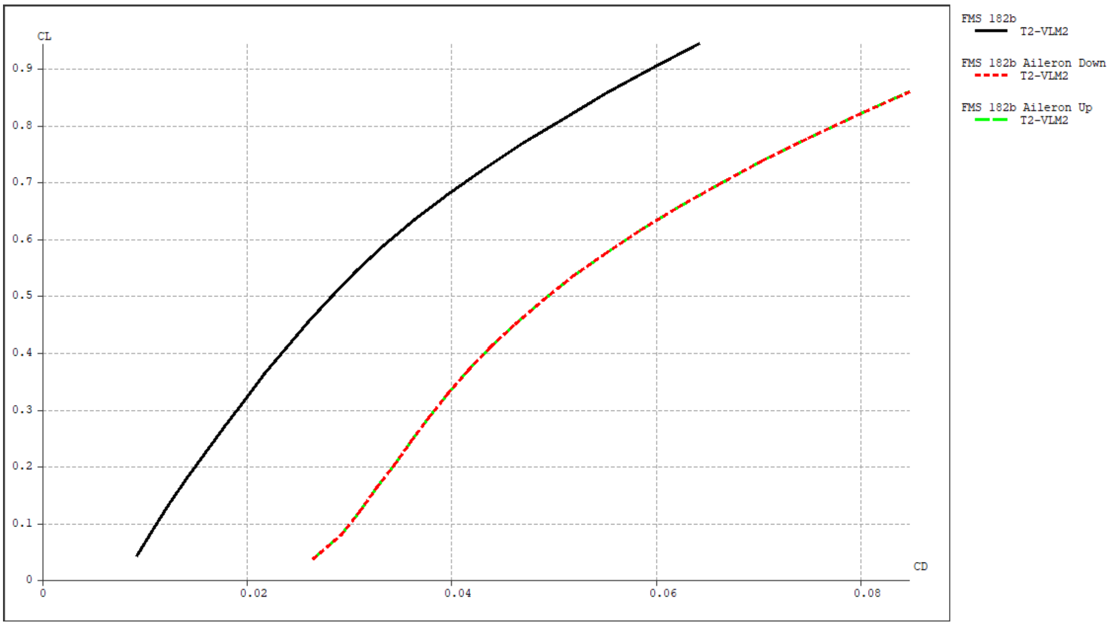


Figure 4.13 Comparison of CL against CD of FMS 182b (black) and Aileron Down (red)/Aileron Up (green) models

As the above figure (Figure 4.13) shows, application of the ailerons results in a horizontal translation of the lift/drag curve due to the additional drag resulting from their deflections. Like the previous example, their curves overlap.

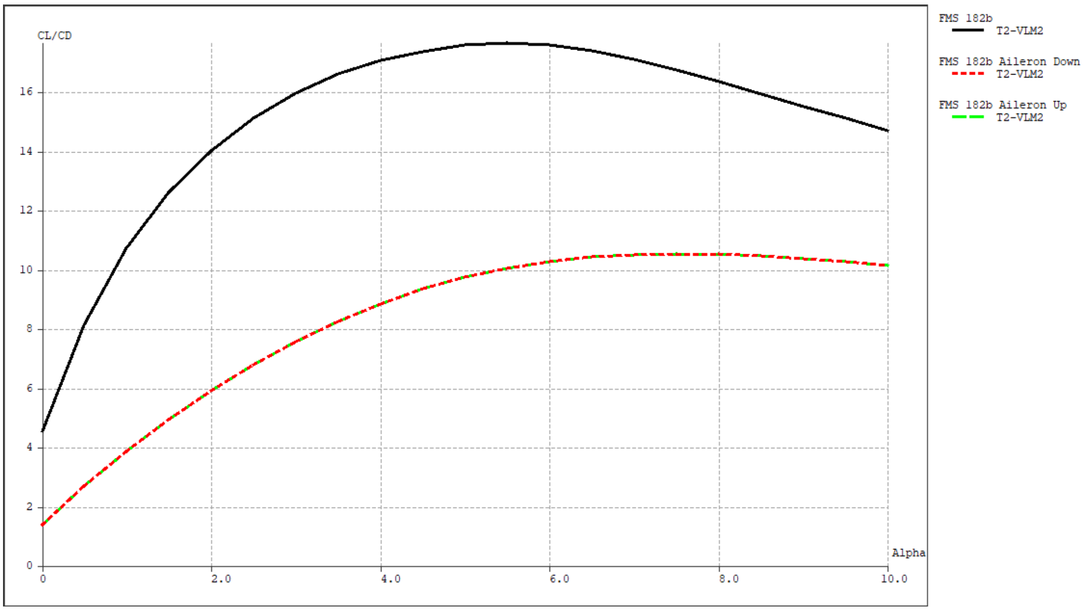


Figure 4.14 Comparison of CL/CD against Alpha of FMS 182b (black) and Aileron Down (red)/Aileron Up (green) models

The same expected overlap of the curves for the aileron deflection models is seen in Figure 4.14. Compared to the base model, the lift to drag ratio is significantly compromised. The maximum L/D is 10.52 at 7.5 alpha (10.98 m/s cruise speed).

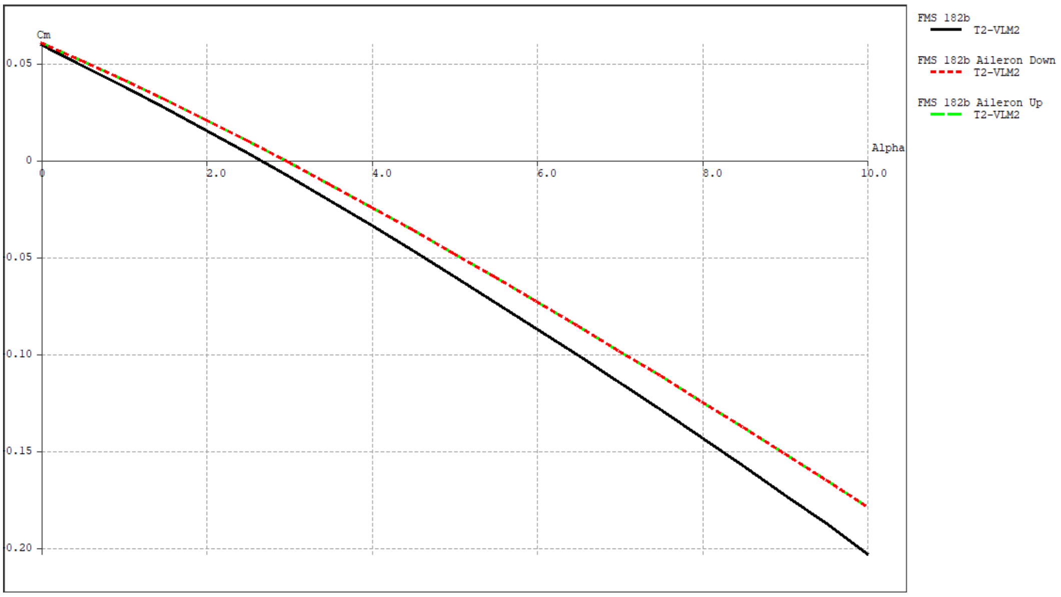


Figure 4.15 Comparison of Cm against Alpha of FMS 182b (black) and Aileron Down (red)/Aileron Up (green) models

From the Cm/alpha curve, it can be seen that the zero CM (trim condition) moves to alpha = 3 degrees (16.62 m/s).

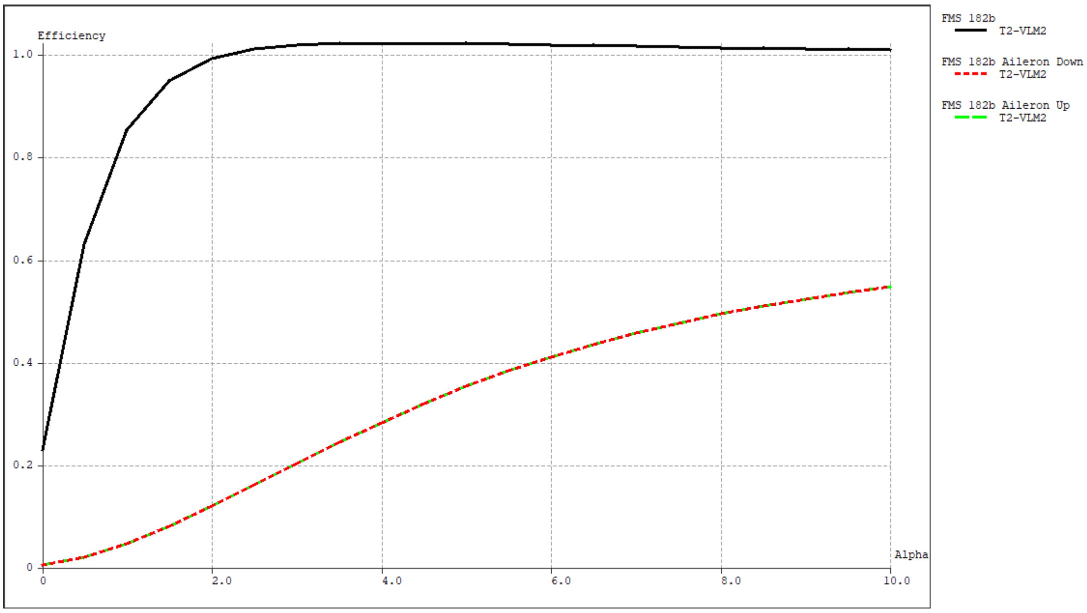


Figure 4.16 Comparison of Ostwald’s efficiency against Alpha of FMS 182b (black) and Aileron Down (red)/Aileron Up (green) models

The above graphs (Figure 4.16) show that the aileron deflections significantly impact the Oswald’s efficiency of the wing. This is to be expected as the Oswald’s efficiency drops due to the rise in the induced drag from the tips of the ailerons (see PAV).

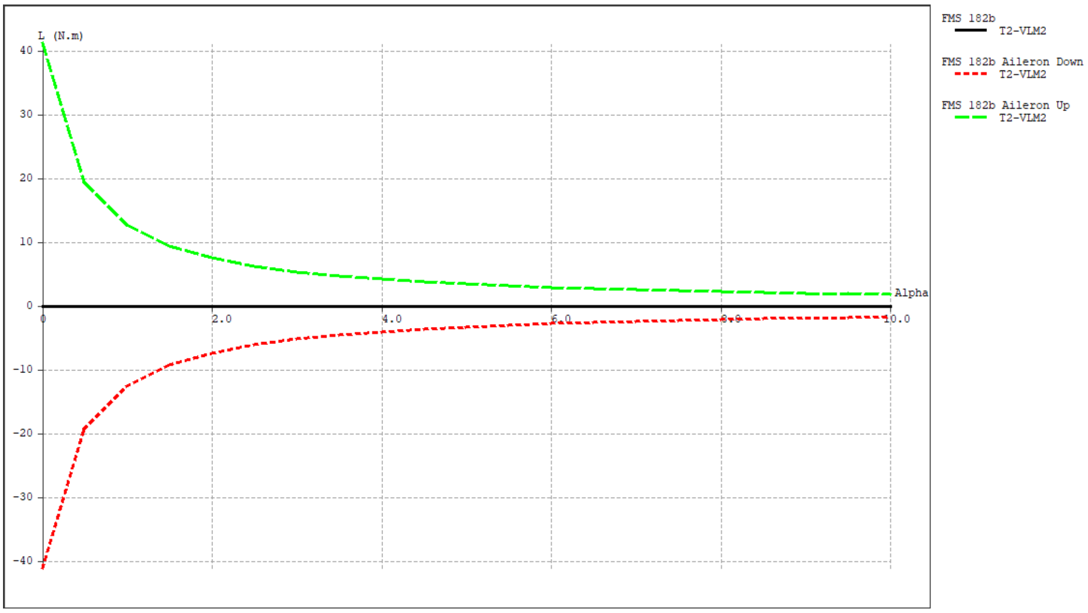


Figure 4.17 Comparison of rolling moment (L) against Alpha of FMS 182b (black) and Aileron Down (red)/Aileron Up (green) models

Figure 4.17 illustrates the rolling moments against alpha for the 3 models. As expected, the base model produces no rolling moment. The full deflection models are projected to produce 40 Newton meters of rolling moment at 0 alpha (46.33 m/s). As angle of attack increases, the airspeed falls and the resulting reduced control effectiveness is reflected in the graph as decaying moment production by the ailerons.

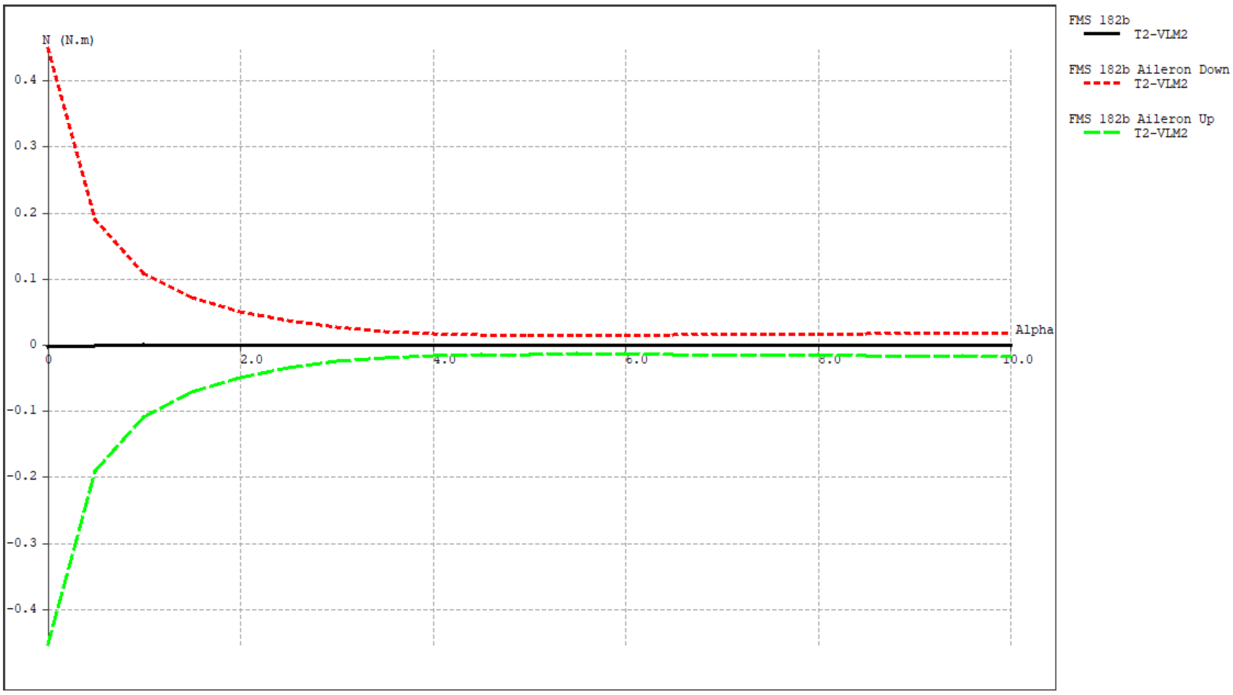


Figure 4.18 Comparison of yawing moment (N) against Alpha of FMS 182b (black) and Aileron Down (red)/Aileron Up (green) models

The aileron deflection creates asymmetric wake from the wing. Which, when it impacts the vertical stabiliser, causes an asymmetric response resulting in a yawning moment. This is shown in the above graph (Figure 4.18). While the trend is similar to that of the rolling moment, the magnitude is significantly low and the direction is reversed. Note that this excludes the flight dynamic effects of actual roll which may/may not counteract this disturbance.

4.1.3.4 Elevator:

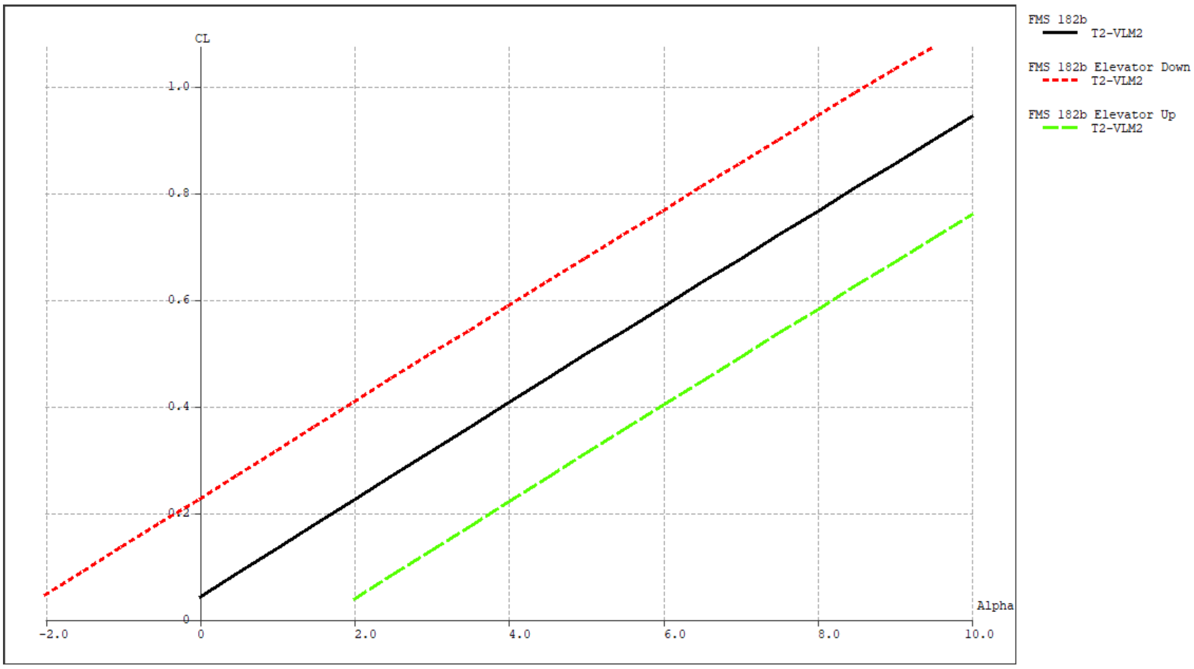


Figure 4.19 Comparison of CL against Alpha of FMS 182b (black) and Elevator Down (red)/Elevator Up (green) models

The positive deflection (down) of the elevator causes the entire horizontal stabilizer to produce lift which usually results in a pitching moment and causes a pitch shift of the aircraft. However, the pitch is held constant for every operating point simulation in this test, thus the resulting lift is added to the total lift generated by the model. This shifts the curve upwards while preserving the general trend of the base model's curve. The opposite is observed for the elevator deflection in the opposite direction as the resulting lift is directed downward and the net lift is subtracted from the main wing's lift. These plots are therefore not representative of the actual lift that will be generated by the model in flight.

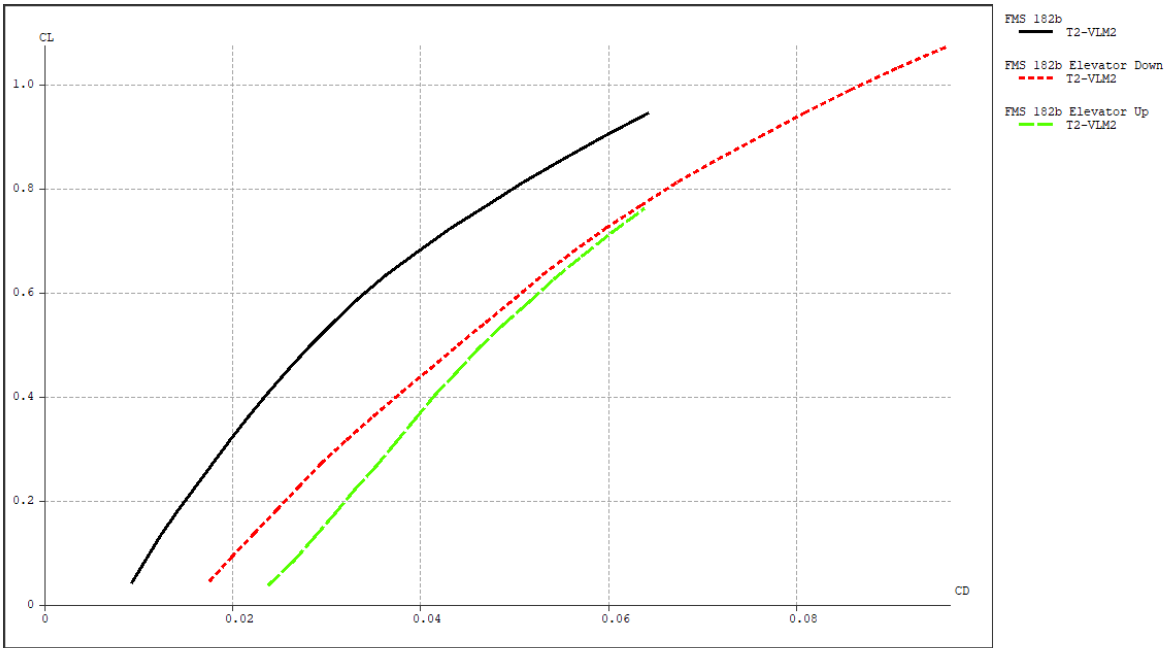


Figure 4.20 Comparison of CL against CD of FMS 182b (black) and Elevator Down (red)/Elevator Up (green) models

For the lift to drag ratio, additional drag resulting from the application of the elevator can be seen in the graph represented in the form of a horizontal shift. Unlike the ailerons, elevator deflection causes asymmetric changes in total airframe lift and thus the curves for the deflection models don’t overlap.

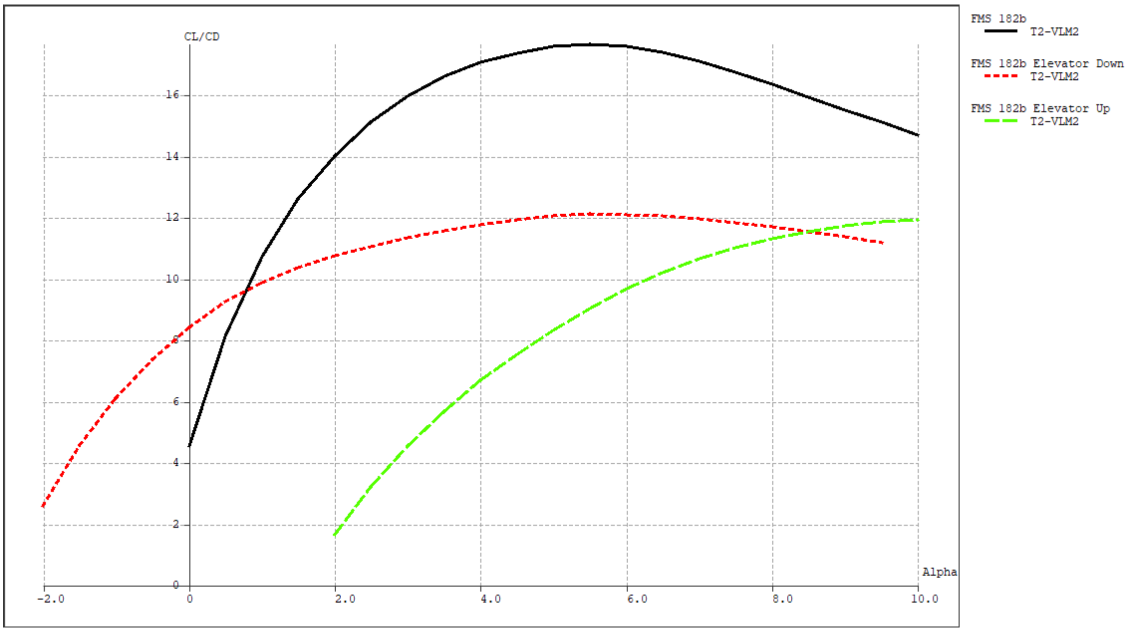


Figure 4.21 Comparison of CL/CD against Alpha of FMS 182b (black) and Elevator Down (red)/Elevator Up (green) models

The additional drag from the deployment of the control surfaces result in a reduction in the overall aerodynamic efficiency. For elevator down, L/D max is at alpha 5.5 (l/d 12.1 v= 12.1) and for elevator up, L/D max is at alpha 10 (l/d 11.9 v =10.20).

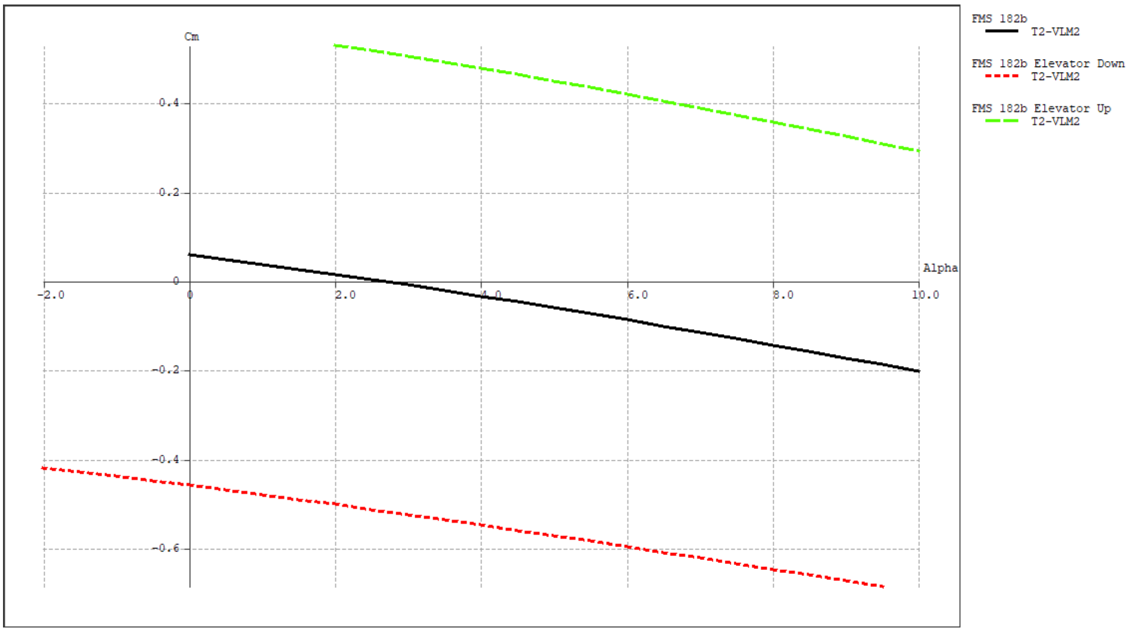


Figure 4.22 Comparison of Cm against Alpha of FMS 182b (black) and Elevator Down (red)/Elevator Up (green) models

Maximum elevator deflections cause significant pitching moments which alter the trim condition outside the flight envelope of the aircraft. This is good for the control authority but should be used with care as sustained control input may result in unsustainable flight attitudes (e.g. stall) potentially resulting in a crash.

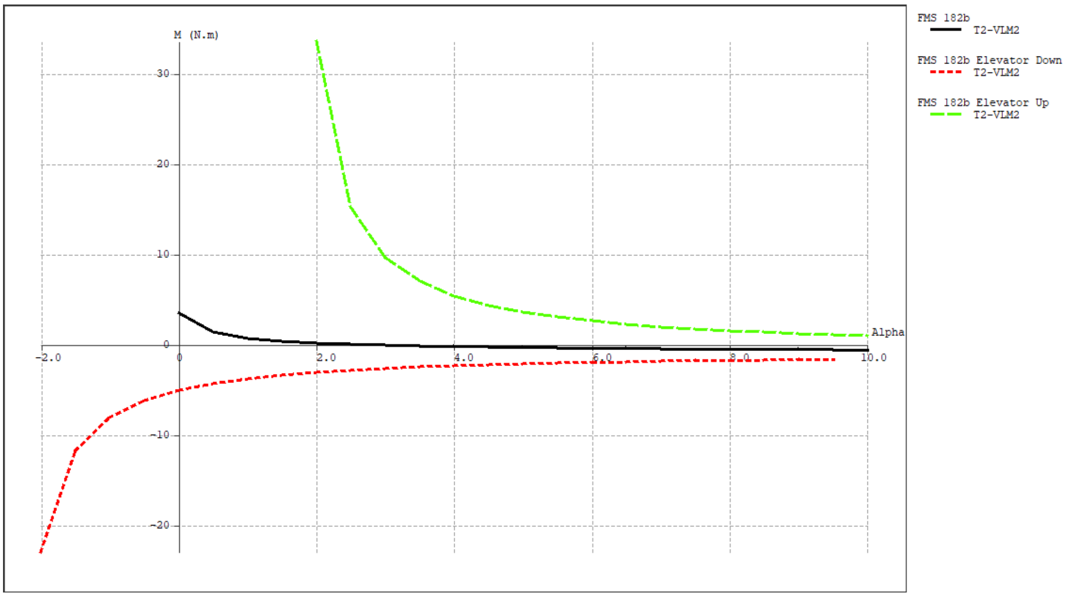


Figure 4.23 Pitching moment comparison of FMS 182b (black) and Elevator Down (red)/Elevator Up (green) models

For the base model, the incidence of the horizontal stabiliser results in a positive pitching moment even at 0 alpha but this is counteracted as alpha increases. Due to the aforementioned addition/subtraction of lift, the curves are phase shifted.

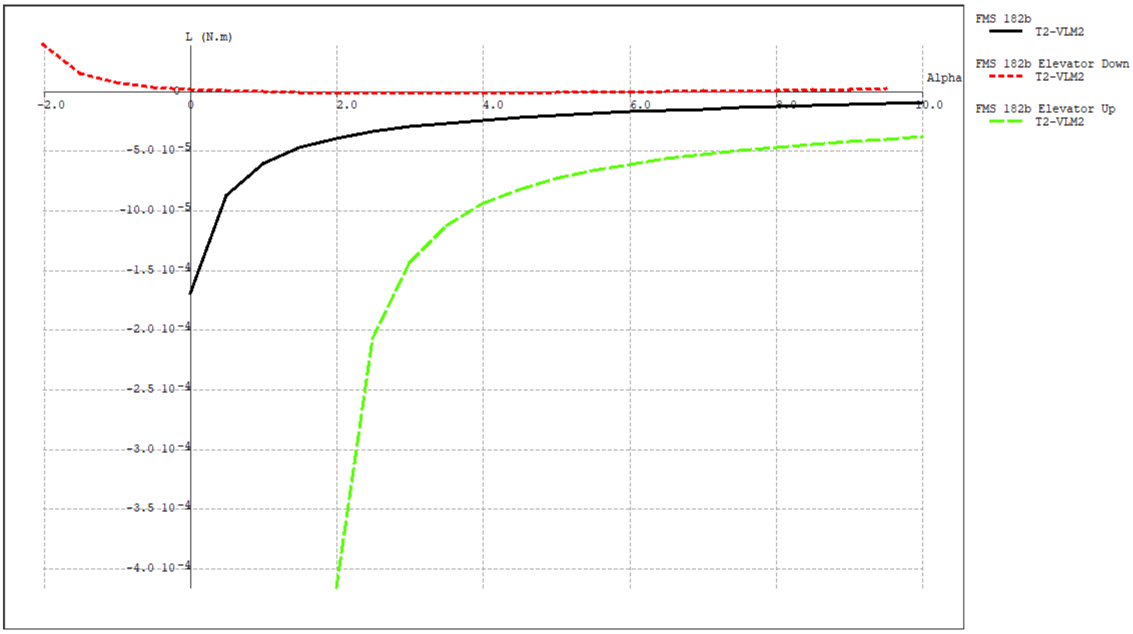


Figure 4.24 Rolling moment comparison of FMS 182b (black) and Elevator Down (red)/Elevator Up (green) models

For the rolling moments, one would expect zero across the board as no asymmetric lift production is expected. However, XFLR5’s is imperfect and created some spurious forces and moments. Fortunately, their magnitude is on the scale of a thousand times smaller and negligible as can be seen on the graph. Therefore it should not impact the simulation fidelity.

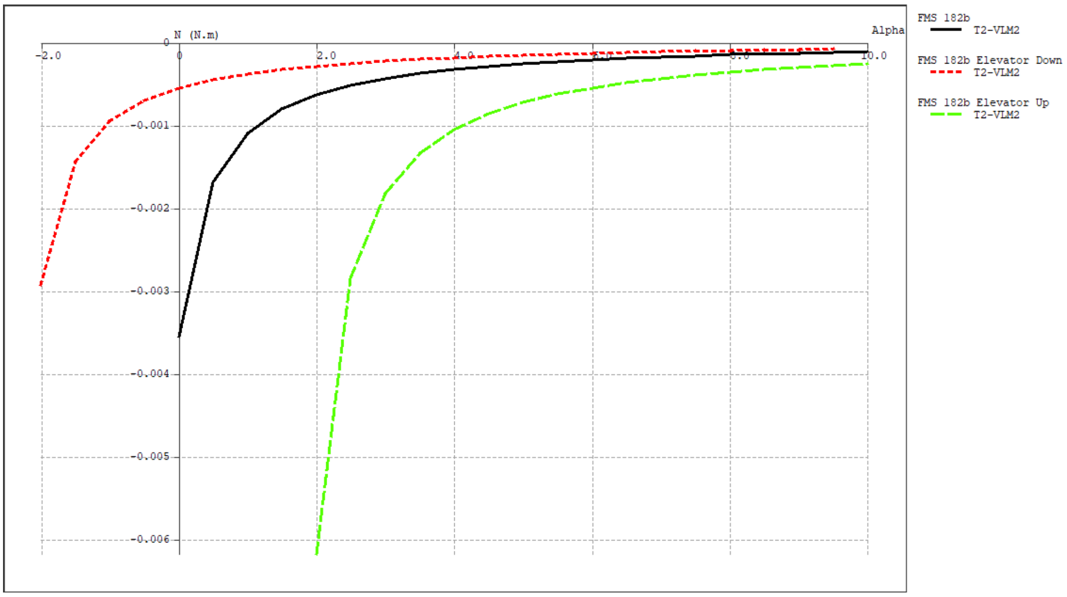


Figure 4.25 Comparison of yawing moment against Alpha of FMS 182b (black) and Elevator Down (red)/Elevator Up (green) models

For the yawing moments, the same type of simulation artefacts is observed. Much like the previous issue, their magnitude is on the scale of a thousand times smaller and negligible as can be seen on the graph. Therefore it should not impact the simulation fidelity.

4.1.3.5 Rudder

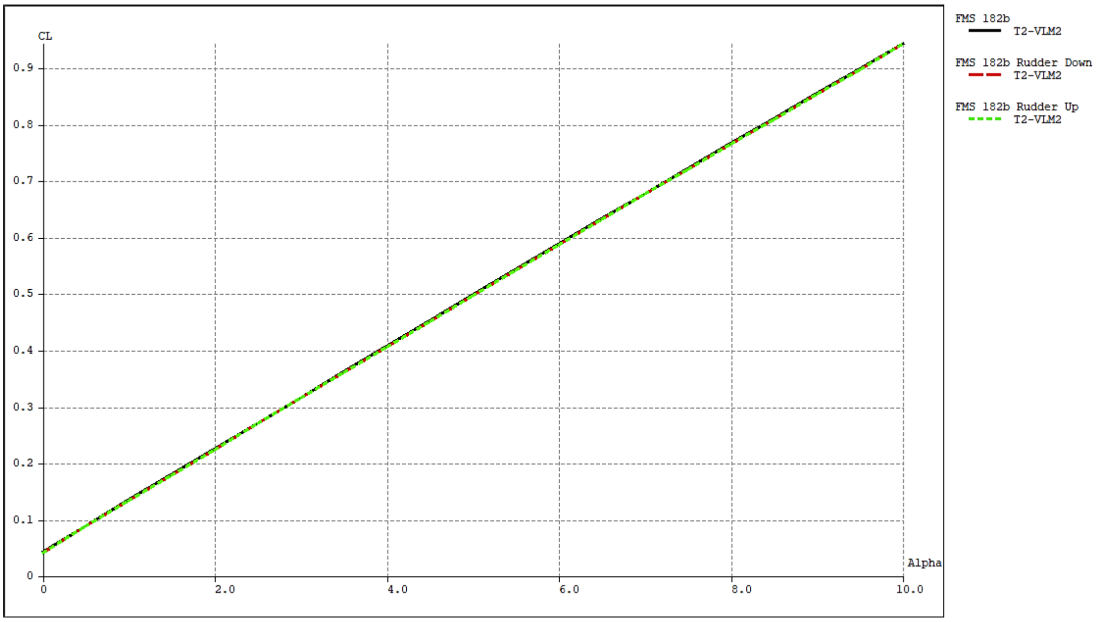


Figure 4.26 Comparison of CL against Alpha of FMS 182b (black) and Rudder Down (red)/Rudder Up (green) models

Application of the rudder causes no noticeable change in the overall lift production of the models as the above graph shows. The magnitudes are close enough such that the curves all overlap.

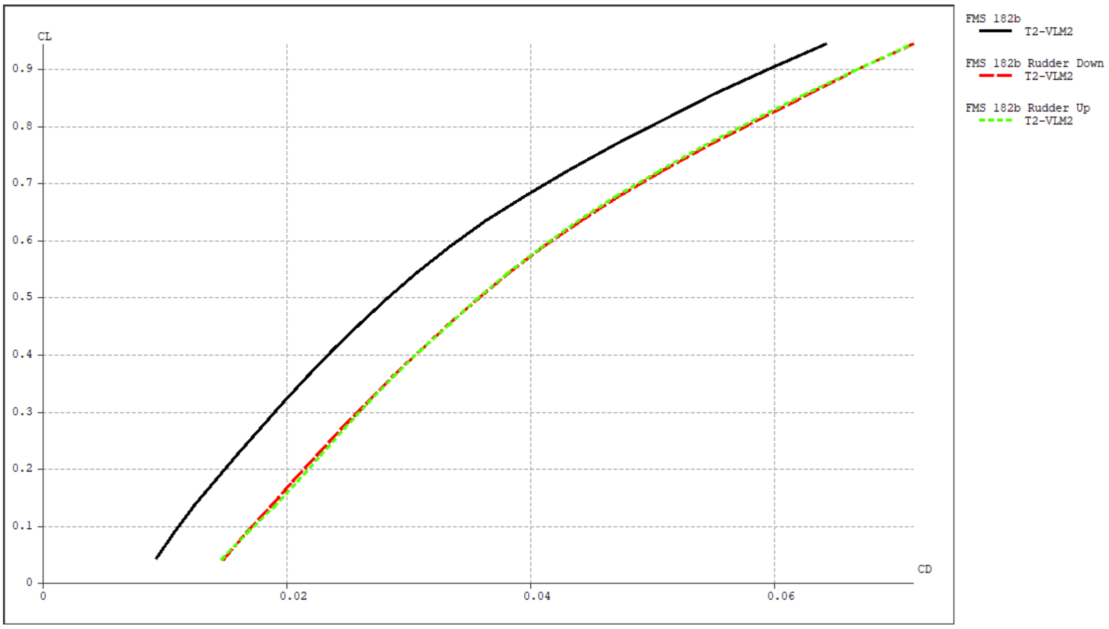


Figure 4.27 Comparison of CL against CD Alpha of FMS 182b (black) and Rudder Down (red)/Rudder Up (green) models

The lift-drag curve of the models with control surface deflection is shifted in the horizontal direction due to the added drag resulting from the application of the rudder.

Otherwise, the save trend is followed by the deflection models when compared to the base models.

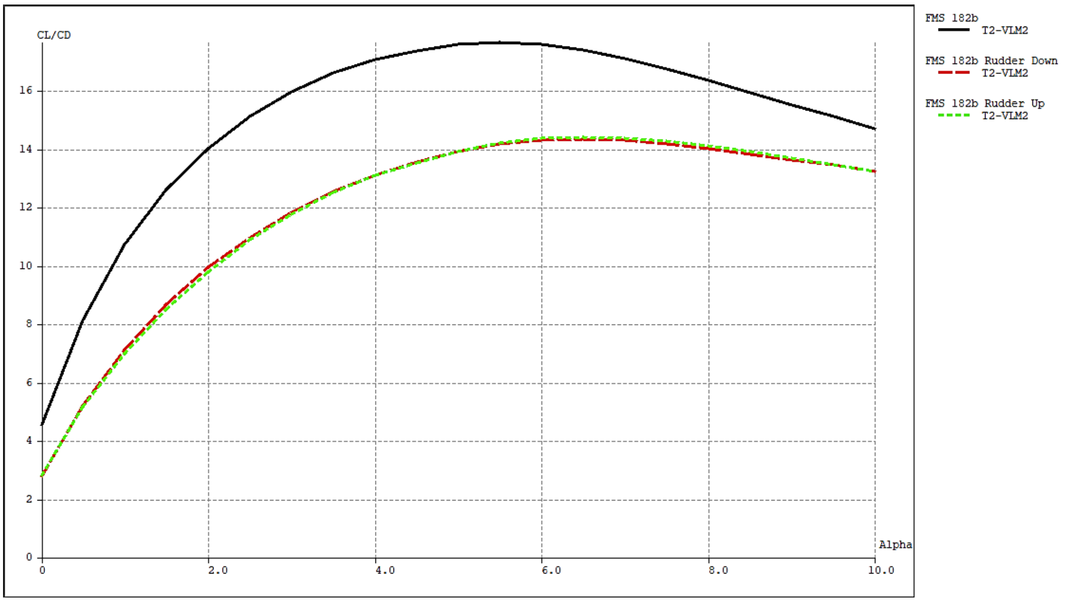


Figure 4.28 Comparison of CL/CD against Alpha Alpha of FMS 182b (black) and Rudder Down (red)/Rudder Up (green) models

As can be seen from Figure 4.28, the aerodynamic efficiency of the aircraft is expected to drop with the application of the rudder. The L/D ratio falls to 14.28 at alpha 6 and v = 11.61 m/s.

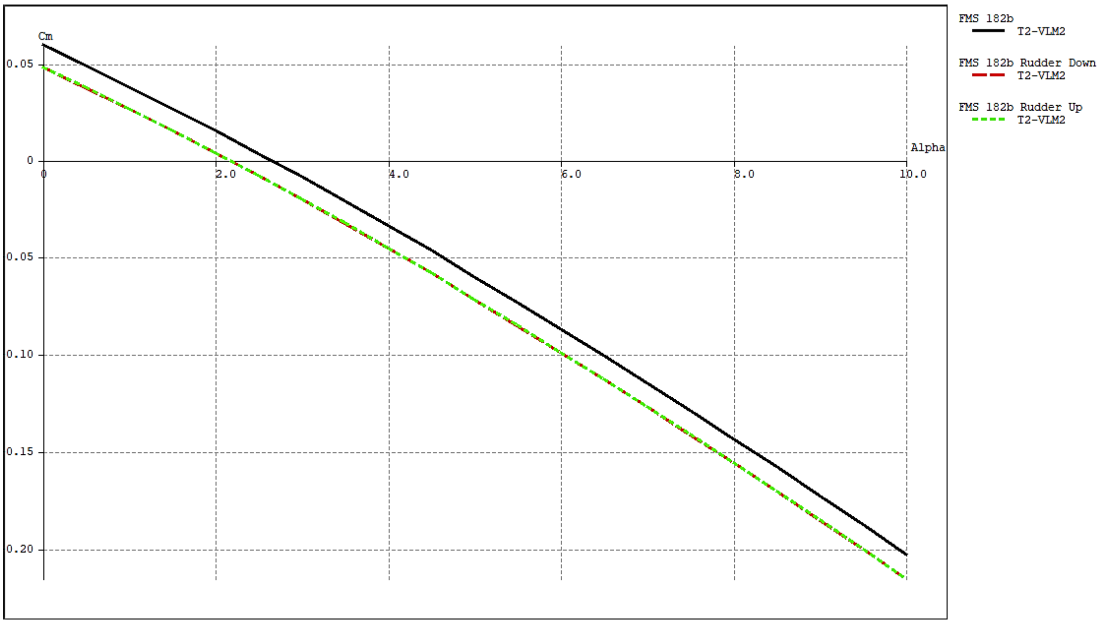


Figure 4.29 Comparison of Cm against Alpha Alpha of FMS 182b (black) and Rudder Down (red)/Rudder Up (green) models

The additional drag (away from the pitch axis) from the application of the rudder (max deflections) causes a slight change in the pitching moment as it can be seen in Figure

4.29. This shifts the zero moment trim condition to 2 degrees alpha (v = 18.79 m/s). Otherwise the desired negative slope of the curve is preserved.

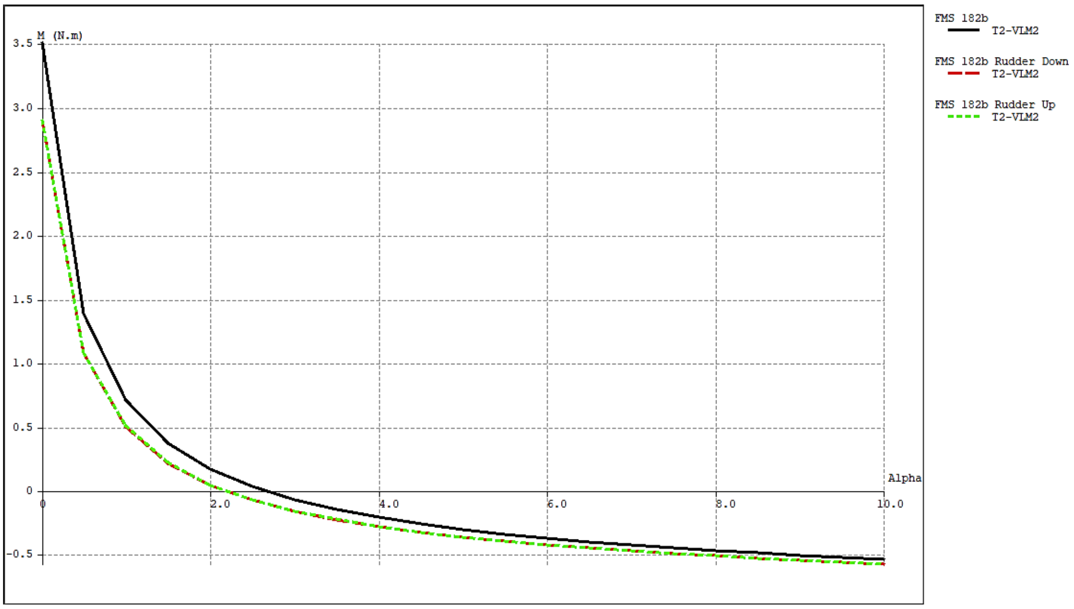


Figure 4.30 Comparison of pitching moment (M) against alpha Alpha of FMS 182b (black) and Rudder Down (red)/Rudder Up (green) models

The base model’s pitching moment curve results from the incidence angle of the horizontal stabiliser. The leftward shift of the curve observed for the rudder deflection models results from the additional off-axis drag.

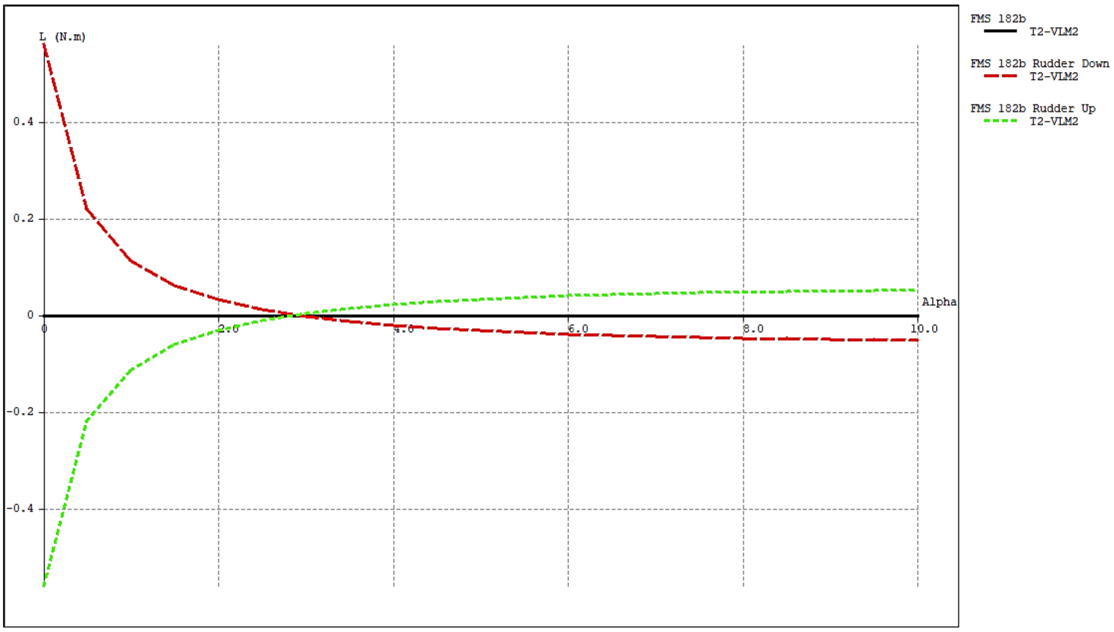


Figure 4.31 Comparison of rolling moment (L) against Alpha Alpha of FMS 182b (black) and Rudder Down (red)/Rudder Up (green) models

As it can be seen Figure 4.31, there is no rolling moment for the base model. For the rudder deflections however, a slight induced rolling moment is observed both above and below alpha = 3 degrees. This can be seen in the following flow visualisation results section. The rudder deflections result in asymmetric lift production on the horizontal stabiliser resulting in this slight roll moment.

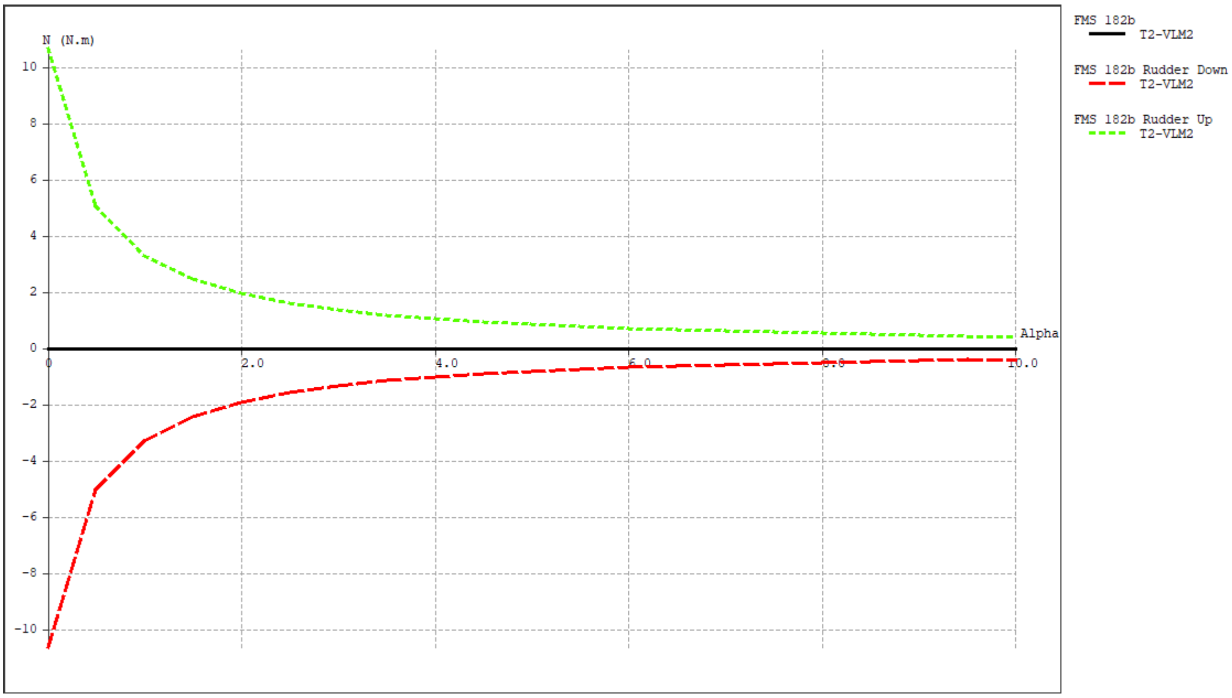


Figure 4.32 Comparison of yawing moment (N) against Alpha Alpha of FMS 182b (black) and Rudder Down (red)/Rudder Up (green) models

As expected, the largest moment produced by the rudder deflections are reflected on the yawning moment curves. They are symmetrical for both the positive and negative deflections and have the identical magnitude. As with all the other control surfaces, the decreased speed at higher alphas result in reduced control authority (production of less moment).

4.1.4 Performance Analysis Visualisation

The following sections present excerpts from the performance analysis conducted on the generated model variants of the FMS 182 in XFLR5.

4.1.4.1 Base Model

The following results show the simulation outputs for the base model for certain areas of interest under certain conditions of interest.

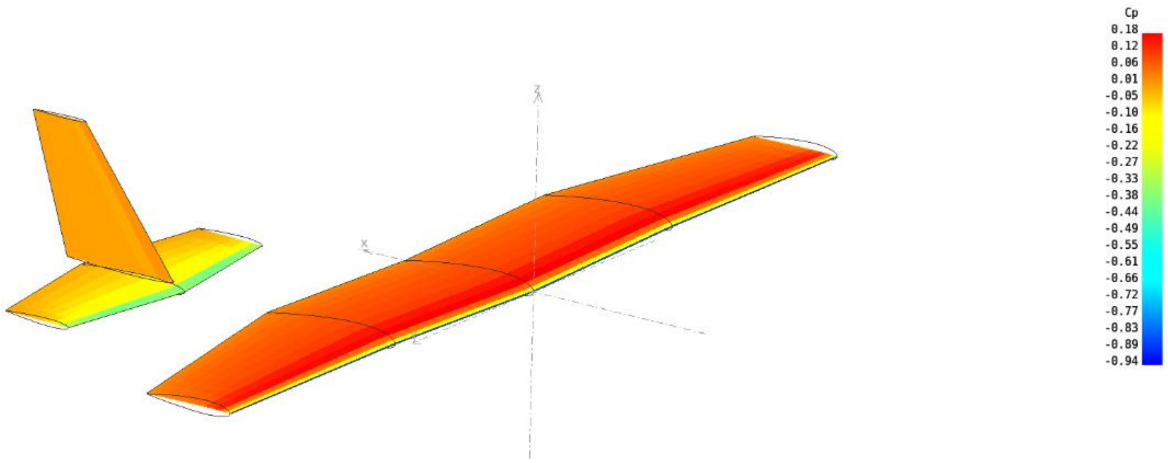


Figure 4.33 Pressure distribution at 0 degrees alpha

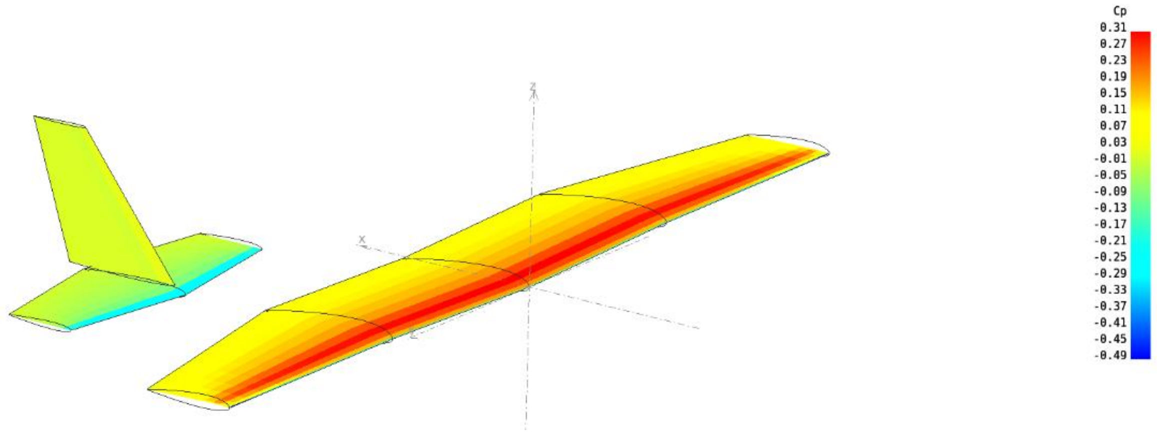


Figure 4.34 Pressure distribution at 1 degree alpha

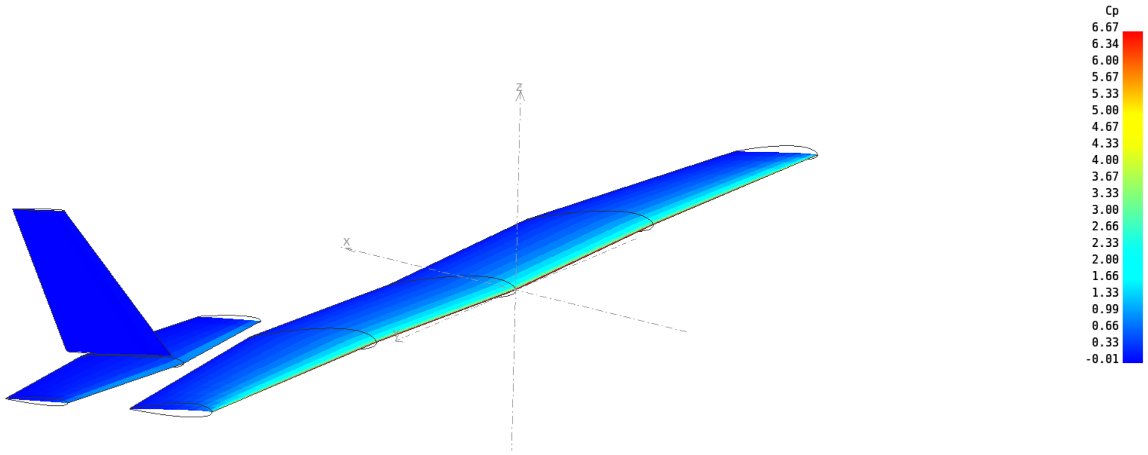


Figure 4.35 Pressure distribution at 10 degrees alpha

As Figure 4.33-4.35 show, the pressure distribution over the aircraft varies with the angle of attack (from zero to ten). At low angle of attack, the distribution is fairly even. This trend starts shifting with increased angle of attack until at high angle of attack the peak moves to the tip of the wing.

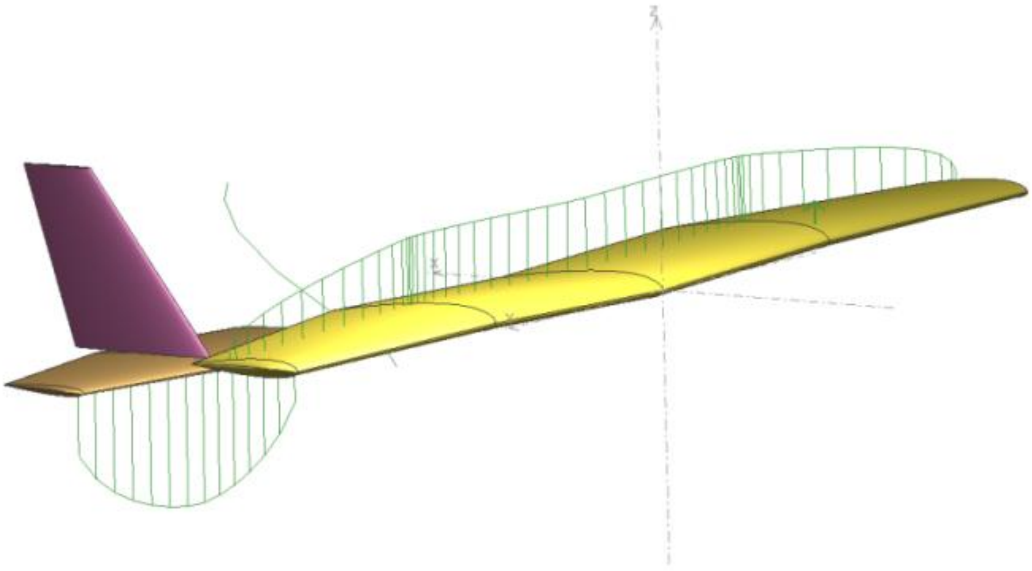


Figure 4.36 Lift at 0 degree alpha

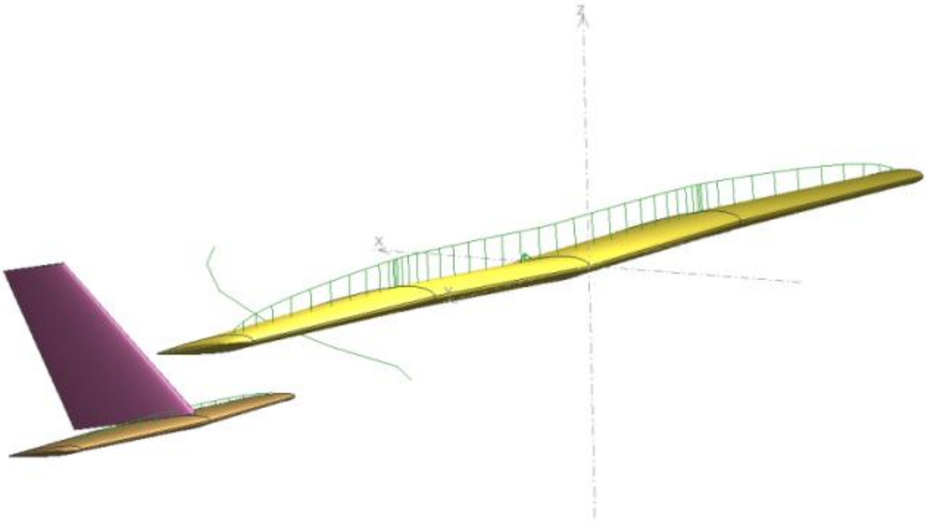


Figure 4.37 Lift at 10 degrees alpha

As seen in Figure 4.37, at low angle of attack, the lift produced by the horizontal stabiliser is in the opposite direction of the lift produce by the main wing. This helps maintain the pitching moment balance and keeps the aircraft stable in the pitch axis. At higher angles of attack (without any elevator input), the lift direction (horizontal stabiliser) reverses causing an opposing pitching moment which corrects and restores the incidence of the airplane.

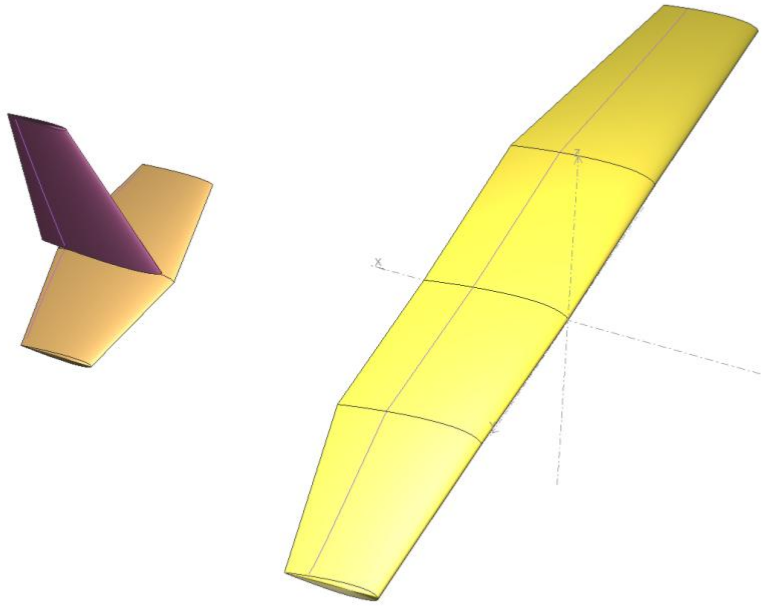


Figure 4.38 Boundary later at 0 degree alpha

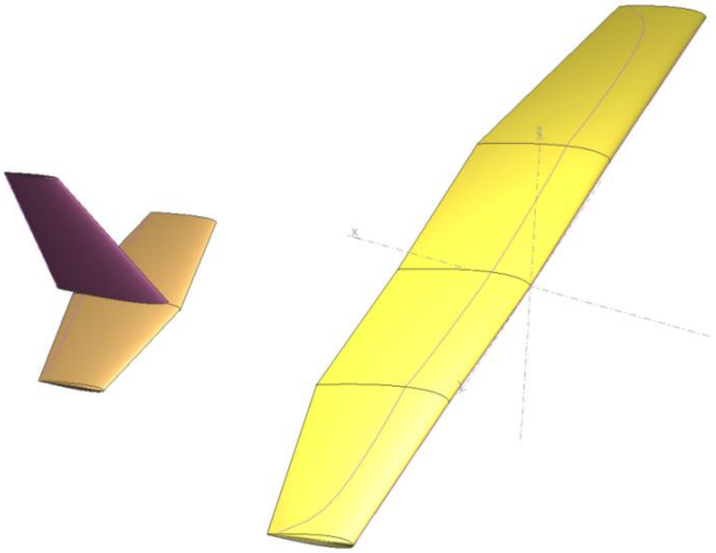


Figure 4.39 Boundary later at 10 degree alpha

As it can be seen from the above visualisations of the transition of the flow over the wing, at lower angle of attack the flow transition from laminar to turbulent happens closer to the trailing edge. But as the angle of attack is increased, additional turbulence in the boundary layer leads to the transition point shifting toward the leading edge.

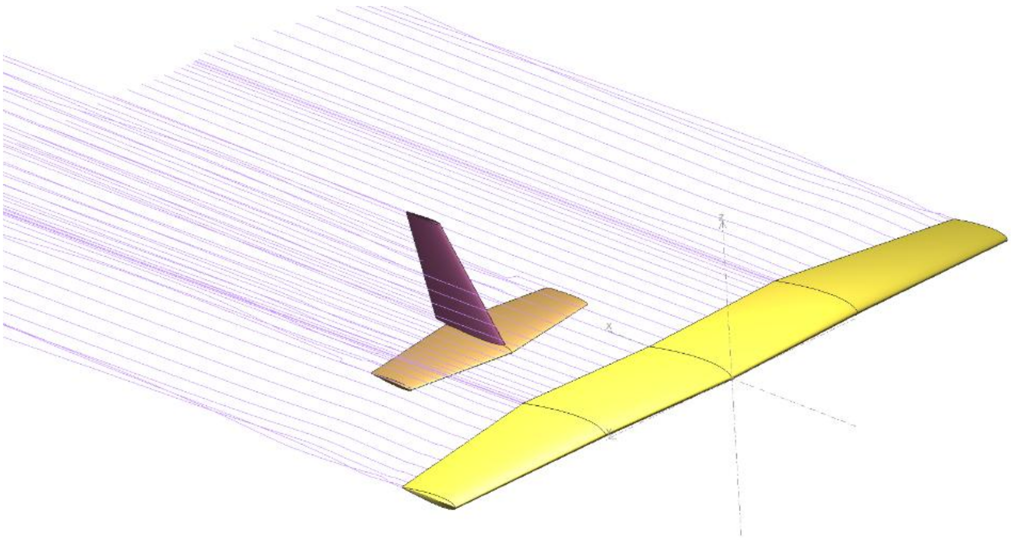


Figure 4.40 Flow streamlines at zero degree alpha

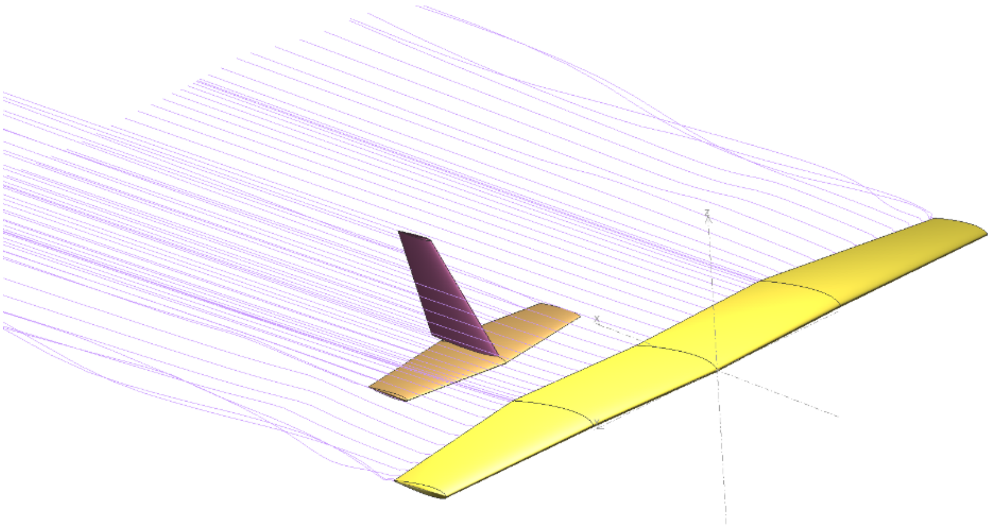


Figure 4.41 Flow streamlines at 3.5 degrees alpha

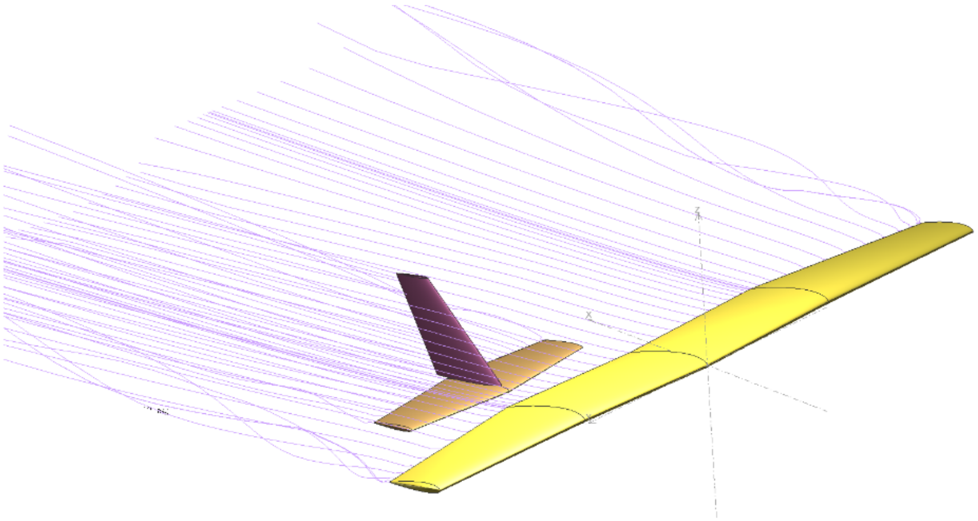


Figure 4.42 Flow streamlines at 10 degrees alpha

Figure 4.40-4.42 shows the development of the wing tip vortices as the angle of attack is increased. This results from an increase of the wing's lift coefficient with increased alpha. The low pressure region on top promotes a suction effect which effectively causes the higher pressure air from the bottom to side-slip along the bottom surface (spanwise) and rotate about the wingtip as it gets pulled towards the top surface. The energy that goes into creation of these vortices is experienced by the aircraft as part of its overall drag (lift induced drag). Although the amount of this drag remains proportional to the amount of lift, the relative size/magnitude of these vortices will change with variations in the angle of attack.

4.1.4.2 Flaps Down

The following results have been obtained from the model variant containing the flaps. The following visualisations capture the predicted performance for the flight situation where the flaps are fully deployed.

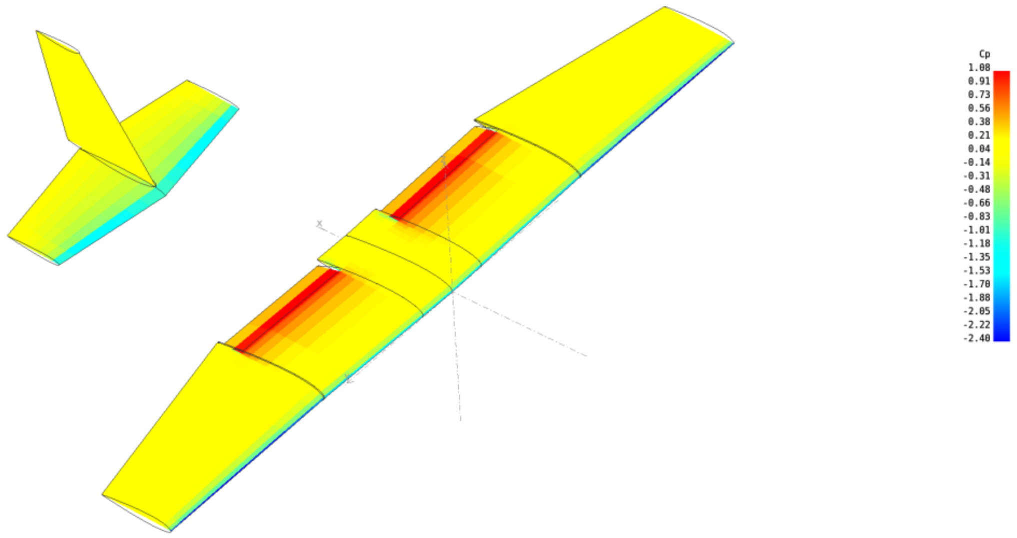


Figure 4.43 Pressure distribution at -3 degrees alpha

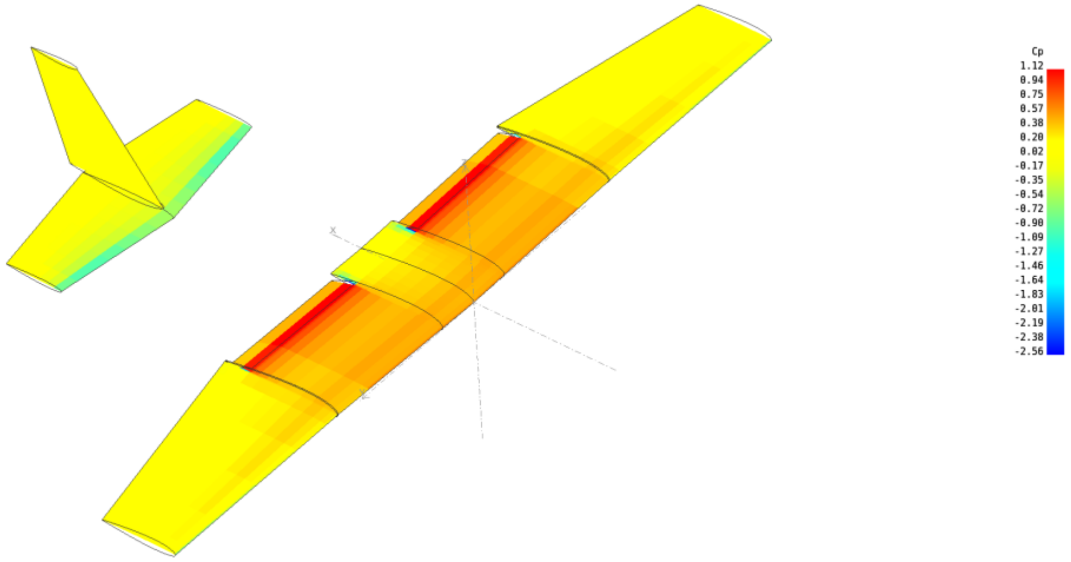


Figure 4.44 Pressure distribution at 0 degrees alpha

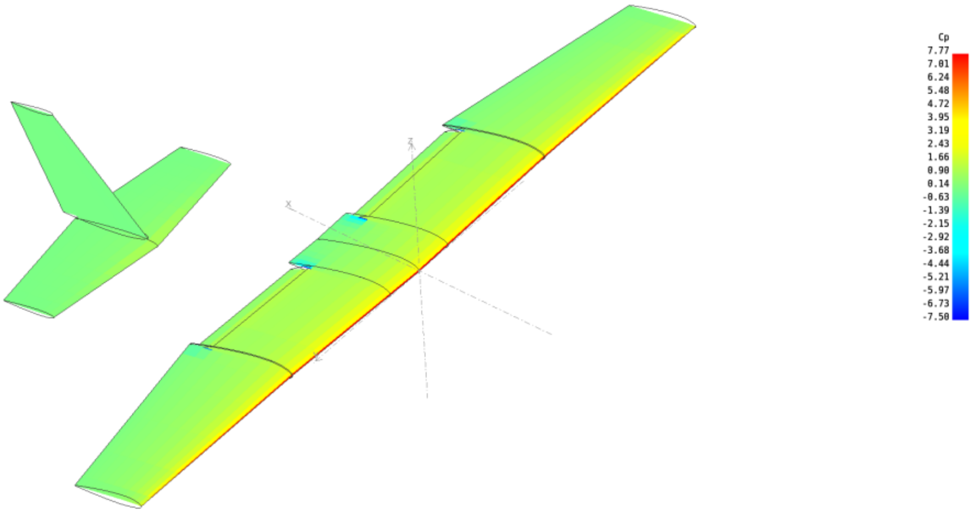


Figure 4.45 Pressure distribution at 10 degrees alpha

As it can be seen from the pressure distribution visualisation for increasing alpha, the deployment of the flaps results in significant changes to the pressure distribution over the wing sections. At low alpha, the highest pressure regions are concentrated around the flaps. This gradient gradually moves toward the leading edge of the wing with increased alpha until it concentrates at the leading edge of the wing at high angles of attack.

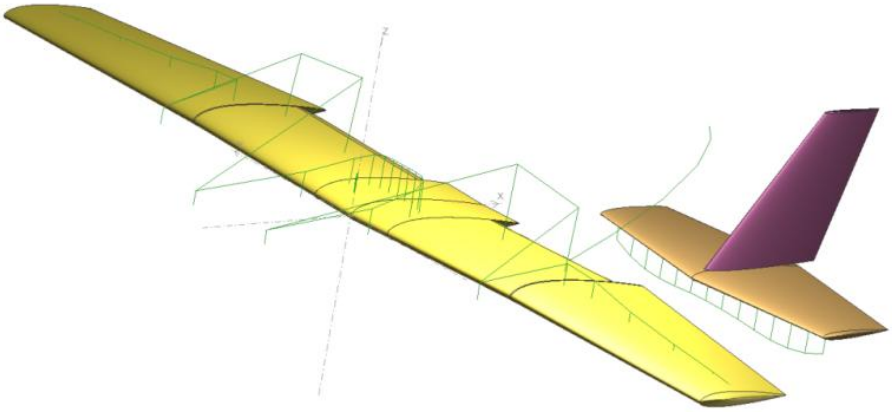


Figure 4.46 Lift distribution at 0 alpha

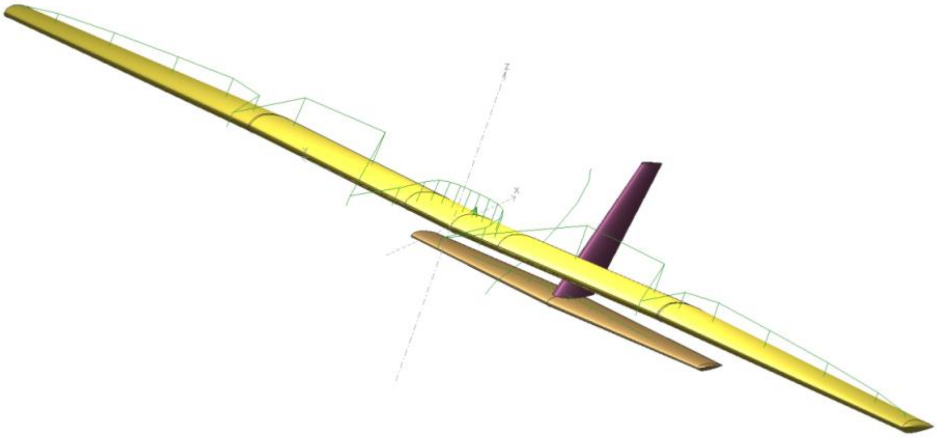


Figure 4.47 Lift distribution at 10 degrees alpha

The above lift distribution visualisations show the expected pattern from low to high angle of attack when the flaps are fully deployed. It should be noted that a software bug results in this visualisation being rendered in the wrong area. However, it’s only shifted back in the x-axis and as such, the output is still meaningful. As it can be seen, the lift distribution has a greater magnitude in the low alpha while that drops as the angle of attack is increased. The highest magnitude is observed over the wing section using the aerofoil and wing modelling for the flaps.

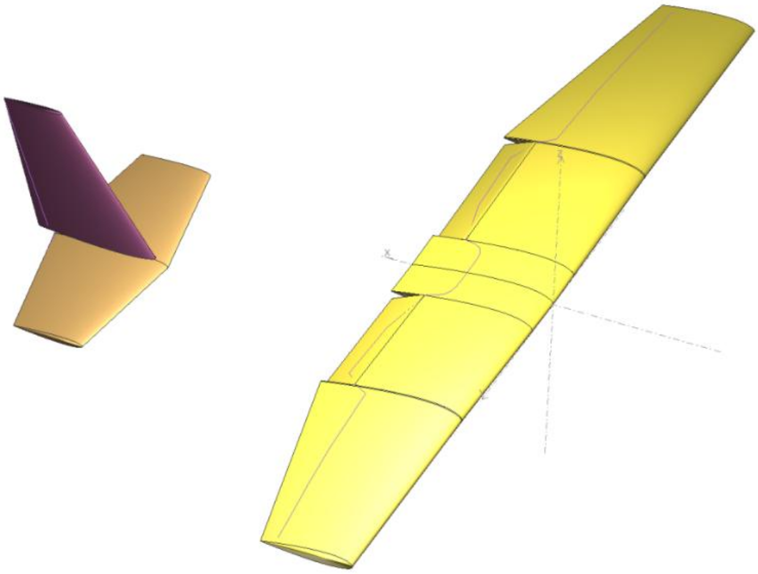


Figure 4.48 Boundary layer at 0 degrees alpha

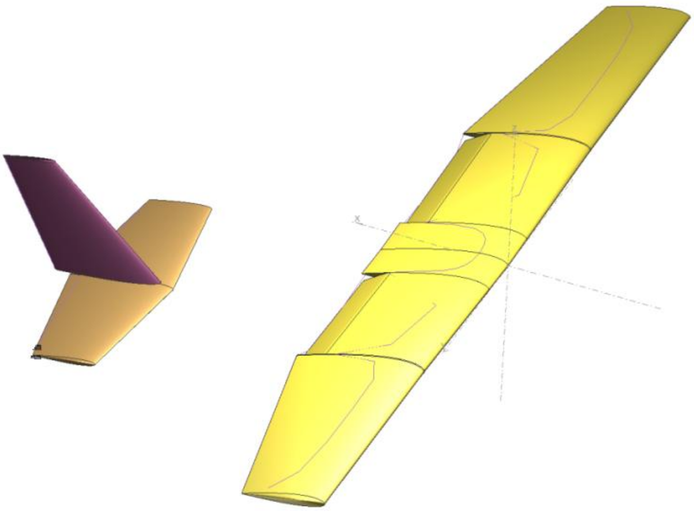


Figure 4.49 Boundary layer at 10 degrees alpha

Figure 4.48-4.49 shows the movement of the transition point with flaps deployed. As it can be seen, the same behaviour of the transition point moving from the trailing edge to the leading edge is observed for increasing angle of attack even with flaps deployed.

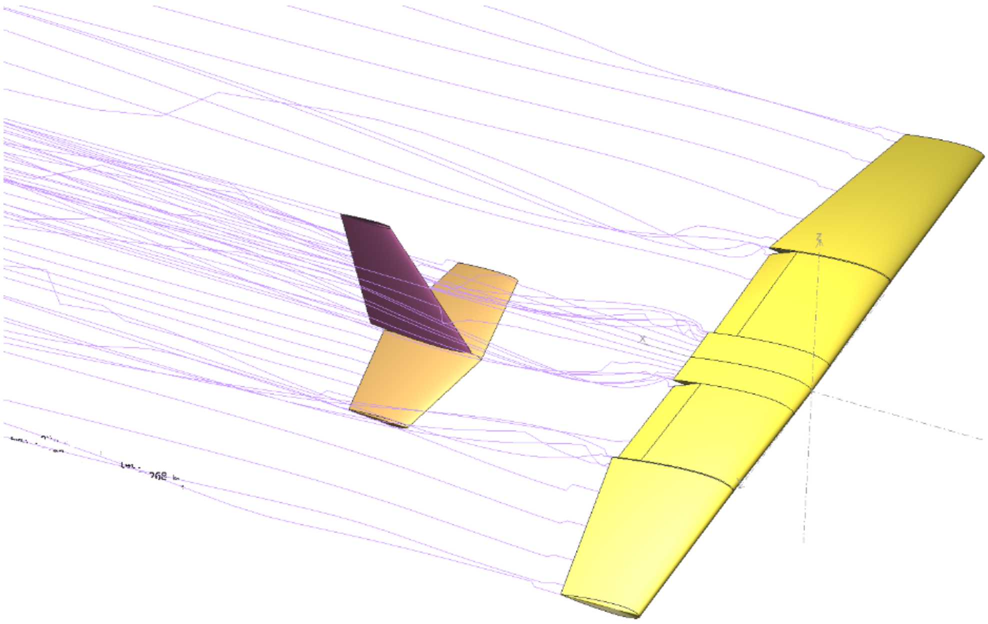


Figure 4.50 Flow streamlines at 0 degrees alpha

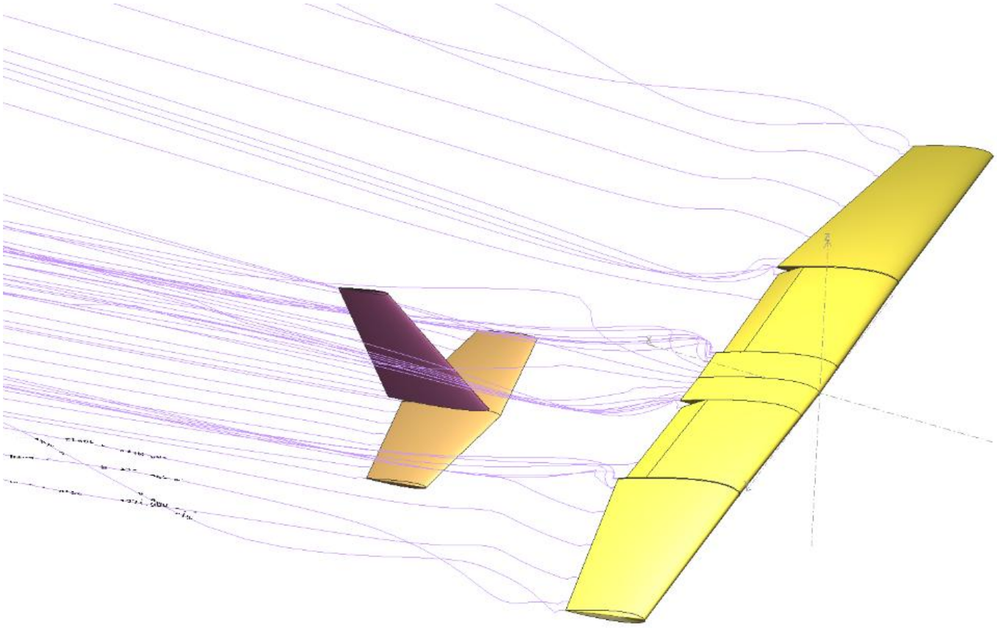


Figure 4.51 Flow streamlines at 10 degrees alpha

Figure 4.50-4.51 shows the disruption caused by the main wing to the free stream for increasing angle of attack. As it can be seen, there is an increased amount of vortex formation resulting from the deployment of the flaps. These disruptions are also observed to interfere with the airflow over the vertical stabiliser. As explained previously, this will be associated with increased drag and some instability due to the transient nature of tip vortices. Inconsistency in the vortex formation may cause localised stall of the vertical stabiliser resulting in reduced control authority. However, this is expected to be mitigated by the size of the vertical stabiliser/rudder (most of which is outside of this influence). Therefore the relatively more significant side effect of any unstable vortices is likely to be oscillations in the yaw direction.

4.1.4.3 Aileron Model:

The following results for flow visualisation have been obtained by using the model variant with the ailerons. Since the results are symmetric, only one configuration is presented here.

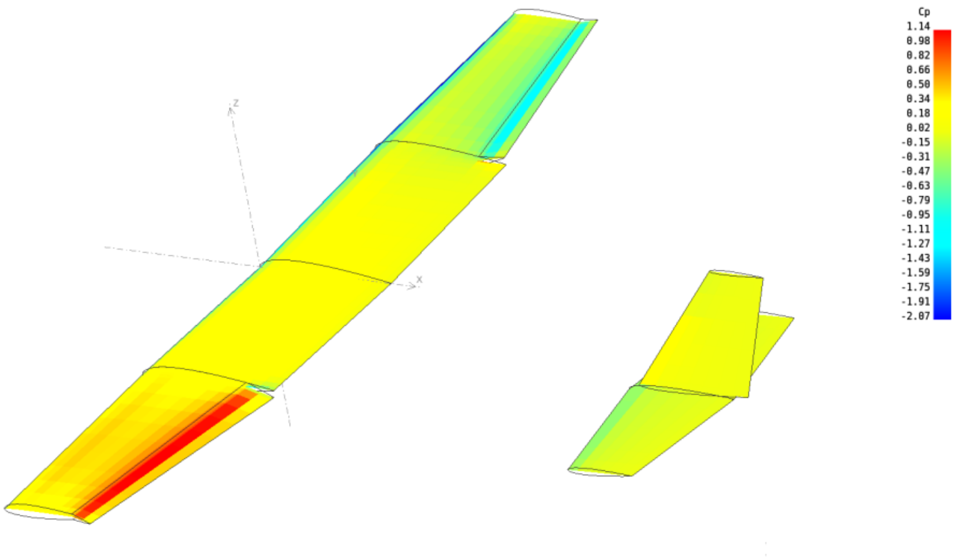


Figure 4.52 Pressure distribution at 0 degrees alpha

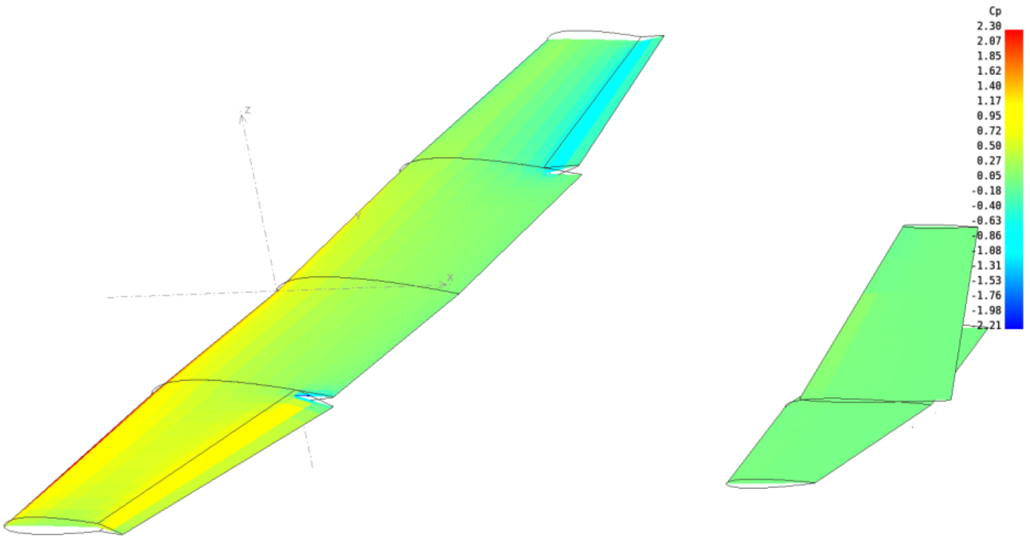


Figure 4.53 Pressure distribution at 3 degrees alpha

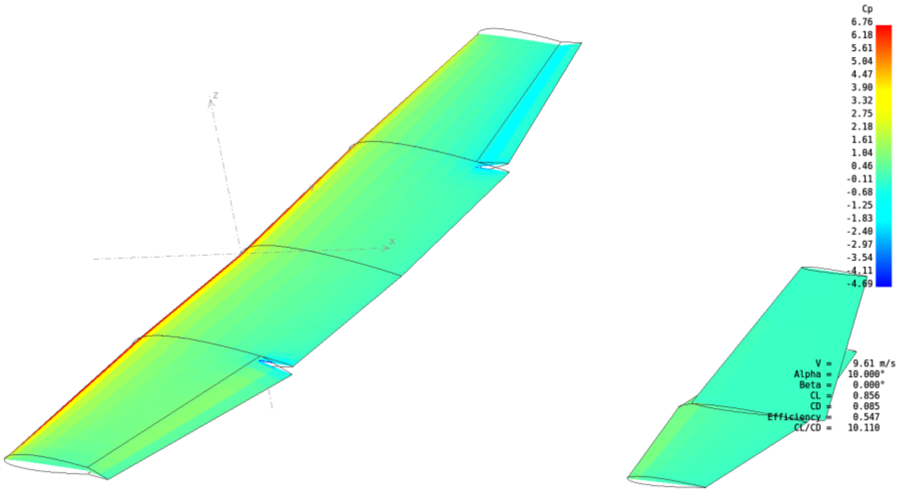


Figure 4.54 Pressure distribution at 10 degrees alpha.

The resultant pressure distribution shift over the wings is consistent with expectation and the lift distribution simulation results. As alpha is increased, this increased pressure gradually moves from over the aileron section toward the leading edge of the wing.

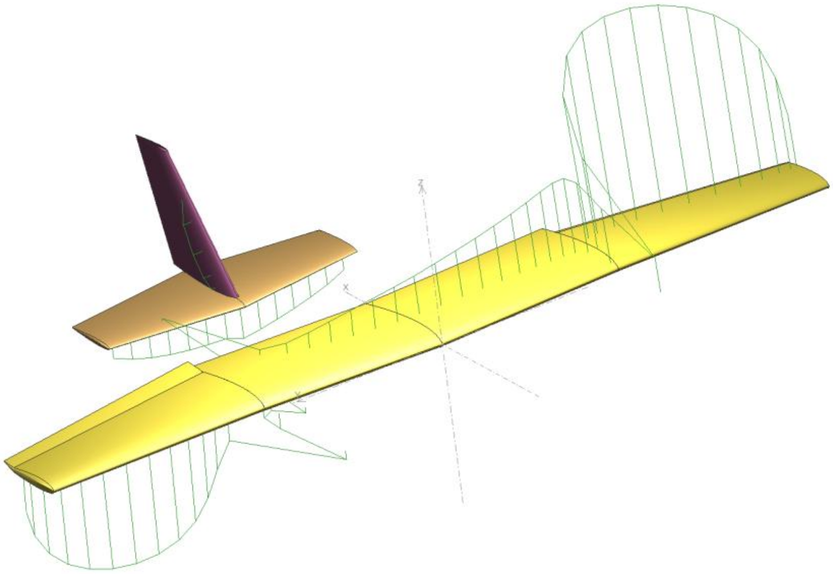


Figure 4.55 Lift distribution at 0.5 degrees alpha

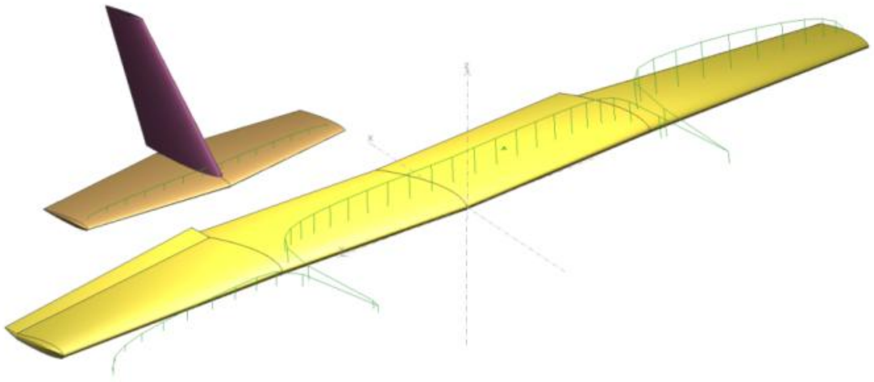


Figure 4.56 Pressure distribution at 10 degrees alpha

The above visualisation (Figure 4.55-4.56) represents the lift distribution over the main wing as a function of angle of attack for the model variant with full aileron deflections. Notable observations are that at lower angle of attack the control surface effectiveness is much greater (owing to the higher airspeed) and results in higher localised lift (over the ailerons) causing greater rolling moment . This effect is observed to gradually decrease with increased angle of attack as the airspeed drops. The centre of pressure also shifts toward the leading edge with the increase in angle of attack. It should be noted that the spikes observed along the points where the wing sections are visible should be ignored as this is merely a simulation artefact resulting from discontinuities in the panel grid where two different grids/aerofoil sections meet.

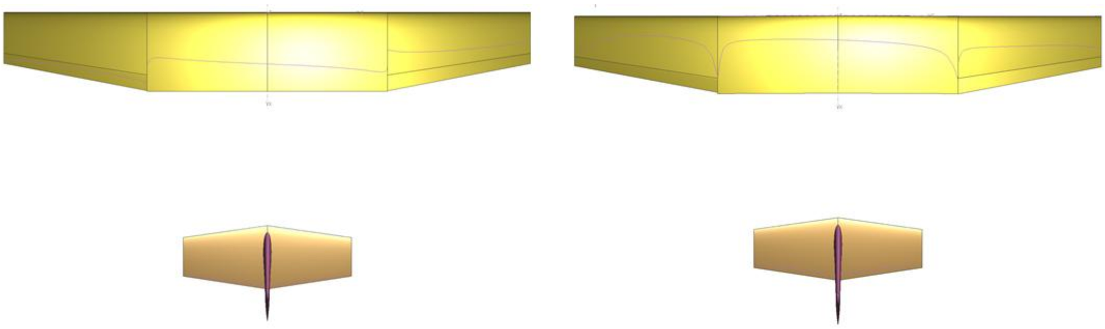


Figure 4.57 Boundary layer at 0 degrees(left) and 10 degrees(right) alpha

As Figure 4.57 show, aileron deflections at increasing angle of attack still results in the transition point of the flow over the wing surface shifting. The behaviour of the transition point is consistent with expectation.

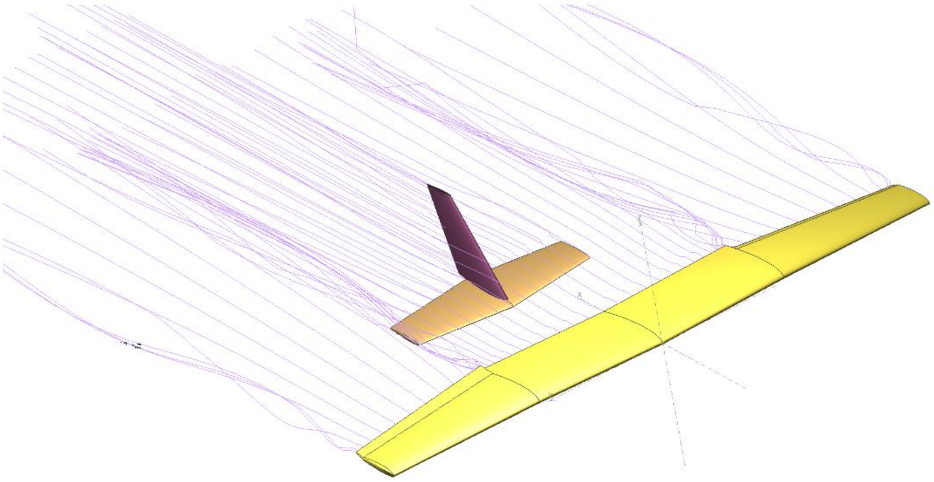


Figure 4.58 Flow streamlines at 0 degrees alpha

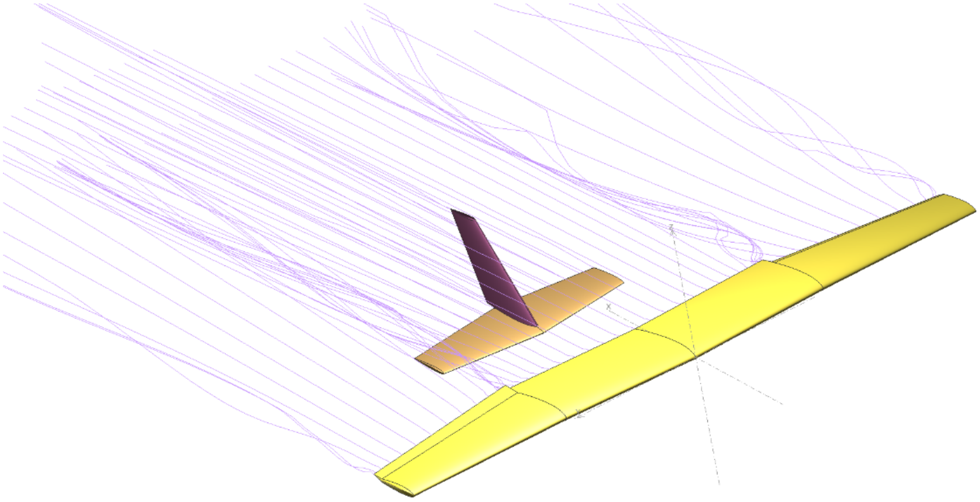


Figure 4.59 Flow streamlines at 3 degrees alpha

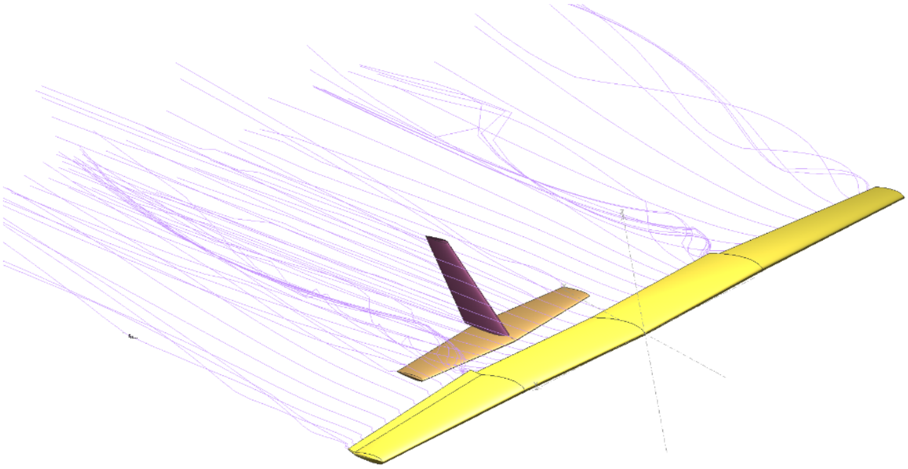


Figure 4.60 Flow streamlines at 10 degrees alpha

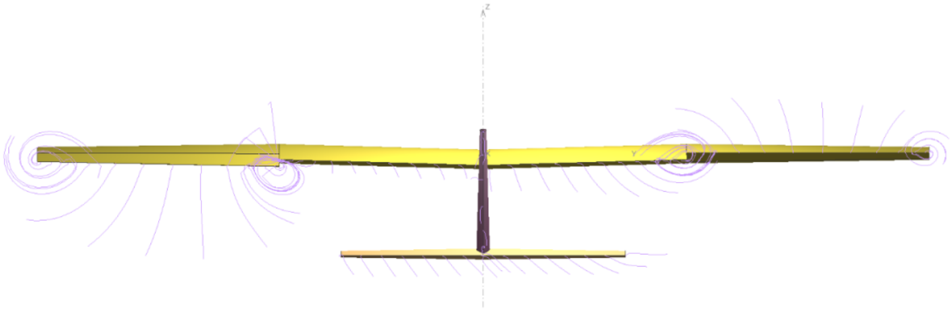


Figure 4.61 Flow streamlines at 3 degrees alpha (rear view)

As Figure 4.58-4.61 shows, the application of the ailerons is expected to produce tip vortices resulting in significant disruption to the stream and additional drag. For increasing angle of attack, there is more unsteady and bigger vortex formation observed. Even then, at high angle, this disruption to the free stream is not big enough to significantly interfere with the tailplane.

4.1.4.4 Elevator Down:

The following results have been obtained based on the three model variants with the maximum elevator deflection (15 degrees) neutral, upward and downward.

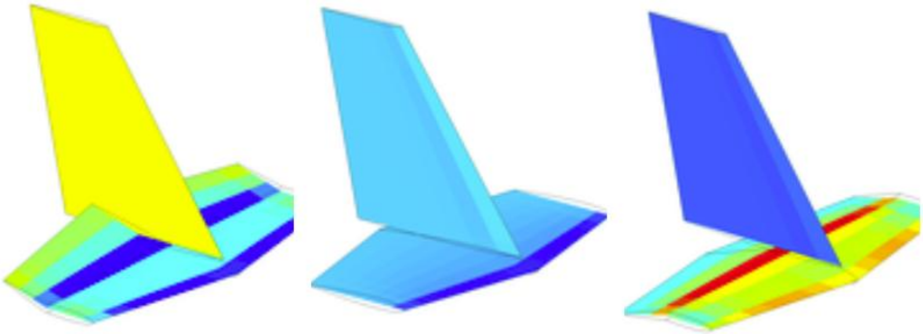


Figure 4.62 Pressure distribution for elevator up (left), elevator 0 and elevator down at 2 degrees alpha

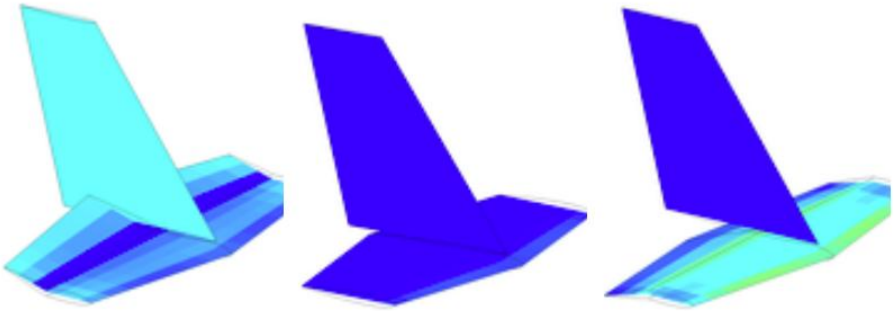


Figure 4.63 Pressure distribution for elevator up (left), elevator 0 and elevator down at 5 degrees alpha

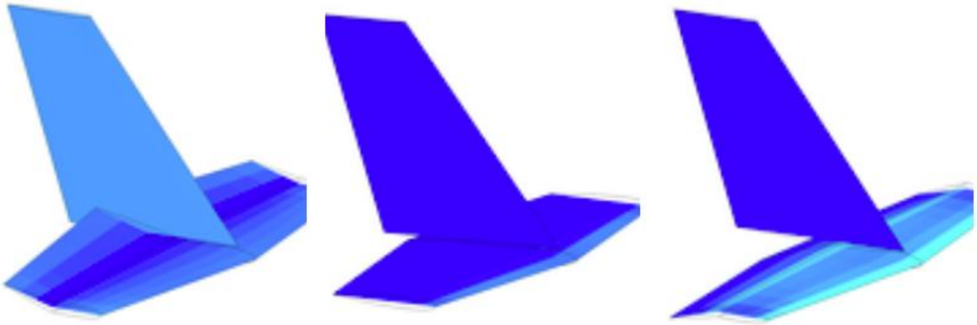


Figure 4.64 Pressure distribution for elevator up (left), elevator 0 and elevator down at 9.5 degrees alpha

As Figure 4.62-4.64 show, the overall pressure distribution over the elevator section drops with increasing angle of attack. However, even at lower angle of attack, elevator deflections are predicted to increase the resultant pressure distribution in the positive direction.

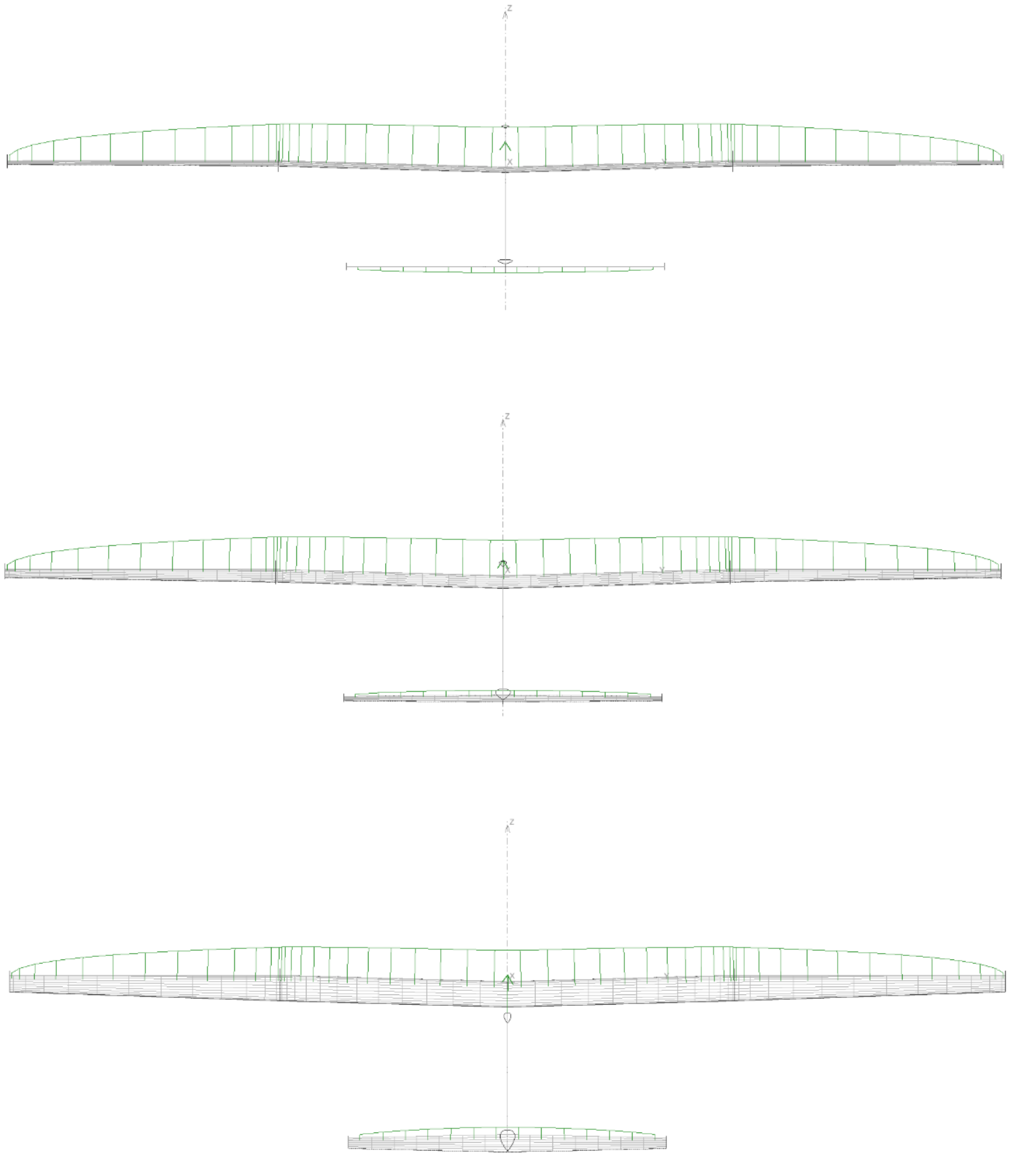


Figure 4.65 Lift distribution for elevator centred at 0, 5 and 10 degrees alpha

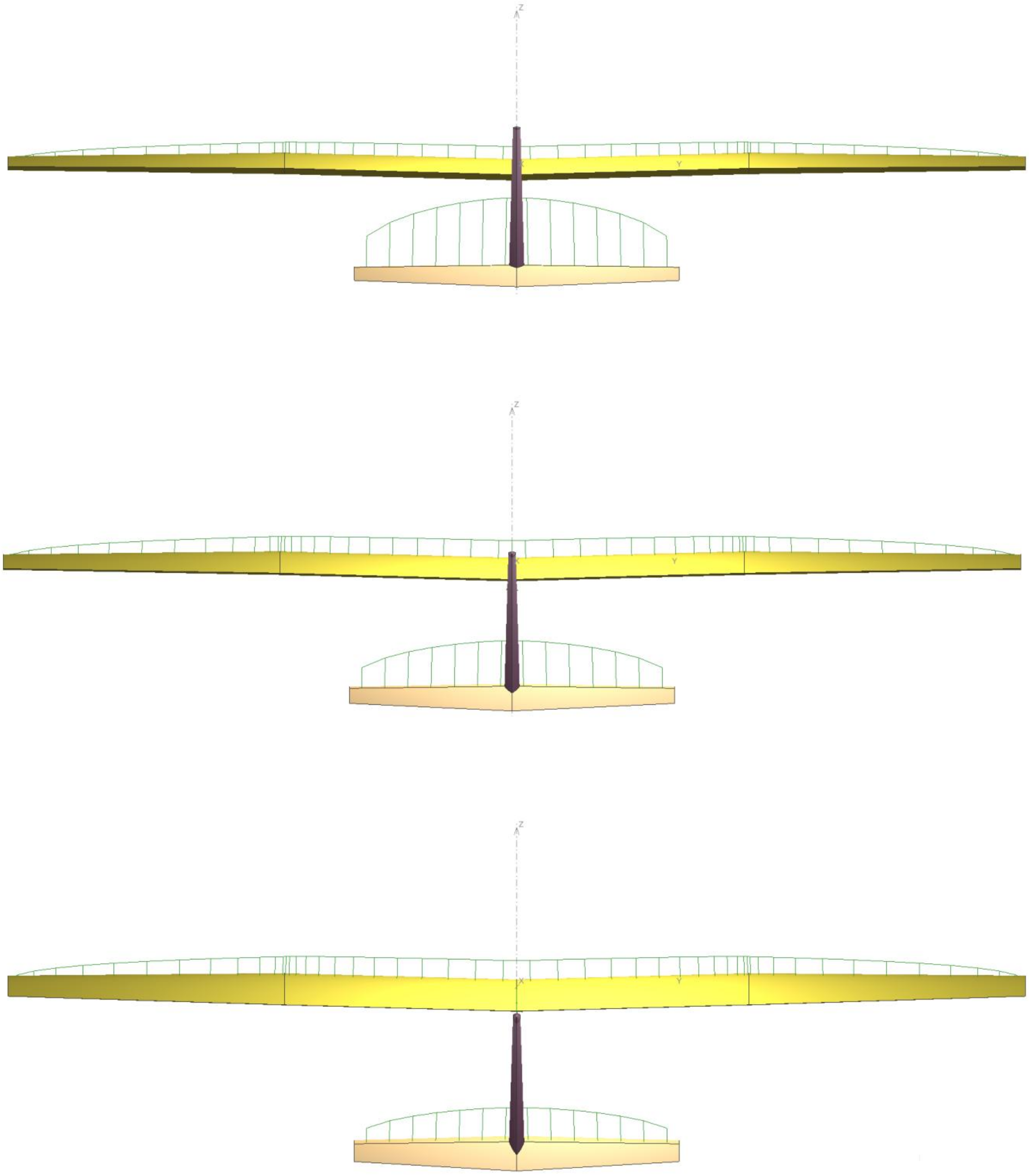


Figure 4.66 Lift distribution variation for elevator down at 0, 5 and at 9.5 degrees alpha

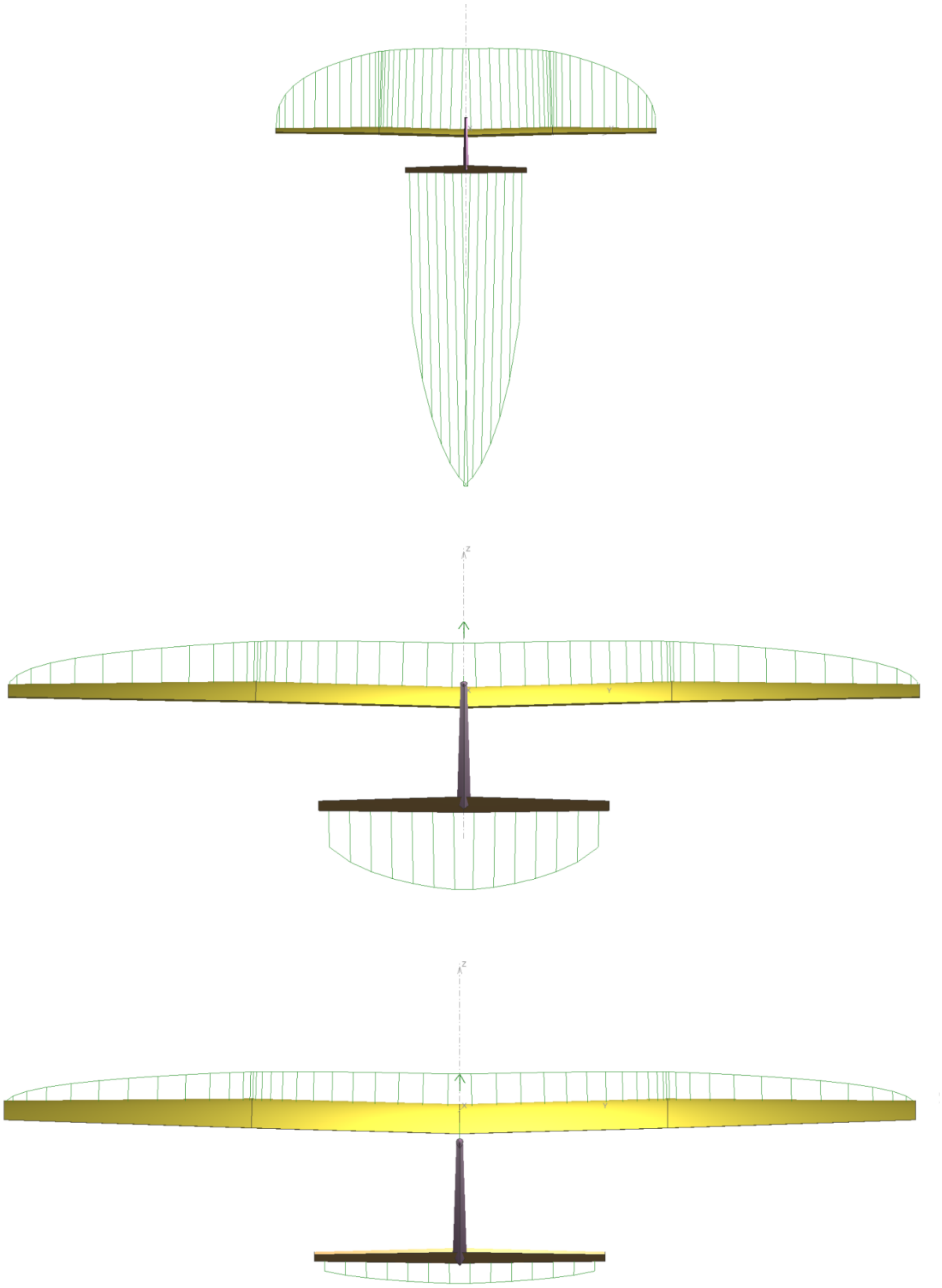


Figure 4.67 Lift distribution of elevator up at 0, 5 and 10 degrees alpha

Figure 4.65-4.67 show the variations in the lift distribution across an increasing range of angle of attack (2, 5, 9.5) for the base model compared to the elevator up and down models. In every image, the top one is low alpha and the bottom one is high alpha. For the base model, for low angle of attack, the neutral elevator creates a slight pitch up moment which changes direction by the time the alpha = 5 due to the trim condition of approximately 2.75 degrees. This trend is carried forward at 10 degrees where the lift of the horizontal stabiliser is more obvious in the visualisation as it has a much greater magnitude. For the elevator down model, the control authority is great enough such that in any valid flight condition, significant positive lift is generated by the horizontal stabiliser. This results in significant pitch down moment. Likewise when the elevator is deflected in the opposite direction, it has a great enough control authority to keep the entire horizontal stabiliser in negative lift throughout the test envelope resulting in a very consistent pitch up moment. The top image for the elevator up model contains an unrealistic prediction (excessive downforce) due to the associated flight speed of ~45 m/s which is way outside of any planned operations.

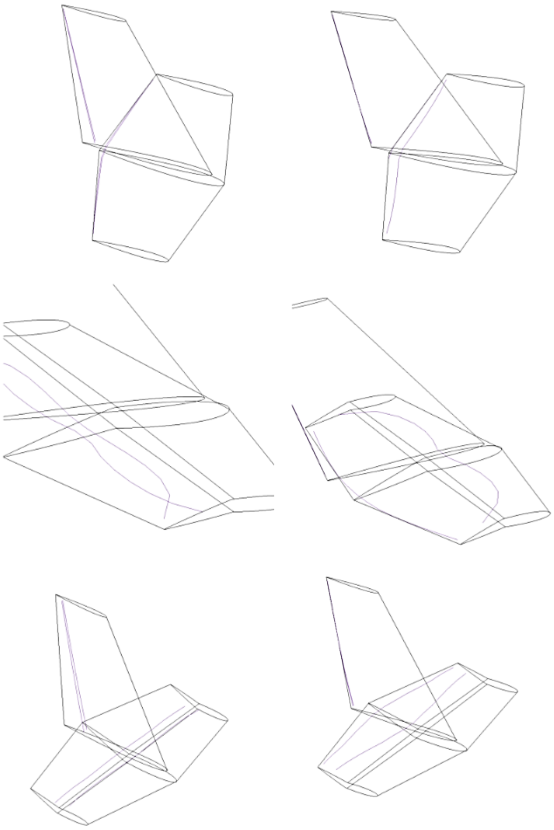


Figure 4.68 Boundary layer transitions for elevator centred, down and up (top to down)

For the base model, the transition point at the horizontal stabiliser moves a little bit toward the leading edge of the section with increasing alpha. However, for the elevator up/down models, this transition point shifts significantly for increasing alpha. For the elevator down model, the transition point moves closer to the centreline until the top section’s transition point crosses that line and approaches the leading edge while the bottom section’s transition point moves back closer to the trailing edge. For the elevator up model, at lower angles of attack, similar transition point behaviour is observed.

However, at high angle of attack, the transition points for both the top and bottom surface moves toward the trailing edge.

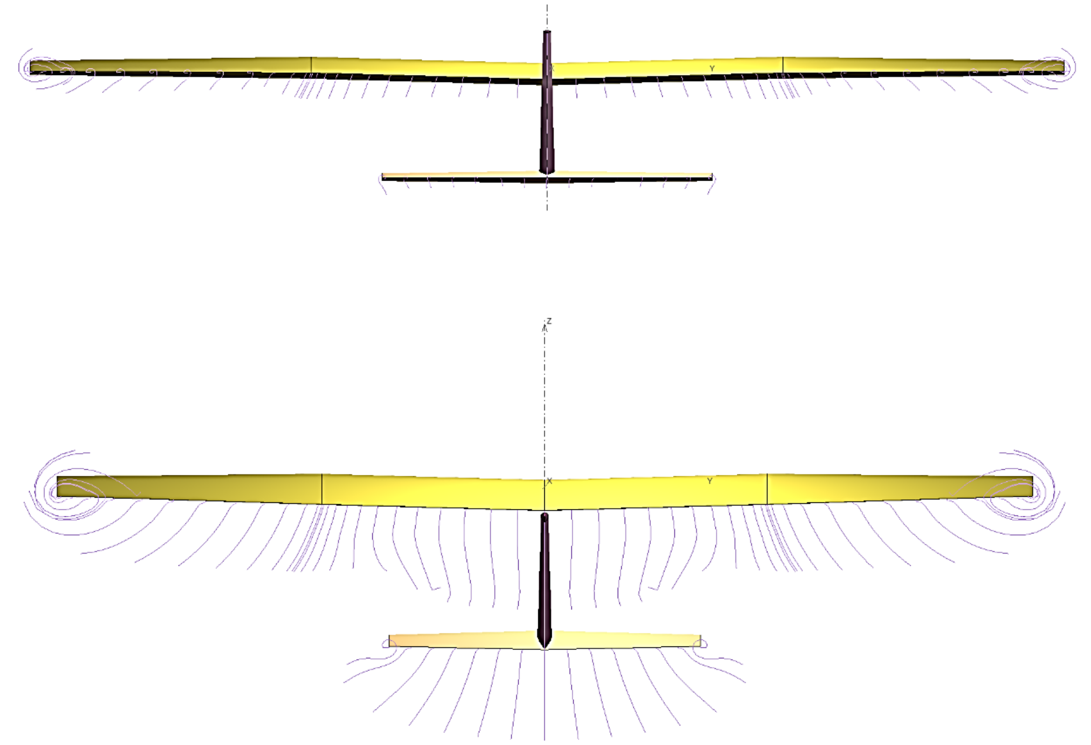


Figure 4.69 Neutral elevator (centred) flow streamlines

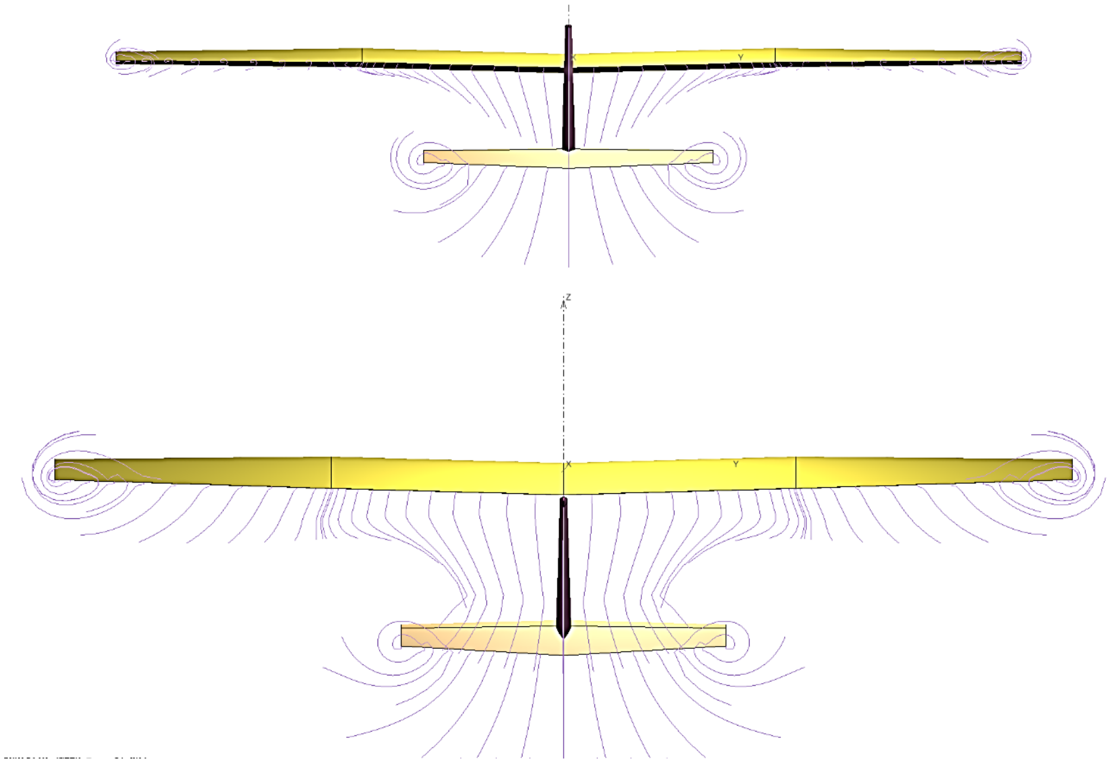


Figure 4.70 Elevator down flow streamlines

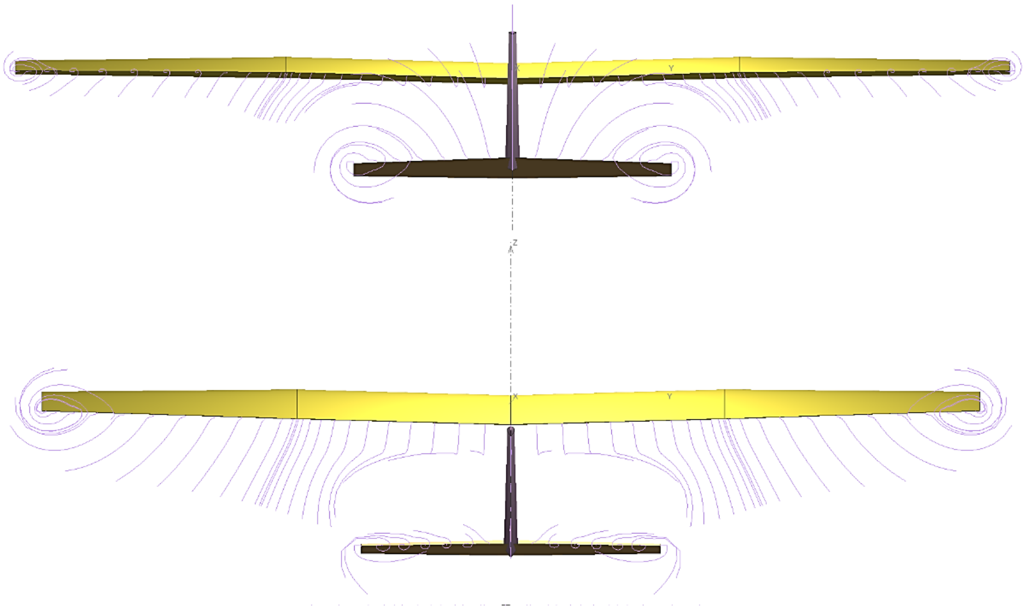


Figure 4.71 Elevator up flow streamlines

With the elevator neutral, at different angles of attack, vortex formation is generally localised to the tips of the wing and is fairly small. However, the predictions based on the full elevator deflection models show the formation of significant tip vortices at the tips of the horizontal stabiliser. The only difference between the two sets of vortices for the up and down situation is the direction of spin and downwash and lift. This however does not alter the magnitude of the drag induced.

4.1.4.5 Rudder:

The following results have been obtained by using the model variant representing full rudder deflection at the vertical stabiliser. Since they are symmetric, only one is used for this illustration.

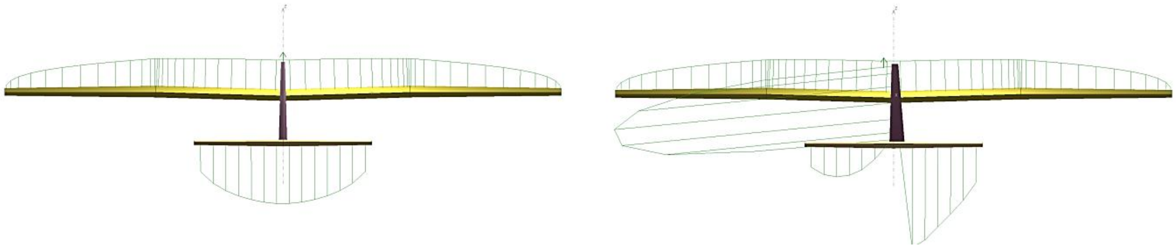


Figure 4.72 Lift distribution without rudder input (left) and with full rudder (right)

As we see in the above visualisation, the application of the rudder causes a great degree of change in the lift distribution over the model. The rudder up configuration results in net positive lift distribution over the surface of the vertical stabiliser in the port side. This is expected to lead to a yaw moment. It’s also interesting to notice that the disruption over the port-side section of the horizontal stabiliser leads to asymmetric lift on that section. This causes a small rolling moment as well as some disruption to the pitching moment.

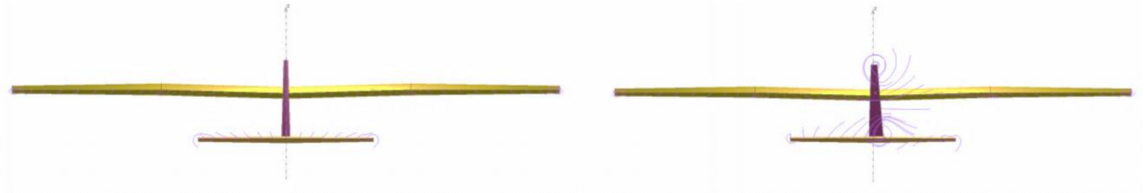


Figure 4.73 Flow streamlines (rear view) without(left) and with(right) rudder input .

As the above visualisation of the stream shows, tip vortices are expected to form at either ends of the rudder section. This is expected to produce it’s own induced drag.

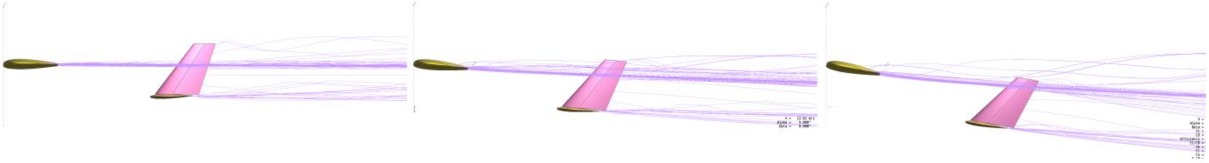


Figure 4.74 Rudder input streamlines side view (alpha increasing left to right)

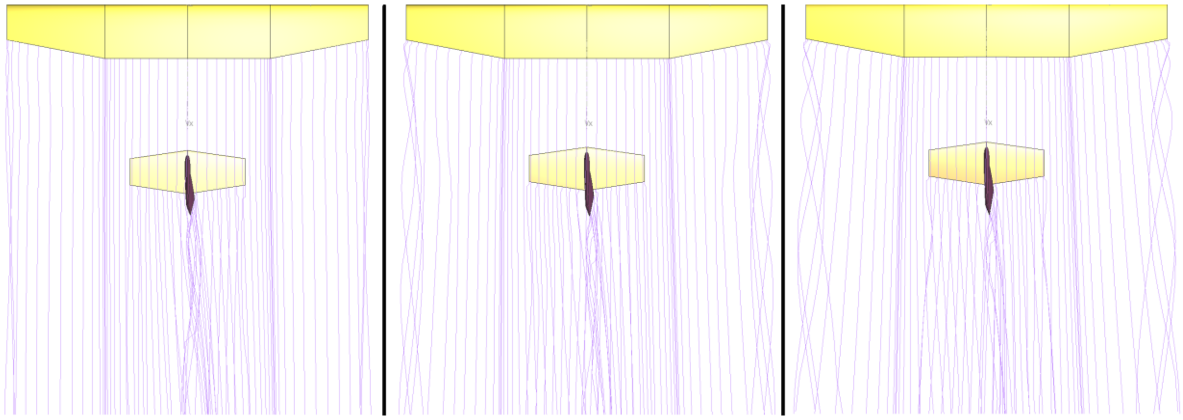


Figure 4.75 Rudder input streamlines top view (alpha increasing left to right)

From the flow visualisation side view, we can see the interference of the stream from the wing to the vertical stabiliser travels up the rudder section with increasing angle of attack. The side view unfortunately includes the wing tip vortex so the top view is presented to show that the rudder section does not experience as much interference from those additional stream lines seen in the side view. This is advantageous in situations such as landing where losing rudder authority may become a serious problem.

Interference in flapped situation:

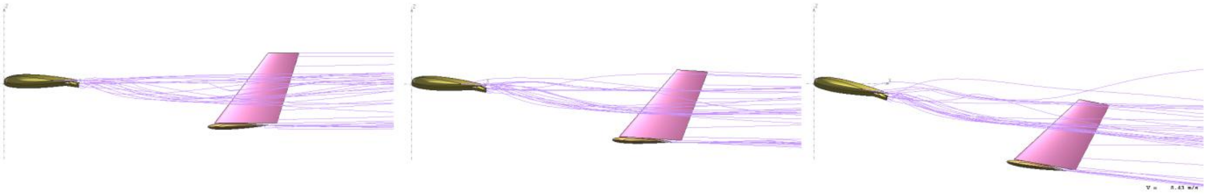


Figure 4.76 Rudder input streamlines side view (alpha increasing left to right)

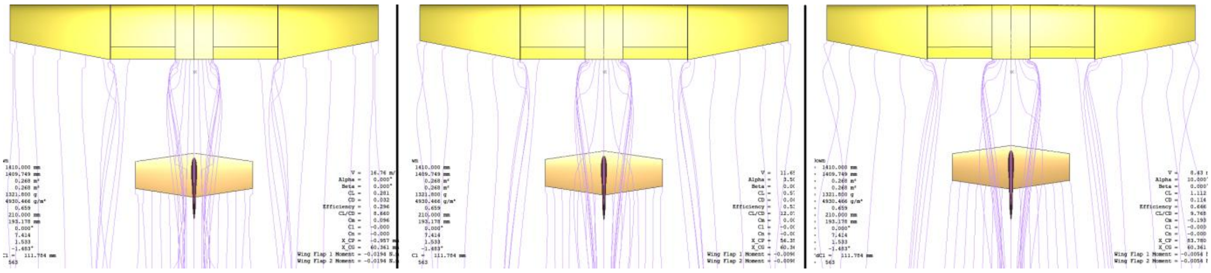


Figure 4.77 Rudder input streamlines top view (alpha increasing left to right)

The simulation output including the flaps predicts formation extra vortices of the inside of the flaps which interfere with the rudder section as shown in Figure 4.76-4.77. However, this visualisation does not factor in the presence of the fuselage which is essentially in contact with the flaps and will hinder the formation of such vortices. Furthermore, the fuselage of the FMS 182 blends smoothly from the trailing edge of the wing to the vertical stabiliser assuring a relatively smooth airflow in regular flight operations.

4.1.5 Prerequisites for Stability Analysis

4.1.6 Mass variation tests

The graph (Figure 4.79) shows the trend for the CL/CD against V for the two variants with incremental weight increase starting from 1321.8 grams and ending at 1921.8 grams. The overall pattern suggests that the flapped model reduces the overall aerodynamic efficiency (due to the additional drag from flap deployment). Incremental increase in weight (in this case by 50 gram increments) resulted in shifting the curves horizontally to the right, increasing the flight speed needed for optimal glide ratio.

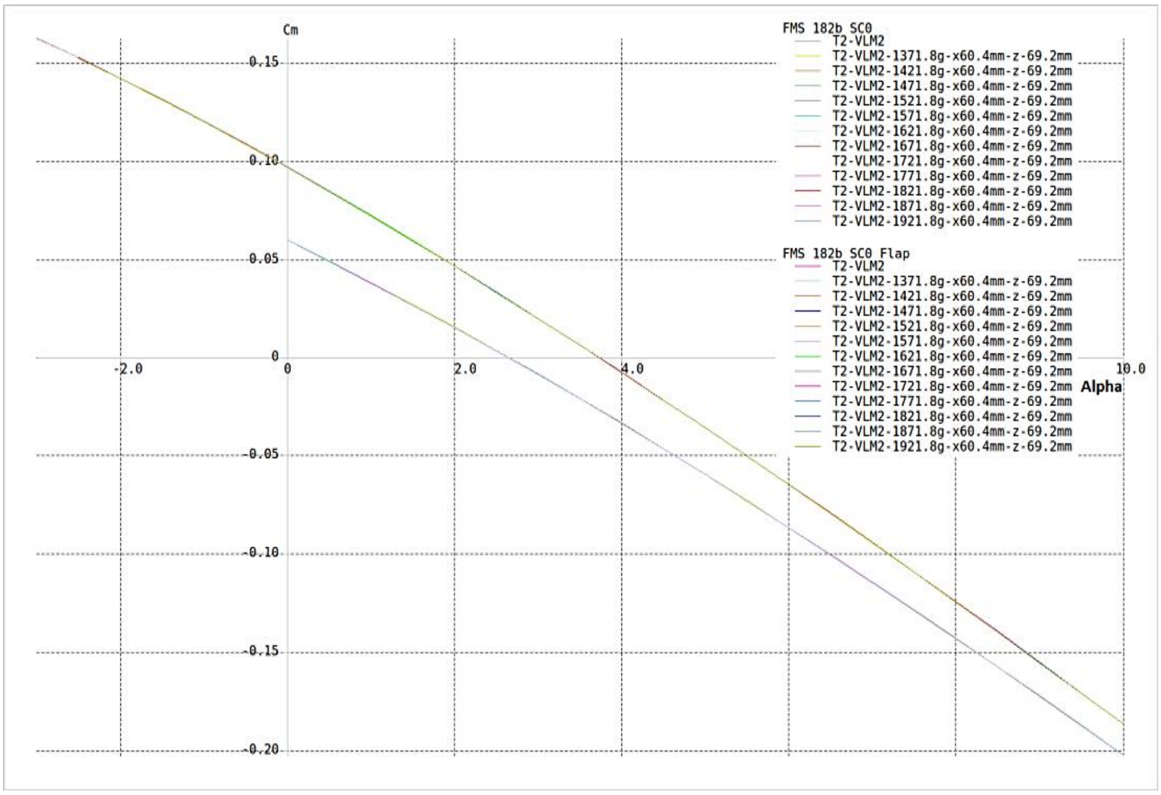


Figure 4.78 Cm against Alpha for both model variants for varying mass of SC0 and SC0 Flap models

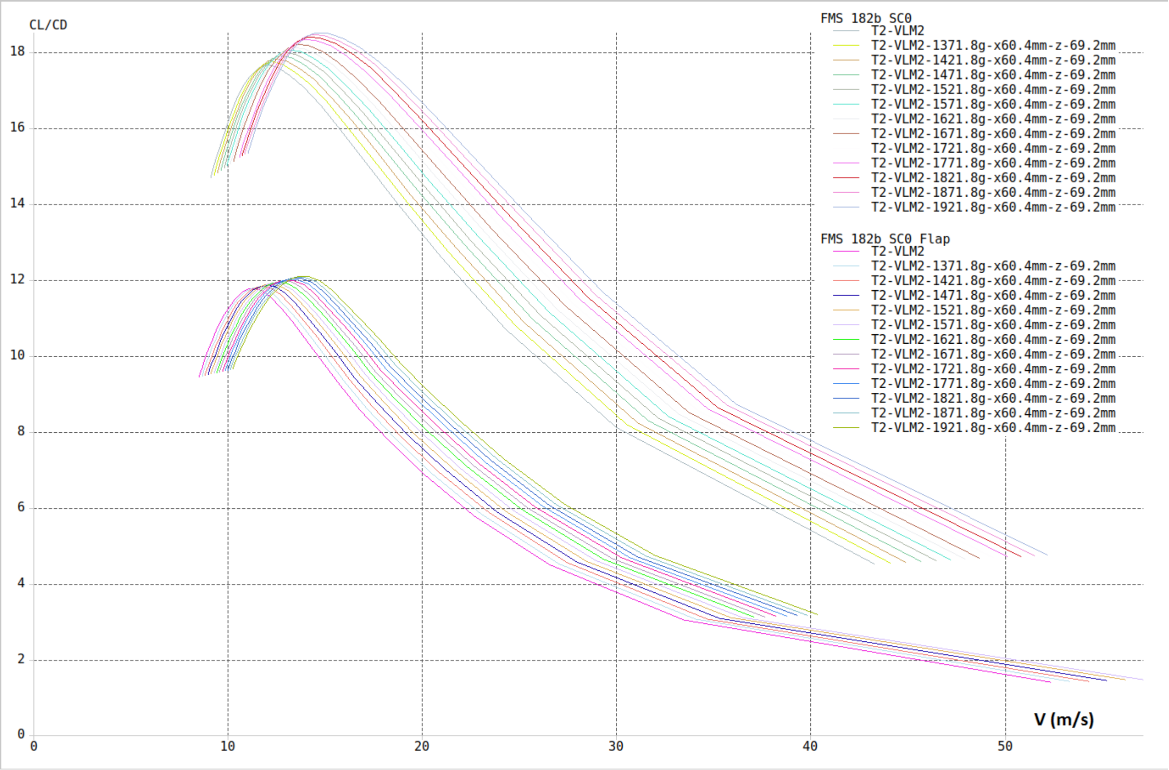


Figure 4.79 CL/CD against alpha for both models for varying mass of SC0 and SC0 Flap models

However, the changes are small enough to be of no concern. The set of curves in Figure 4.78 (FMS 182b SC0 Flap) are missing data points due to computation error. Figure 4.78 shows the change observed in the pitching moment due to the shift in the aircraft's weight. For the FMS 182b SC0, the added weight did not create any shift and the trim condition was at alpha 2.64. For the FMS 182b SC0 flapped model, the trim condition was at alpha 3.72 degrees regardless of the increased weight.

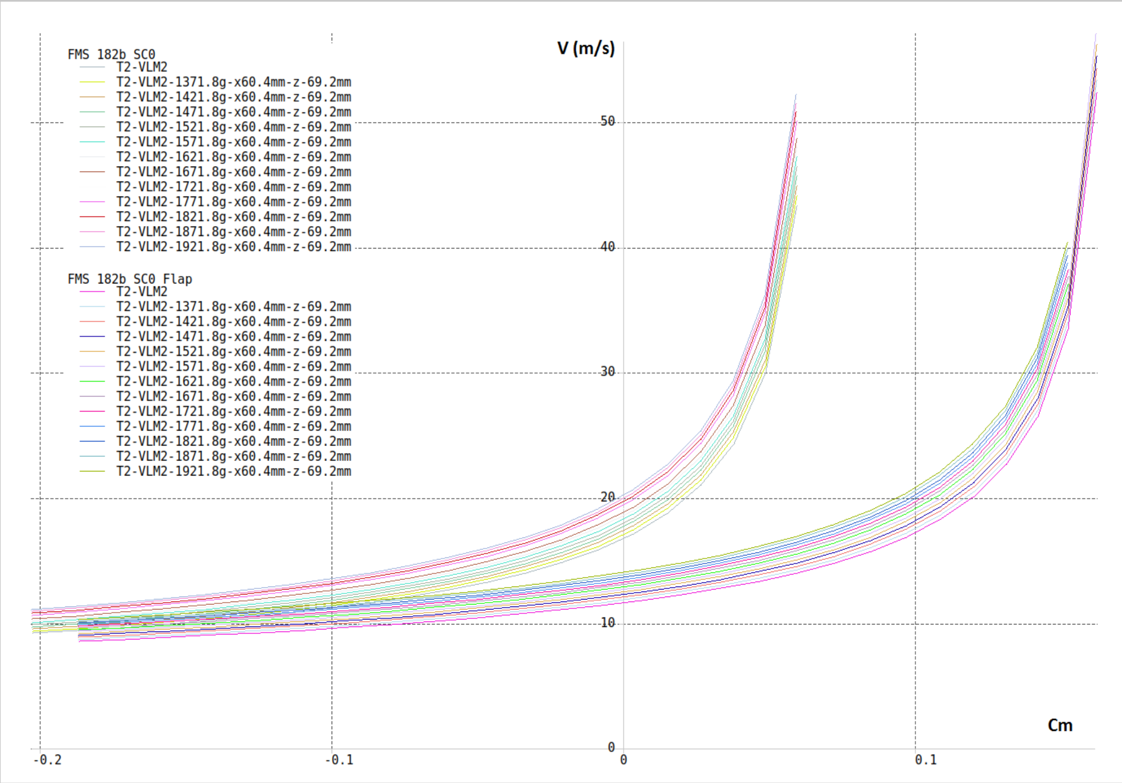


Figure 4.80 Trim velocity variations for change in mass for SC0 and SC0 Flap models

The above graph (Figure 4.80) shows the trend observed for the trim velocities for the mass increasing test. The point of interest is where the curves cross the Y axis (velocity) as that determines the trim velocities for every situation. Table 4-8 provide a summary of the observation.

Weight	Trim speed (SC0)	Trim alpha (SC0)	Trim speed (SC0_flapped)	Trim alpha (SC0_flapped)
1321.8	16.7	2.6425	11.6	3.72
1371.8	17.05	2.6425	11.85	3.72
1421.8	17.35	2.6425	12.05	3.72
1471.8	17.65	2.6425	12.25	3.72
1521.8	17.95	2.6425	12.45	3.72
1571.8	18.25	2.6425	12.65	3.72
1621.8	18.55	2.6425	12.85	3.72
1671.8	18.85	2.6425	13.05	3.72
1721.8	19.1	2.6425	13.25	3.72
1771.8	19.4	2.6425	13.45	3.72
1821.8	19.65	2.6425	13.65	3.72
1871.8	19.95	2.6425	13.85	3.72
1921.8	20.2	2.6425	14	3.72

Table 4-8 Variation in trim airspeed for changing mass for SC0 and SC0 Flap models

4.1.7 Centre of Gravity (C.G.) variation tests

The previous two models were used to conduct a similar test but this time the centre of gravity of the airplane was shifted with its weight held constant. The centre of gravity was shifted in its x position by 0% of the static margin up to 50%. Which is from the neutral point to the 55% of the static margin. For the flapped condition, the test was done up to 55% due to the shift in centre of lift aft of the baseline.

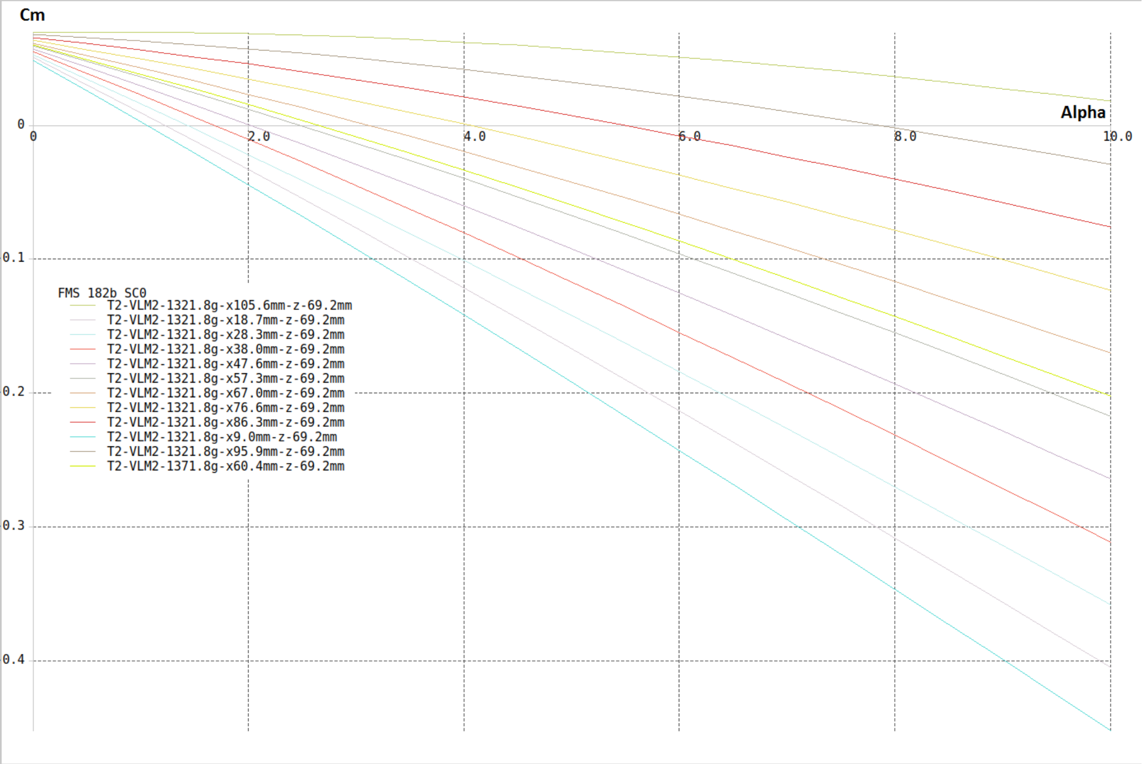


Figure 4.81 Variation of Cm against Alpha for shifting Centre of Gravity for SC0 model

The above graph shows the changes in pitching moment for every unit shift in the centre of gravity location. As it can be seen, beyond a certain amount, the aircraft fails to trim (no valid CM=0 situation for alpha). See data table for details. The general pattern suggests that measured from the reference point (leading edge of the main wing), the further out the CG is shifted, the higher the angle of attack needs to be for trimmed flight and vice versa. However, above a certain distance aft of the reference point the aircraft will fail to find any valid angle of attack for a trim condition. The ideal position of the static margin is a balance between stability and control authority. If the static margin is too low, the short period pitching oscillation will be under-damped and result in undesirable flying characteristics. If however it is too high for the available control authority, it results in excessive control input requirements and in extreme scenarios, insufficient elevator authority altogether.

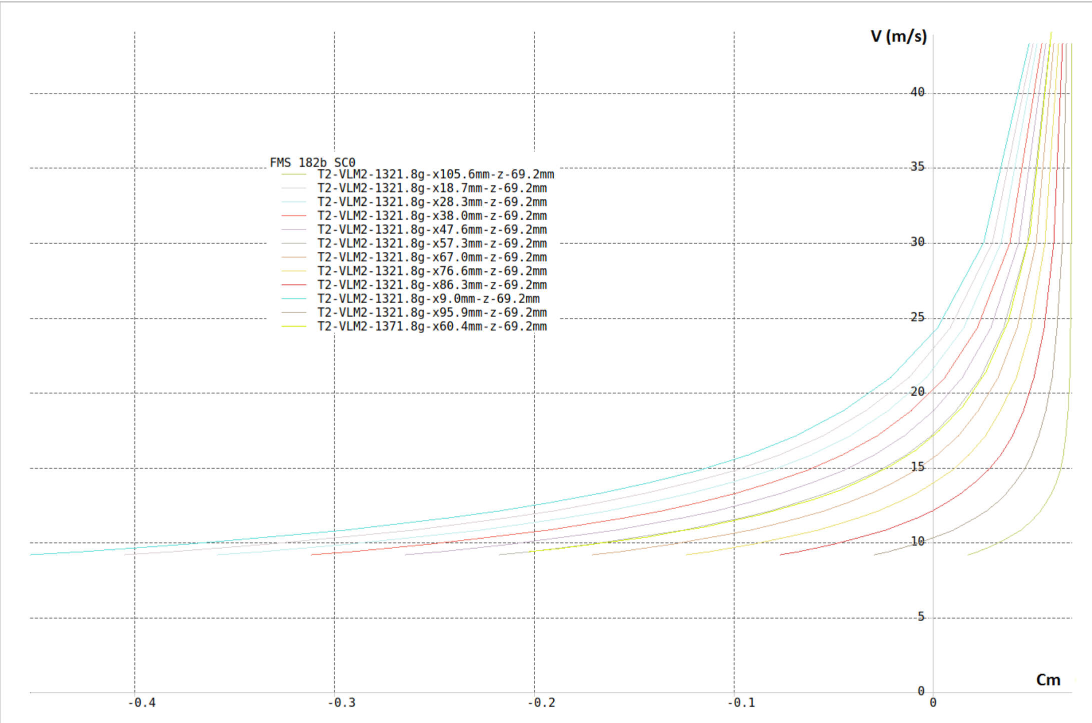


Figure 4.82 Variation in V against Cm for shifting C.G for SC0 model

The above graph shows the changes in airspeed required for level flight against Cm. The point of interest is the CM=0 (y-axis) where the aircraft is in trim. As it can be seen, the general trend is that with shifting of the CG location backwards, the trim velocity goes down until the point where the aircraft fails to trim at all (N/A in the table). See the following table for summary.

Configuration	COG	Static Margin %	Trim Speed	Trim Alpha
sc0	105.599	0	NA	NA
sc0	95.9401	5	10.25	7.825
sc0	86.2812	10	12.1	5.5
sc0	76.6223	15	13.9	4.05
sc0	66.9634	20	15.6	3.1
sc0	60.364	23.42	16.7	2.65
sc0	57.3045	25	17.2	2.475
sc0	47.6456	30	18.7	2.025
sc0	37.9867	35	20.2	1.675
sc0	28.3278	40	21.5	1.425
sc0	18.6689	45	22.9	1.225
sc0	9.01	50	24	1.05

Table 4-9 Variation in Trim speed and Trim alpha with change in C.G for SC0 model

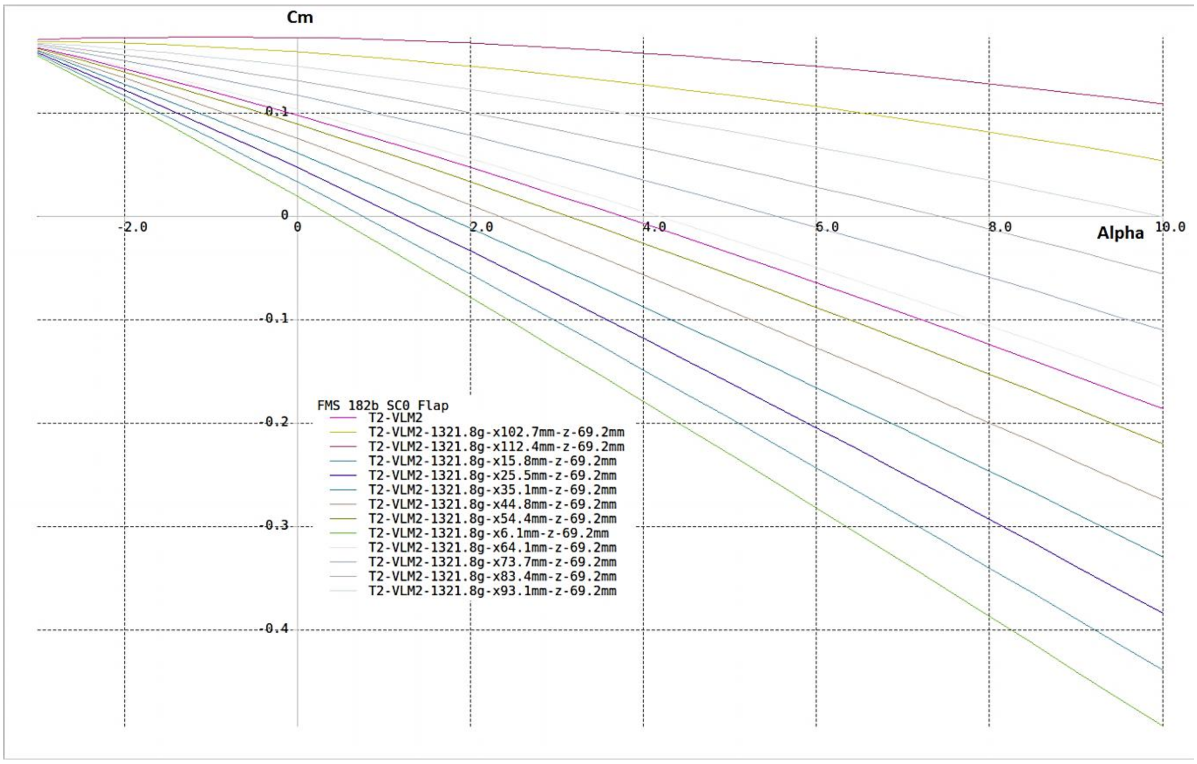


Figure 4.83 Cm against Alpha graph for variation in C.G. for flapped configuration

The above graph shows the outputs for the flapped configuration for CM vs alpha at varied CG intervals. The major difference noticed is that the added camber from the flaps results in trim conditions being available at lower alphas as well as slightly higher lift coefficient from the same alpha. Deploying the flaps shifts the centre of lift back from the baseline neutral point. With it, the CG is also shifted for the same static margin, resulting in failure to trim for the 5% position as well as the 0%. See following tables for a summary.

Configuration	COG	Static Margin %	Trim Speed	Trim Alpha
sc0_flapped	112.381	0	NA	NA
sc0_flapped	102.7221	5	NA	NA
sc0_flapped	93.0632	10	8.50	9.950
sc0_flapped	83.4043	15	9.50	7.400
sc0_flapped	73.7454	20	10.50	5.525
sc0_flapped	64.0865	25	11.25	4.075
sc0_flapped	60.364	26.93	11.60	3.725
sc0_flapped	54.4276	30	12.20	3.125
sc0_flapped	44.7687	35	13.00	2.325
sc0_flapped	35.1098	40	13.75	1.675
sc0_flapped	25.4509	45	14.50	1.175
sc0_flapped	15.792	50	15.25	0.750
sc0_flapped	6.1331	55	16.00	0.400

Table 4-10 Variation of Trim speed and Trim alpha for shifting C.G.

In the following stability and control tests, the aforementioned results are utilised to narrow the scope of the investigation as well as inform important and appropriate test parameters.

4.1.8 Stability Tests

4.1.8.1 Base model (SC0):

The following results have been obtained for the base model (FMS182b SC0).

SPPO:

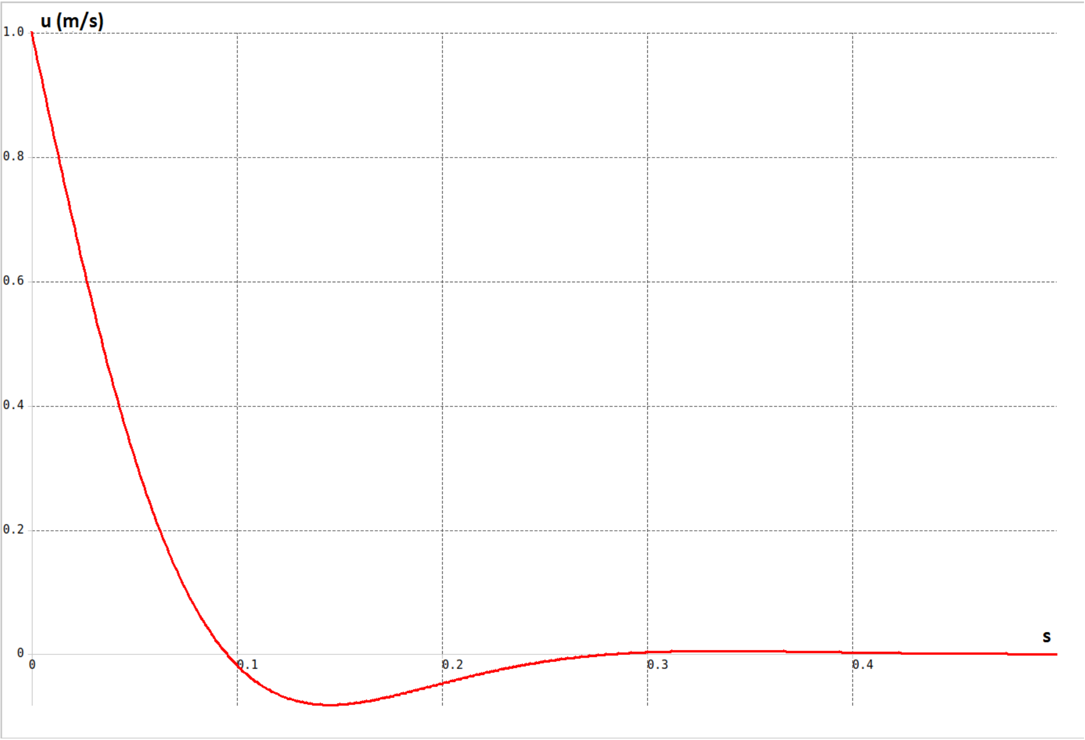


Figure 4.84 Time response: Forward speed (u) against time (s)

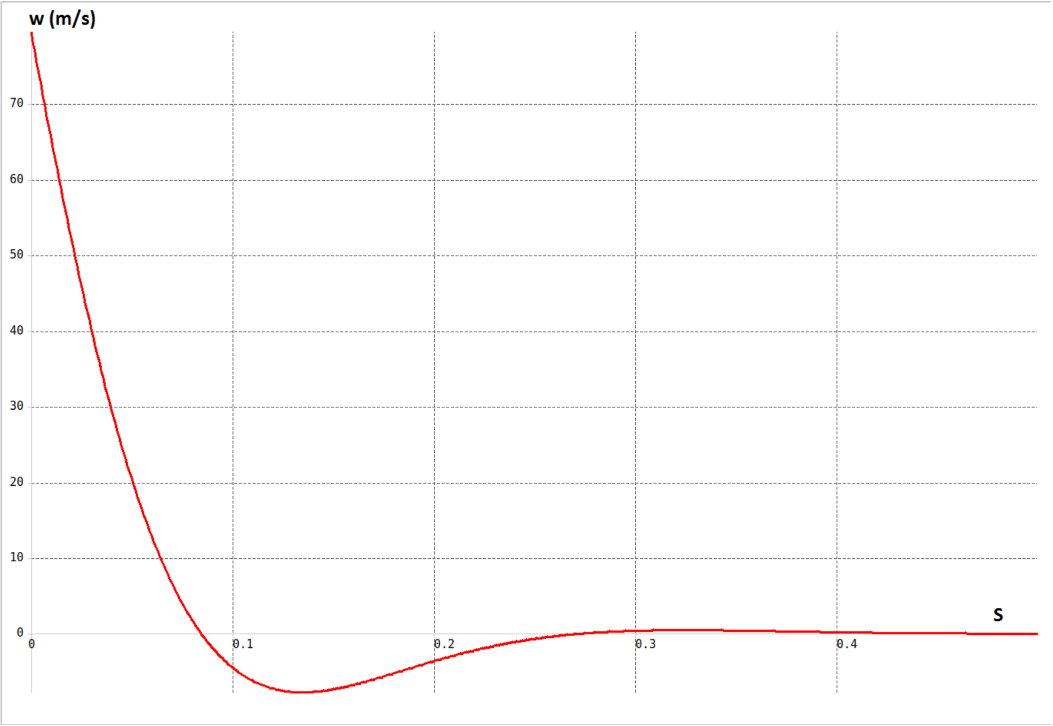


Figure 4.85 Time response: Vertical speed (w) against time (s)

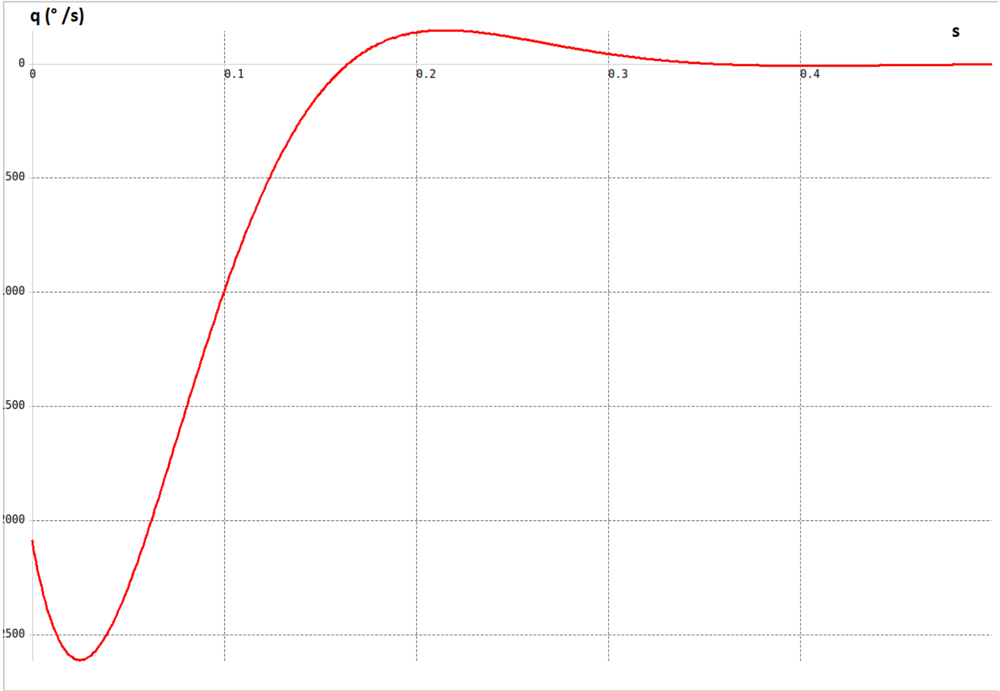


Figure 4.86 Time response: Pitch rate (q) against time (s)

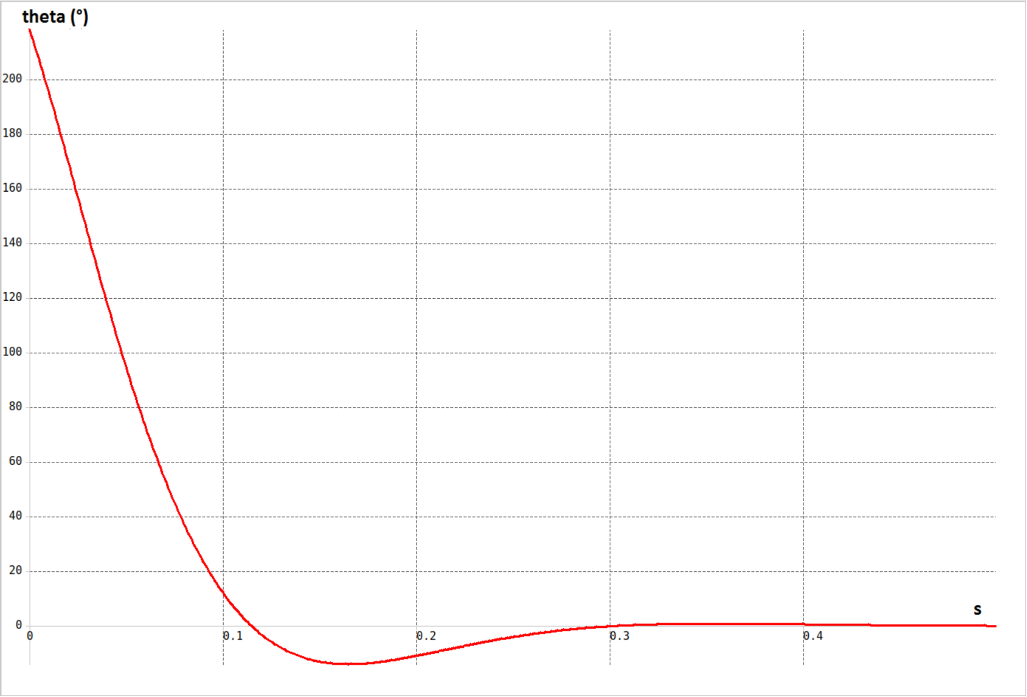


Figure 4.87 Time response: Theta (degrees) against time (s)

Phugoid:

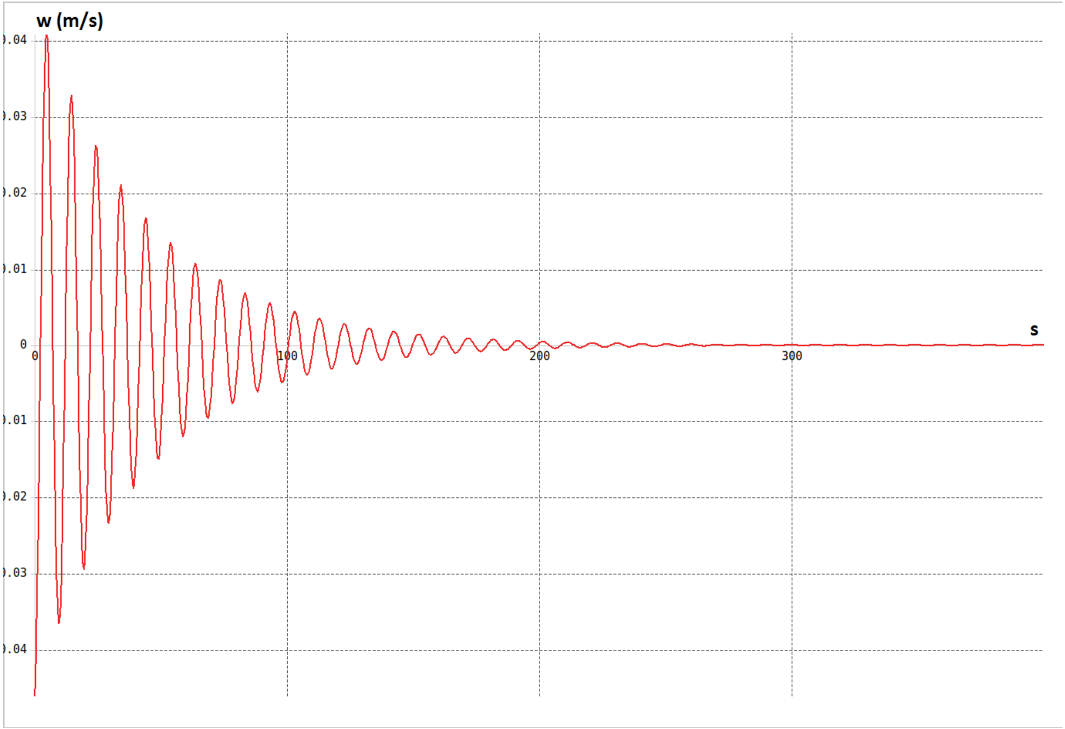


Figure 4.88 Time response: vertical speed (w) against time (s)

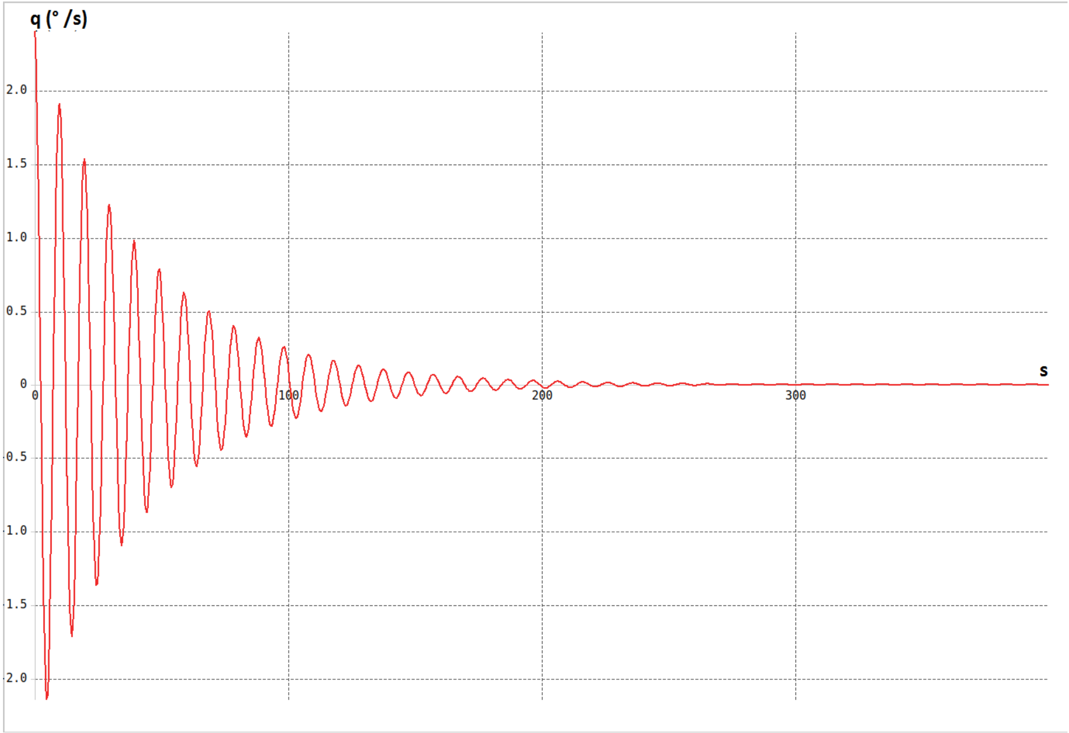


Figure 4.89 Time response: Pitch rate (q) against time (s)

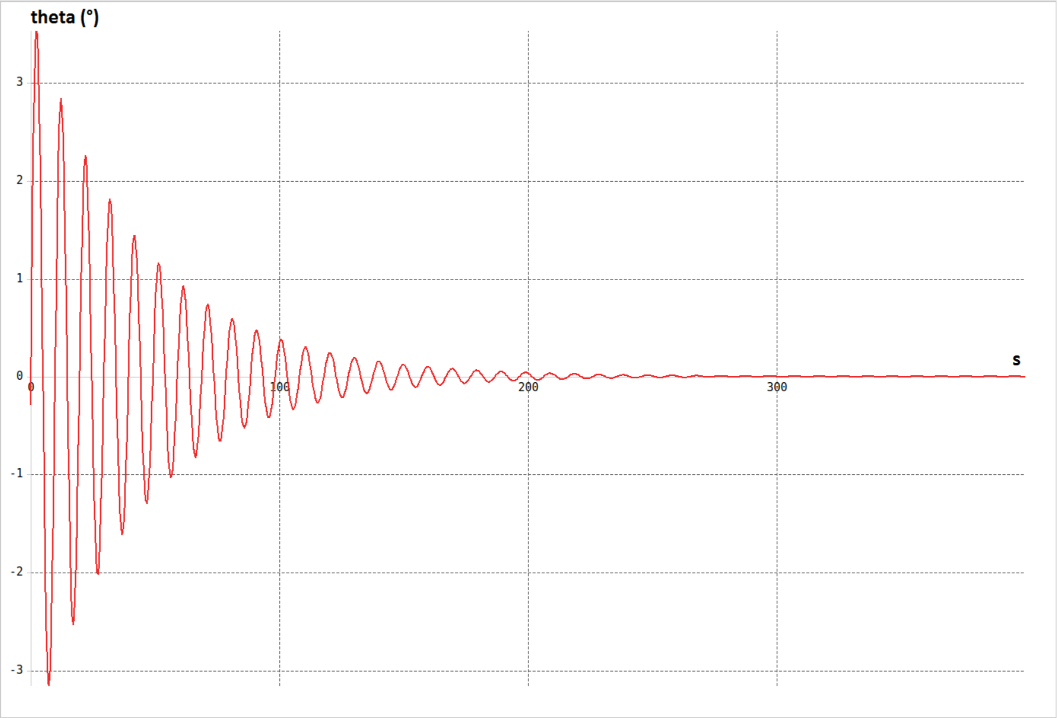


Figure 4.90 Time response: Pitch angle (theta) against time (s)

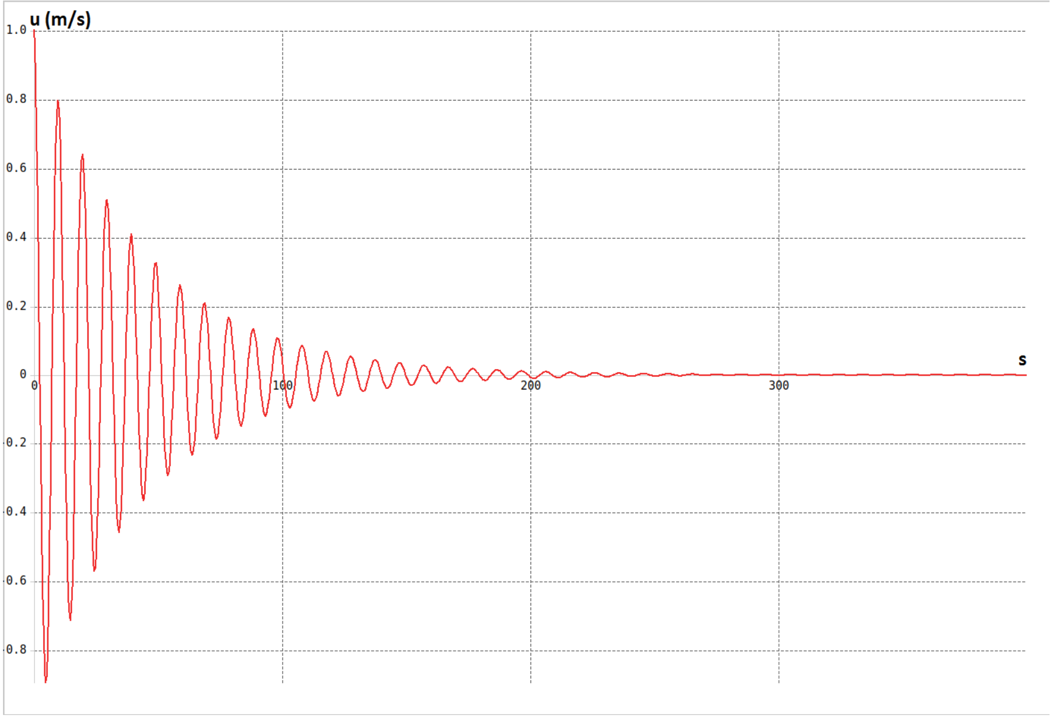


Figure 4.91 Time response: Forward speed (u) against time

Roll Damping:

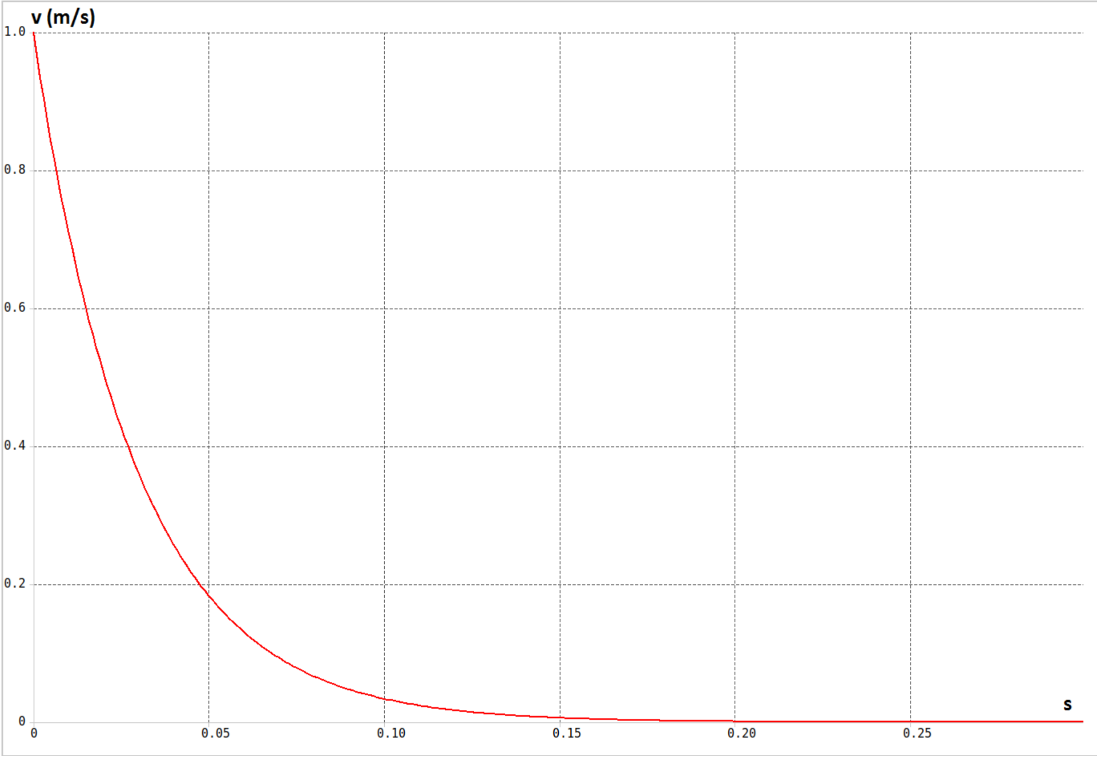


Figure 4.92 Time response: Forward speed (v) against time (s)

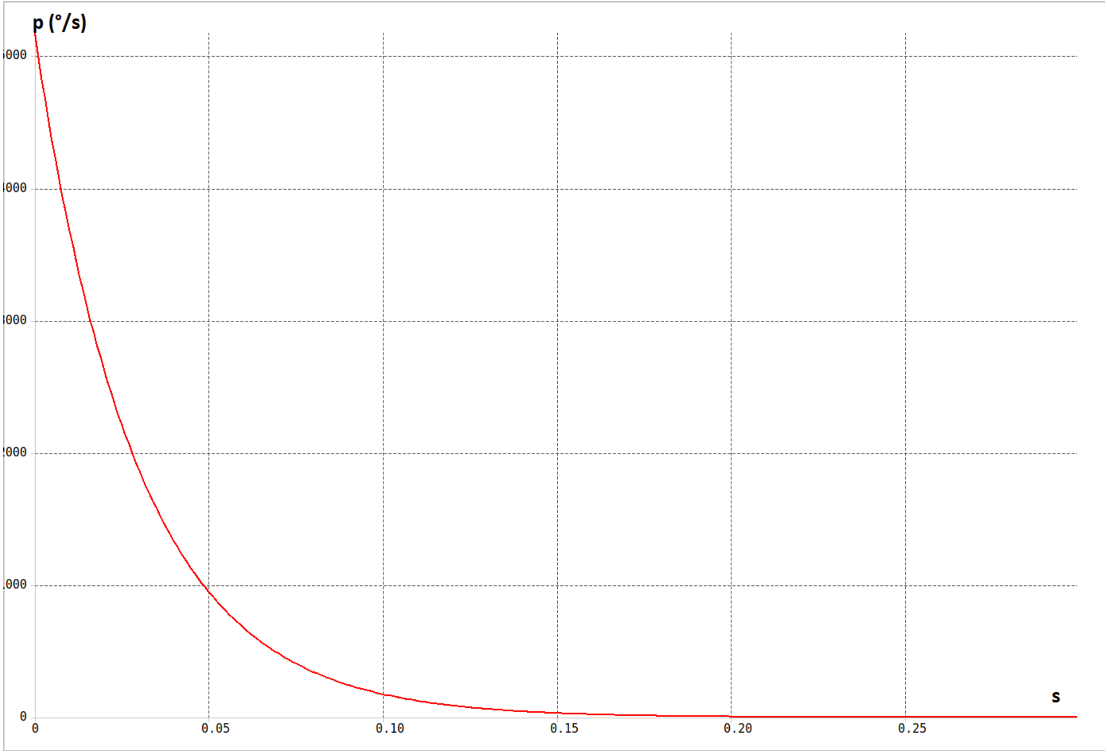


Figure 4.93 Time response: Roll rate (p) against time (s)

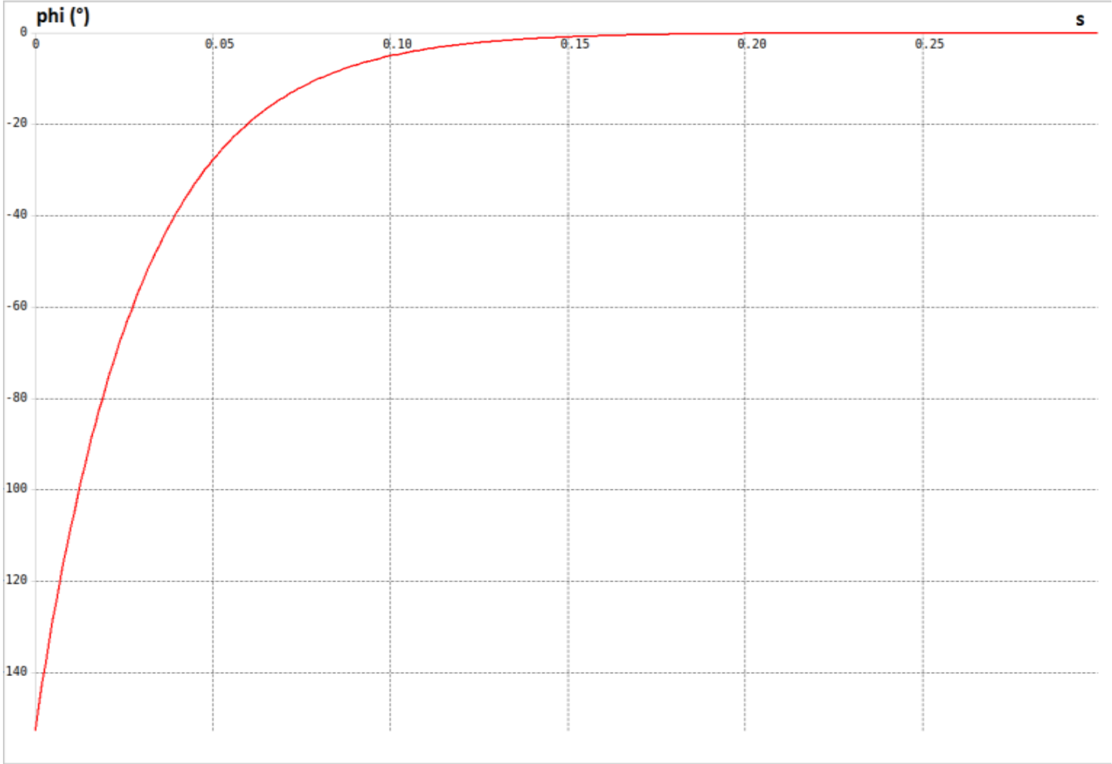


Figure 4.94 Time response: Roll angle (phi) against time (s)

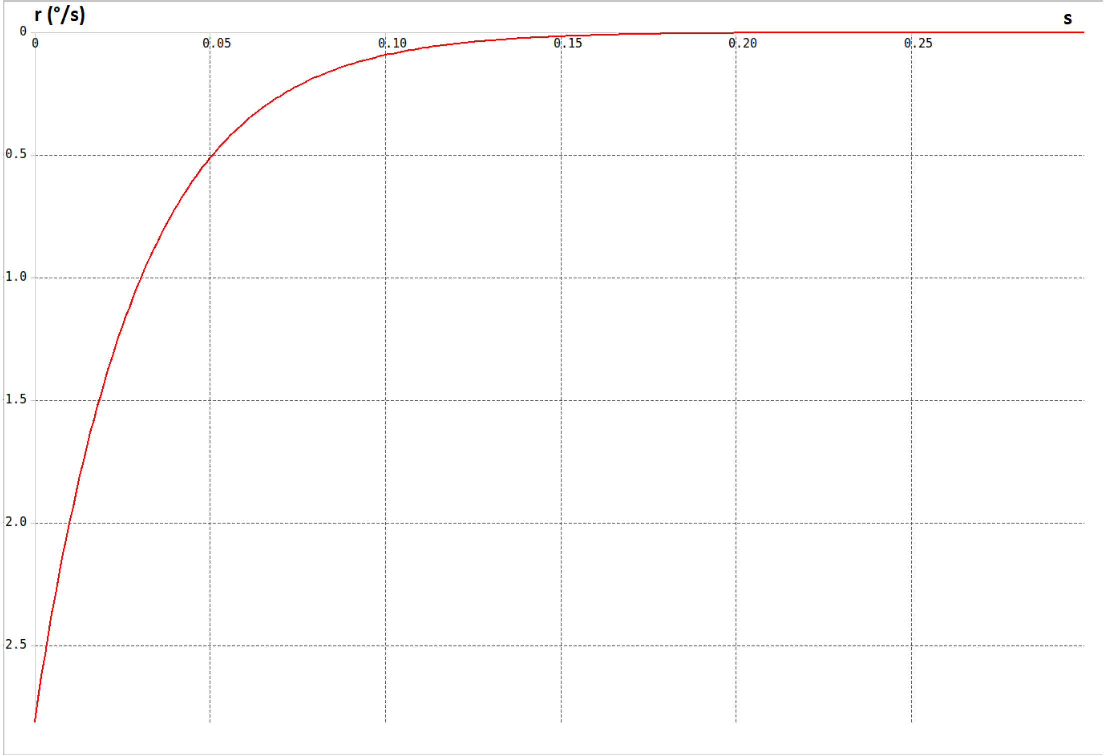


Figure 4.95 Time response: Yaw rate (r) against time (s)

Spiral:

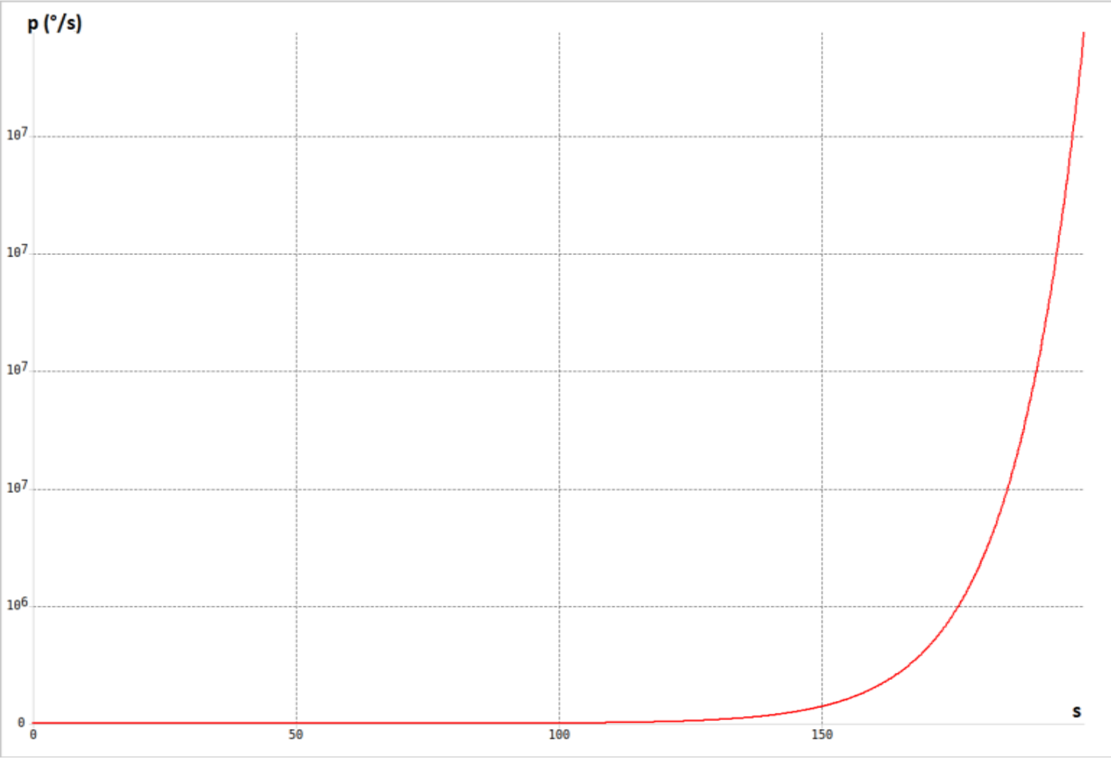


Figure 4.96 Time response: Roll rate (p) against time (s)

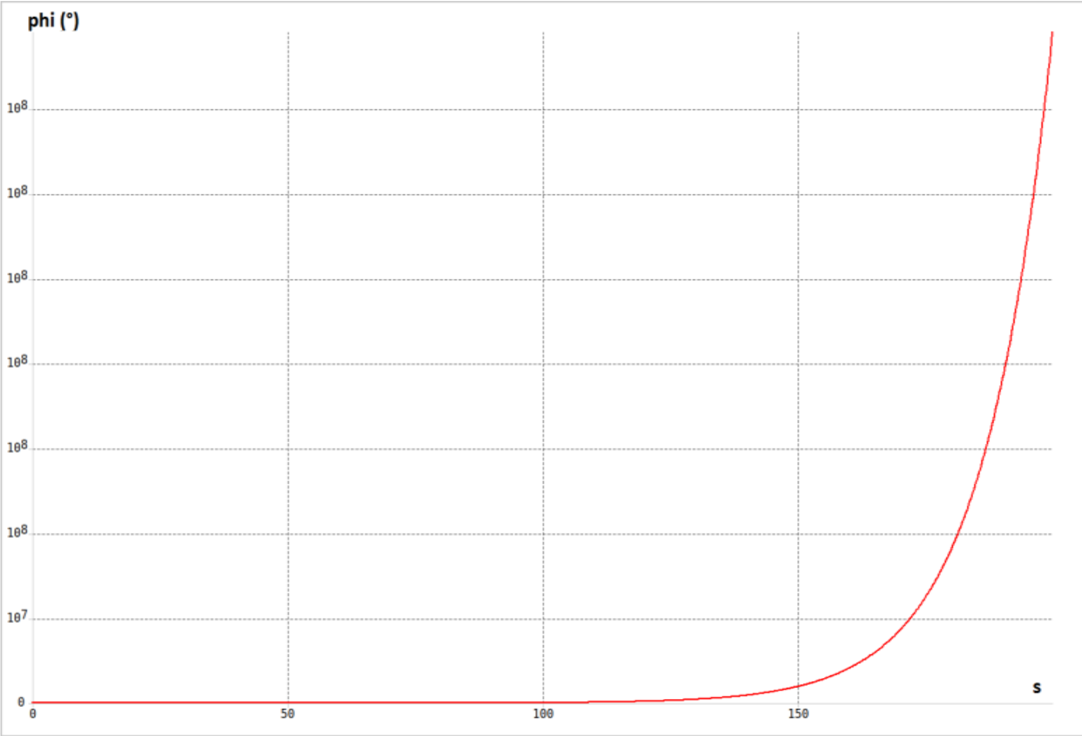


Figure 4.97 Time response: Pitch rate (phi) against time (s)

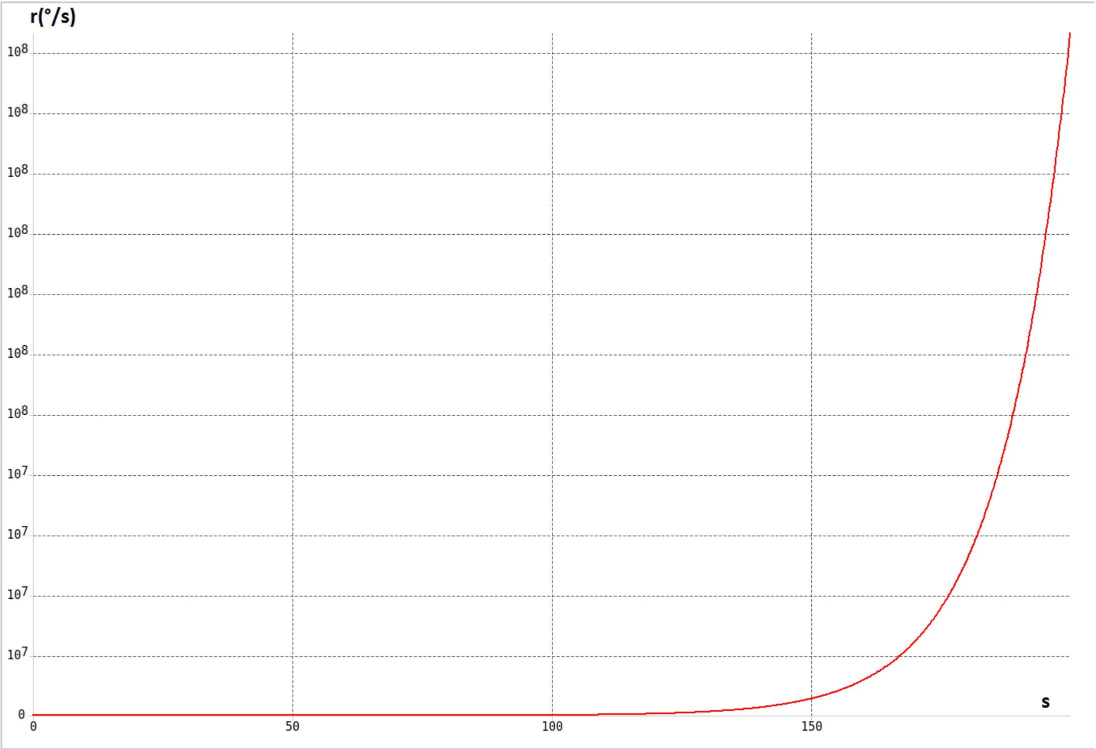


Figure 4.98 Time response: Yaw rate (r) against time (s)

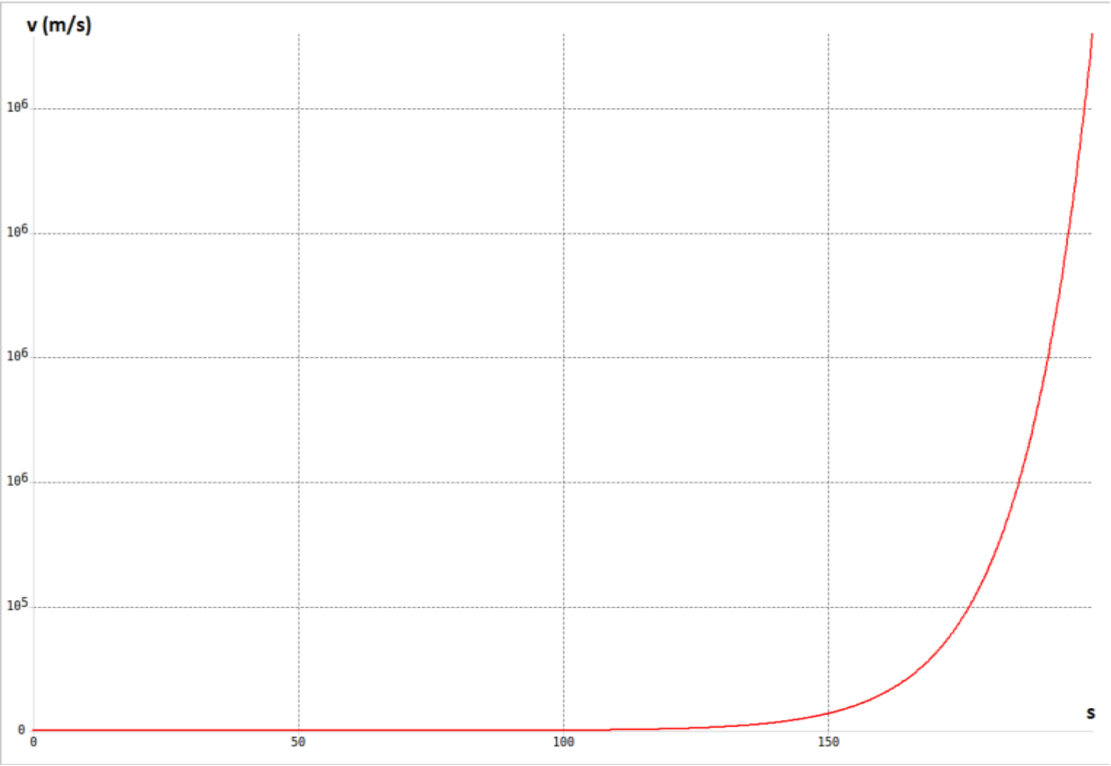


Figure 4.99 Time response: Lateral speed (v) against time (s)

Dutch Roll:

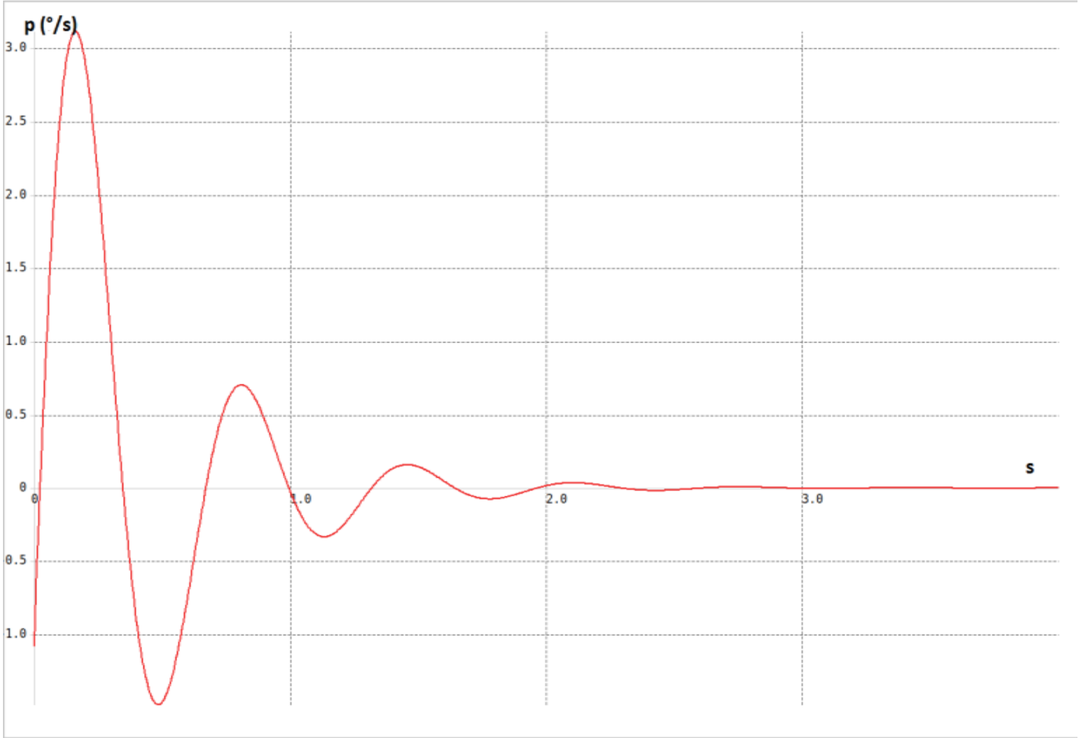


Figure 4.100 Time response: Roll rate (p) against time (s)

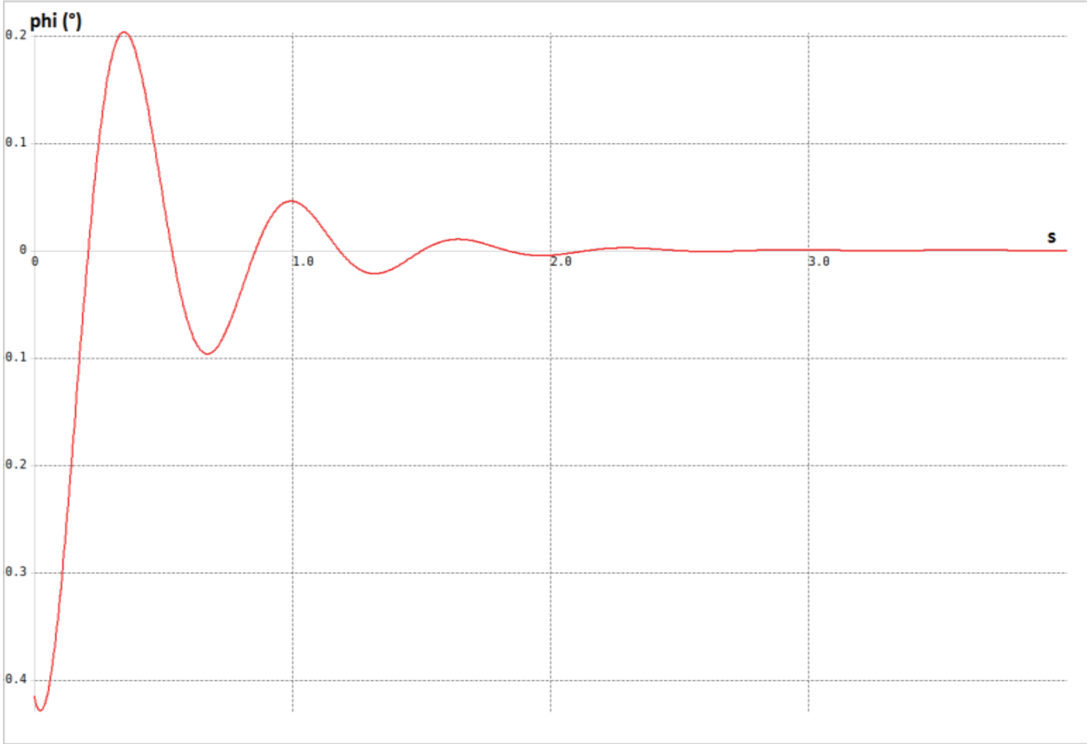


Figure 4.101 Time response: Roll rate (phi) against time (s)

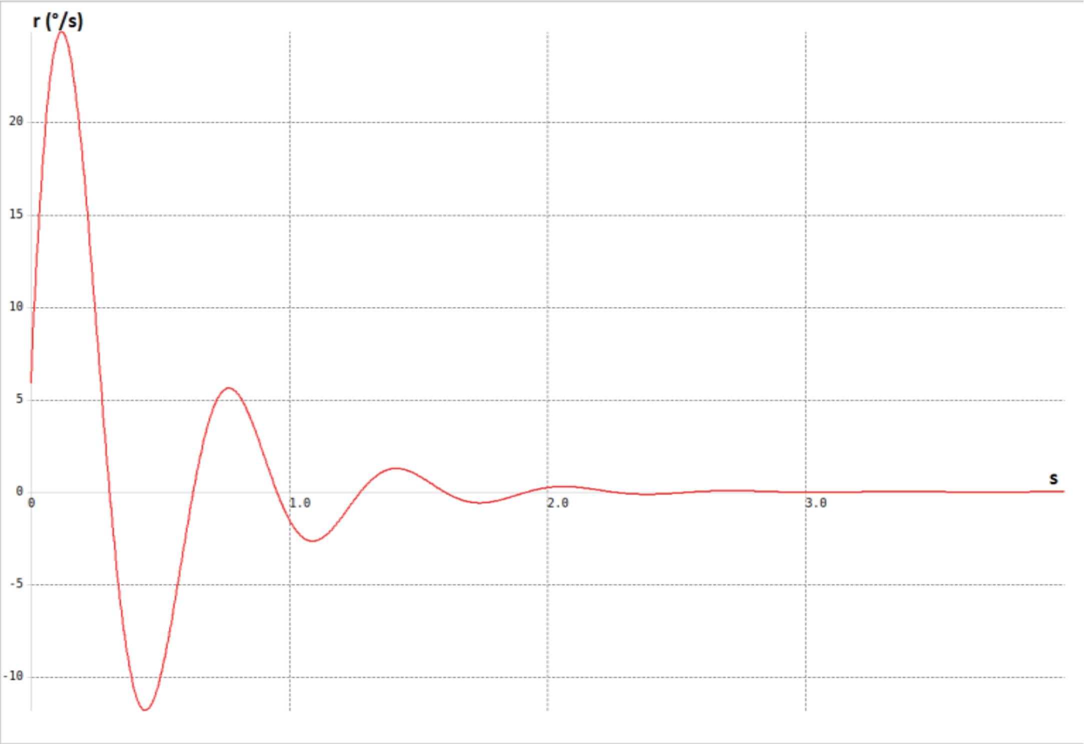


Figure 4.102 Time response: Yaw rate (r) against time (s)

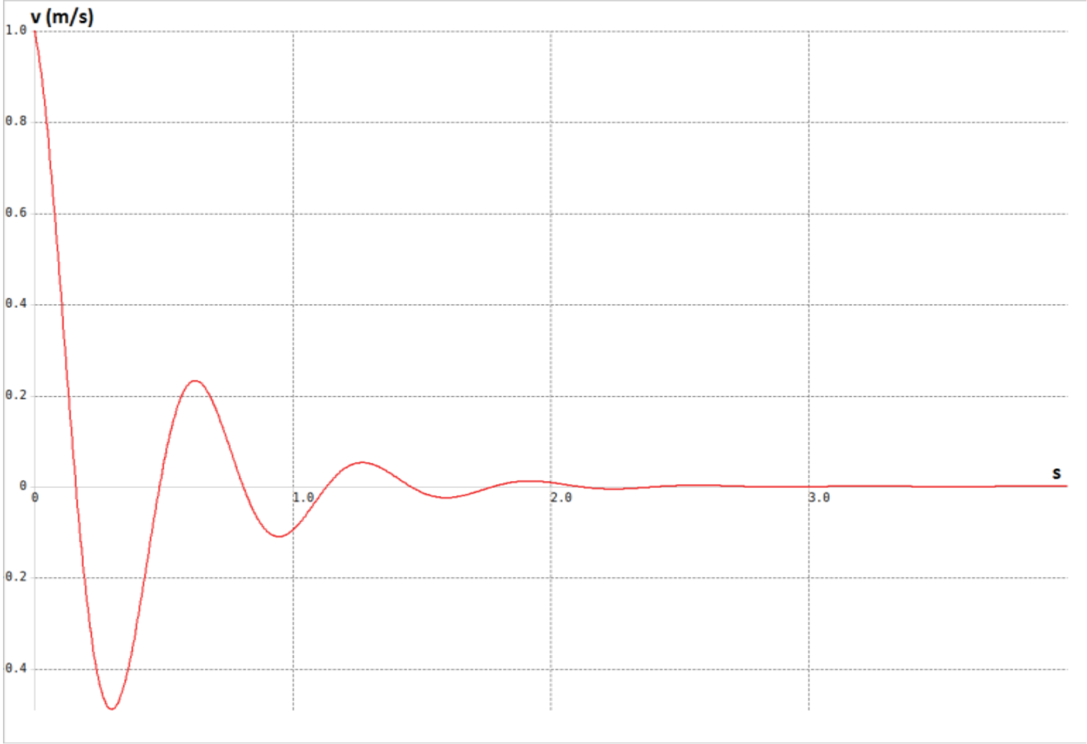


Figure 4.103 Time response: Lateral speed (v) against time (s)

Root Locus:

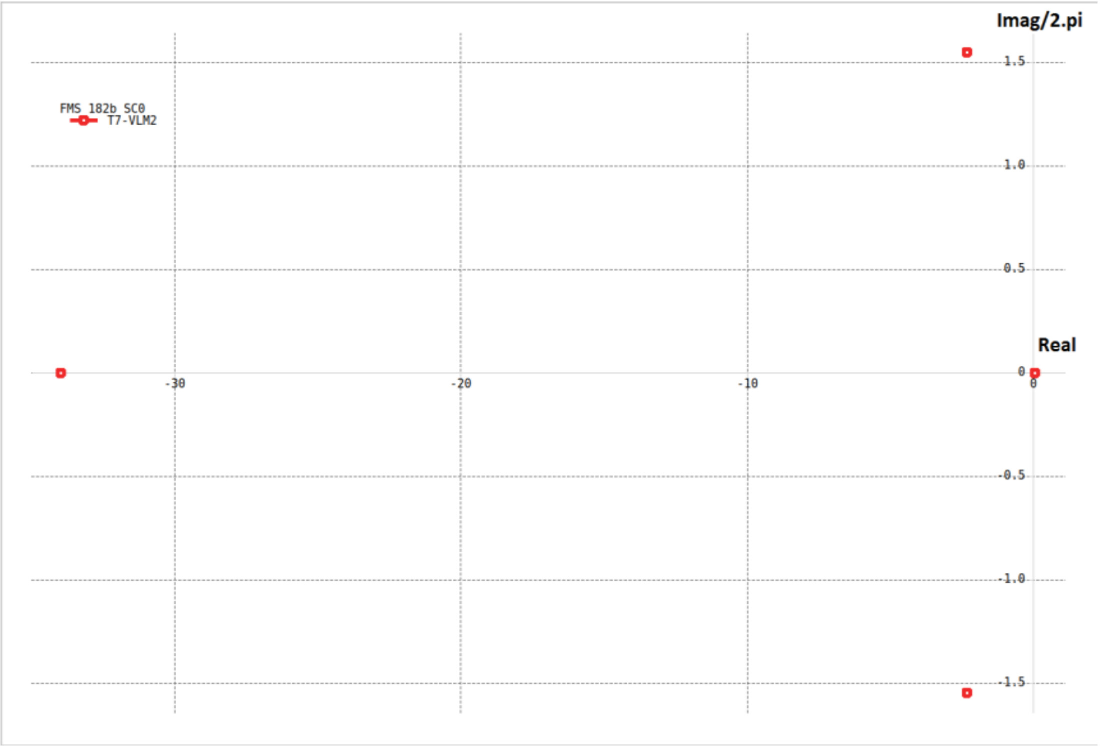


Figure 4.104 Root locus for FMS 182b SC0 (lateral)

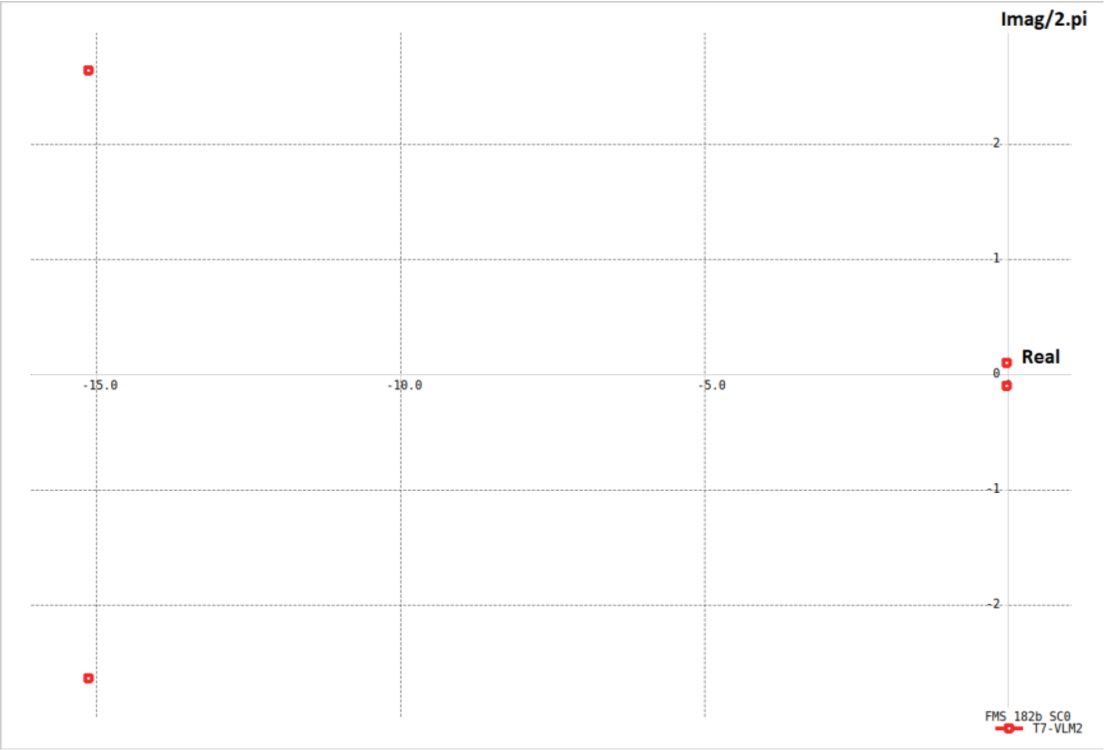


Figure 4.105 Root locus for FMS 182b SC0 (longitudinal)

4.1.8.2 Flapped model (SC0 flap):

The model variant (SC0 flap) was used to conduct the same stability tests as the previous section and the results are presented in the Appendix (Chapter 7.6.2). Instead of repeating similar plots, the following subsections compare the stability characteristics of these two model variants (SC0 & SC0 flap).

4.1.8.3 Stability Results from mass variation experiments:

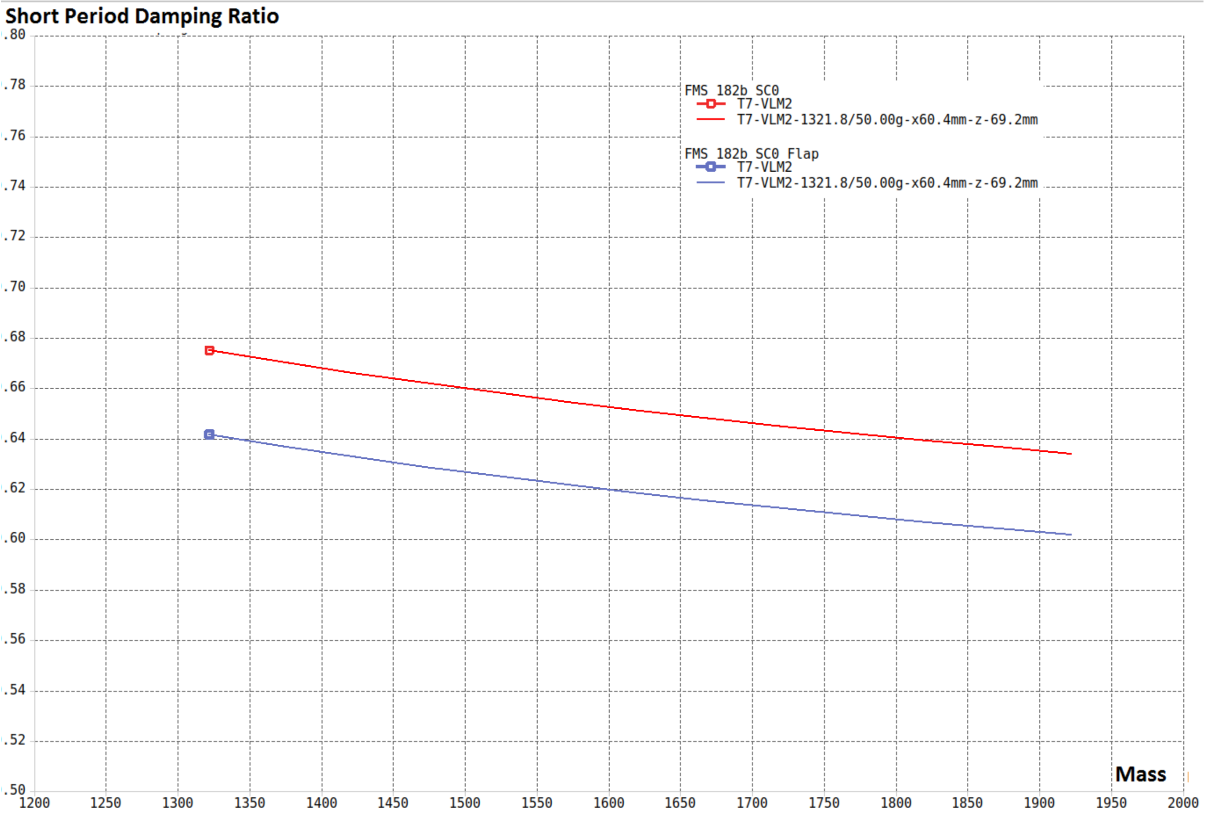


Figure 4.106 Changes to Short Period mode for mass variation of both models

The general trend for the Short Period Damping Ratio (SPDR) is that with increasing mass, the damping ration declines for both configurations. As there are no aerodynamic changes to the tail plane, this type of change is to be expected.

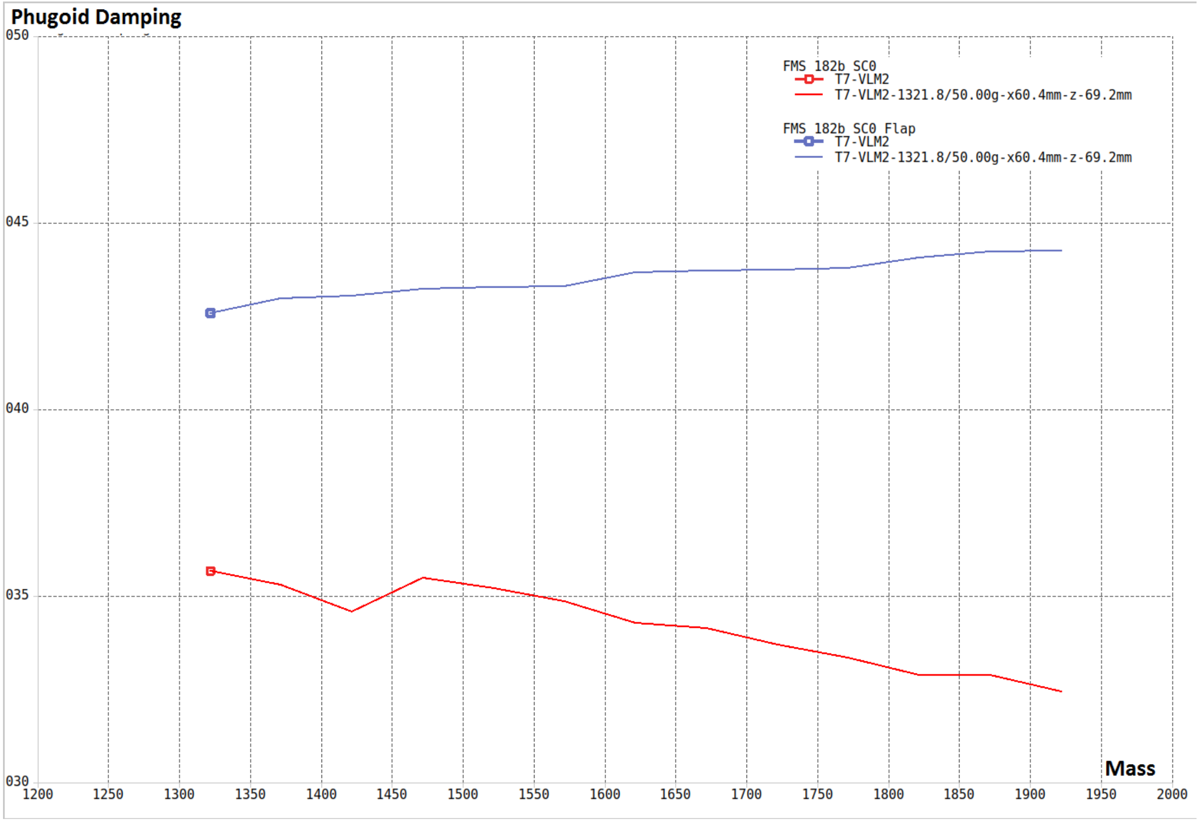


Figure 4.107 Changes to phugoid mode for mass variation of both models

Relative to each other, some change in the damping ratios are observed for the Phugoid mode. However, in absolute terms, it is very small and the aircraft remains stable. In the flapped situation, there is a slight improvement in the damping ratio, whereas, with increasing mass, the base model exhibits worsening of the damping.

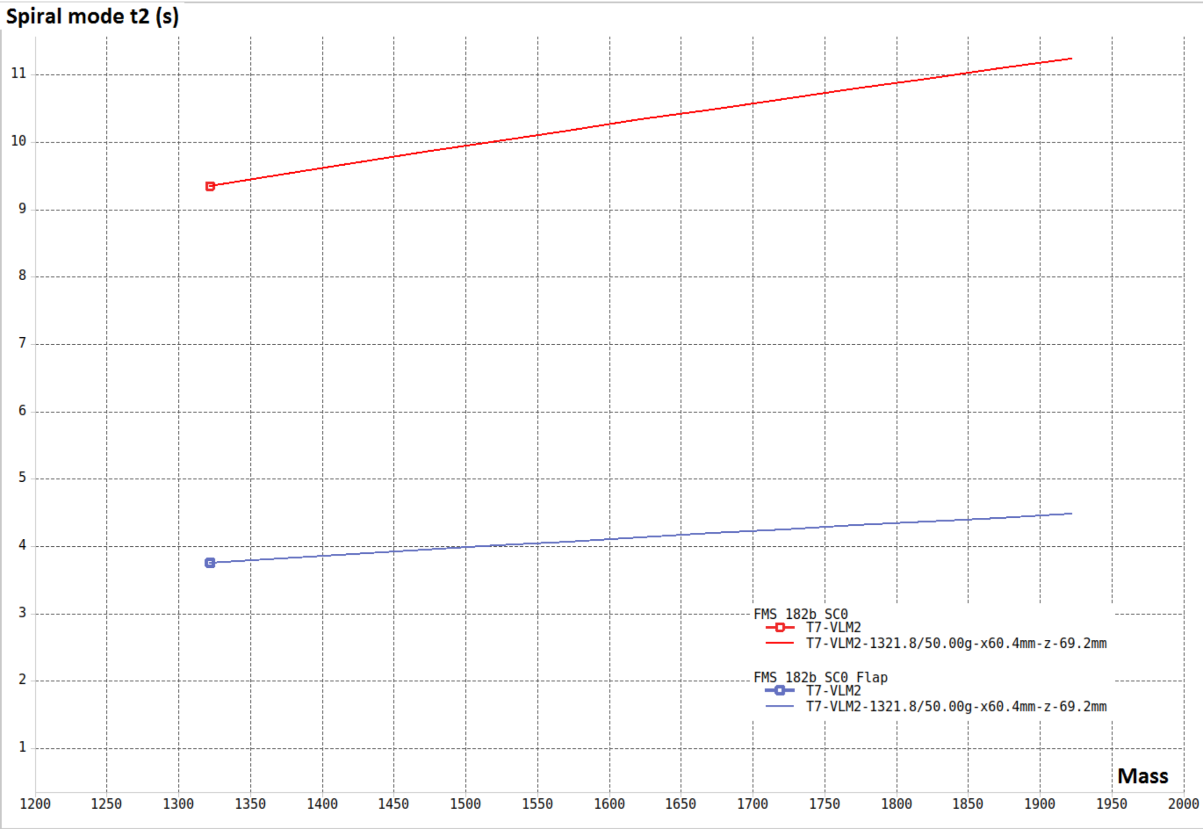


Figure 4.108 Changes in Spiral mode from mass variation for both models

With increasing mass, the spiral mode slows down for both configurations.

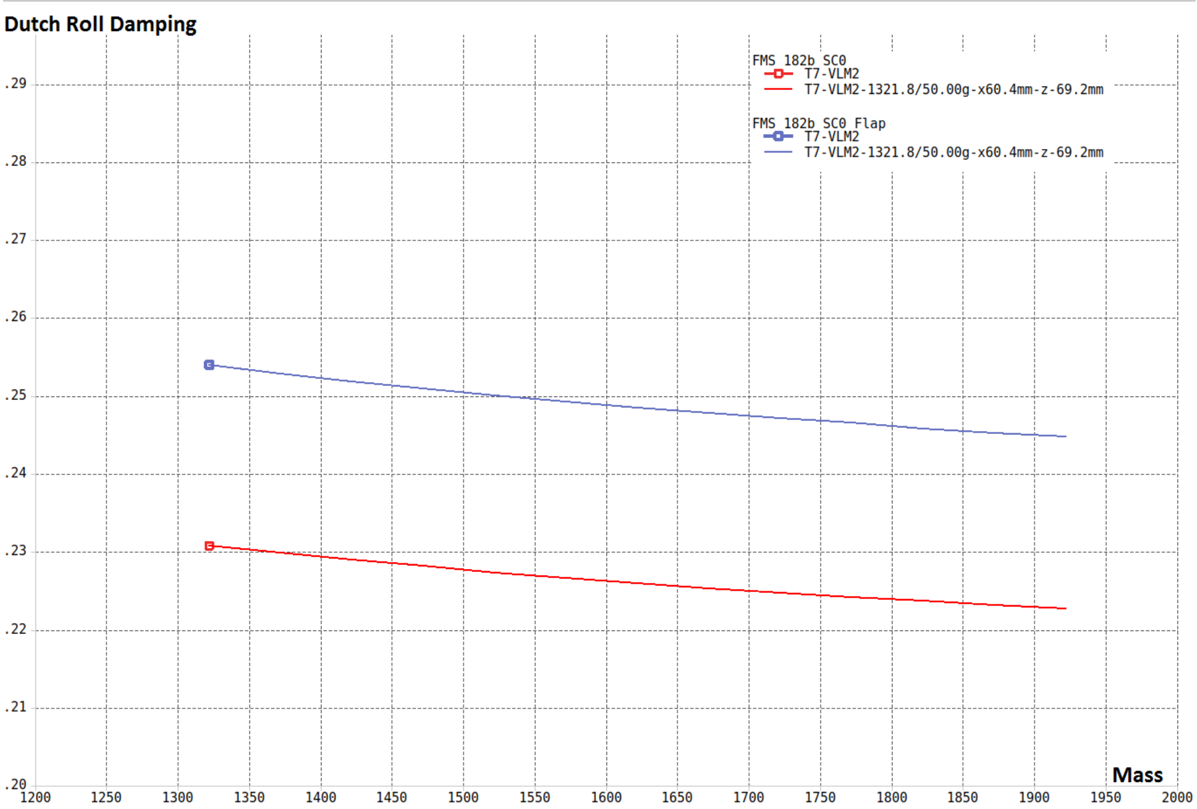


Figure 4.109 Changes to Dutch roll mode for mass variation for both models

Increasing mass causes the Dutch roll mode to be relatively more difficult to be damped.

4.1.8.4 Stability Results from C.G. variation experiments

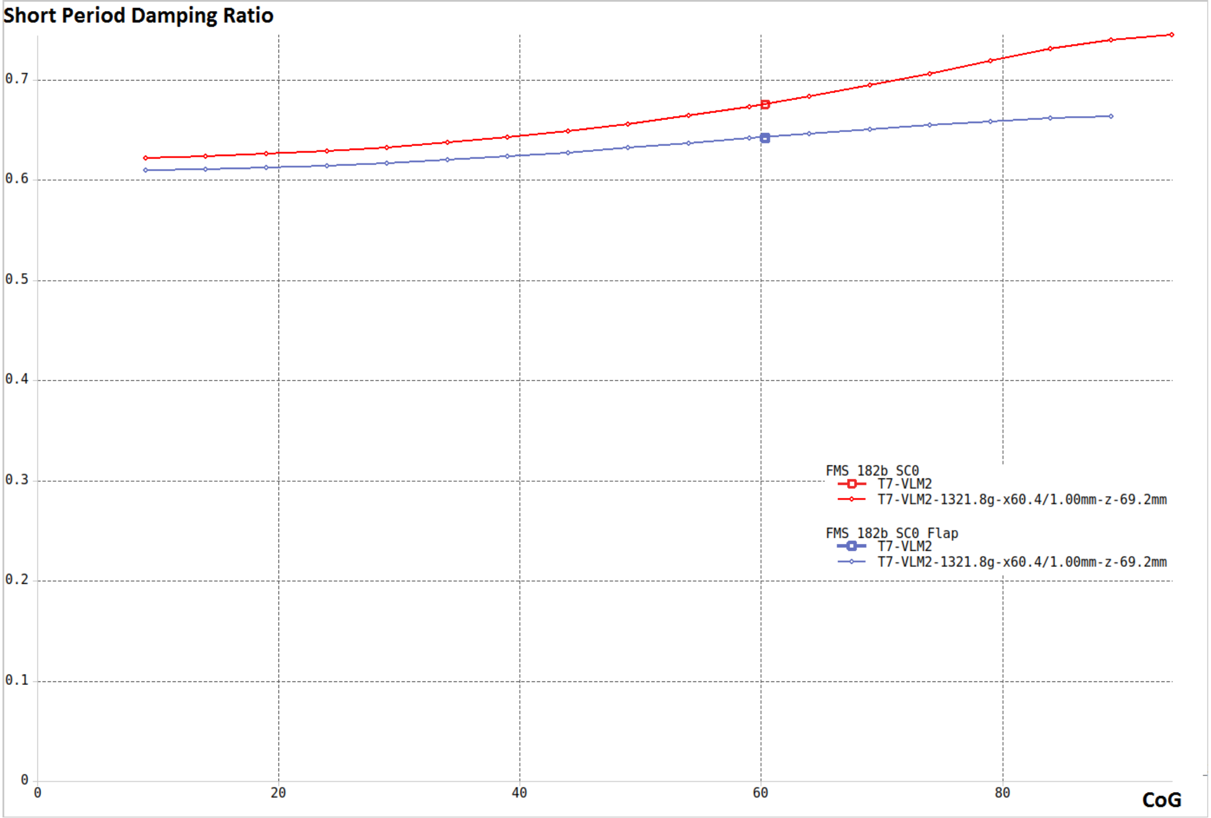


Figure 4.110 Changes to Short Period for C.G. variation for both models

As the Centre of Gravity (CG/CoG/C.G.) is shifted toward the leading edge of the wing, the damping ratio drops in magnitude in similar ways for both configurations. However, as the CG is shifted aft of the reference CG point (toward the tail), the damping ratio for both configuration changes with the base model showing the greatest amount of change.

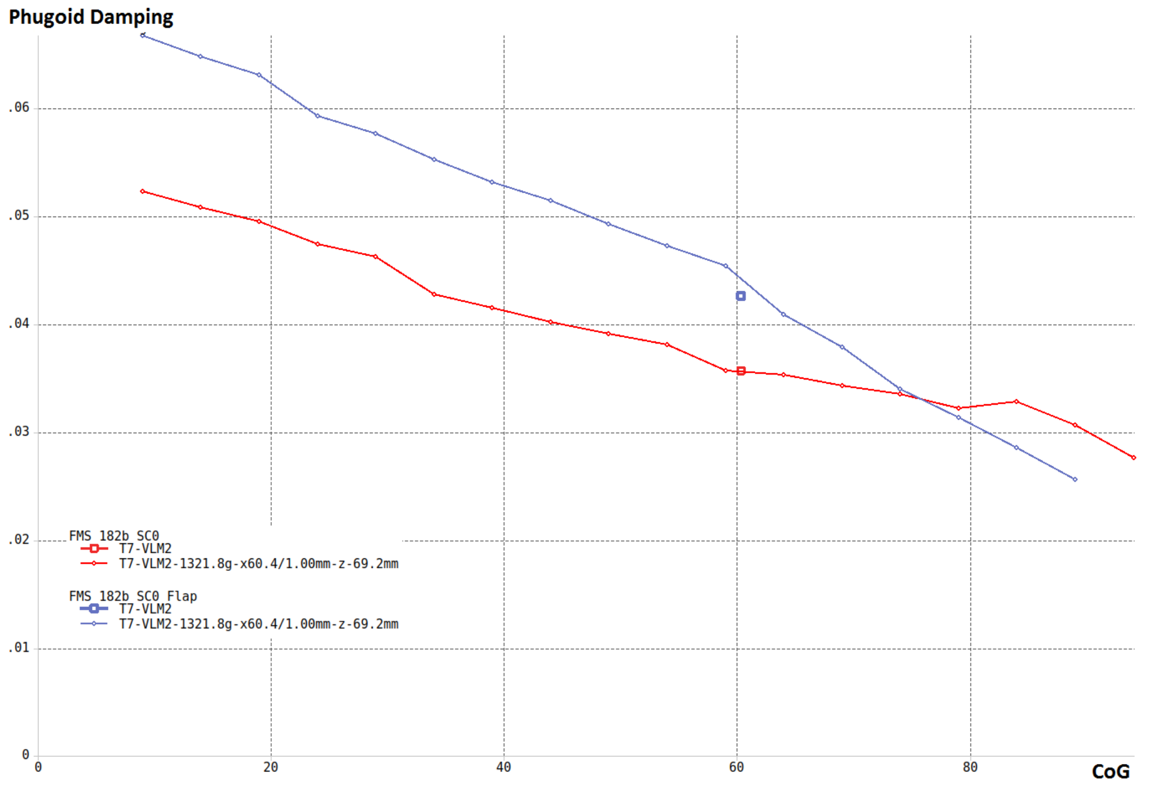


Figure 4.111 Changes to phugoid mode for C.G. variation for both models

The magnitude of Phugoid damping is increased as the COG is shifted forward of the reference point and decreased as the COG is moved in the opposite direction. There is a point between CoG_X 70 and 80 where both configurations share identical damping ratio.

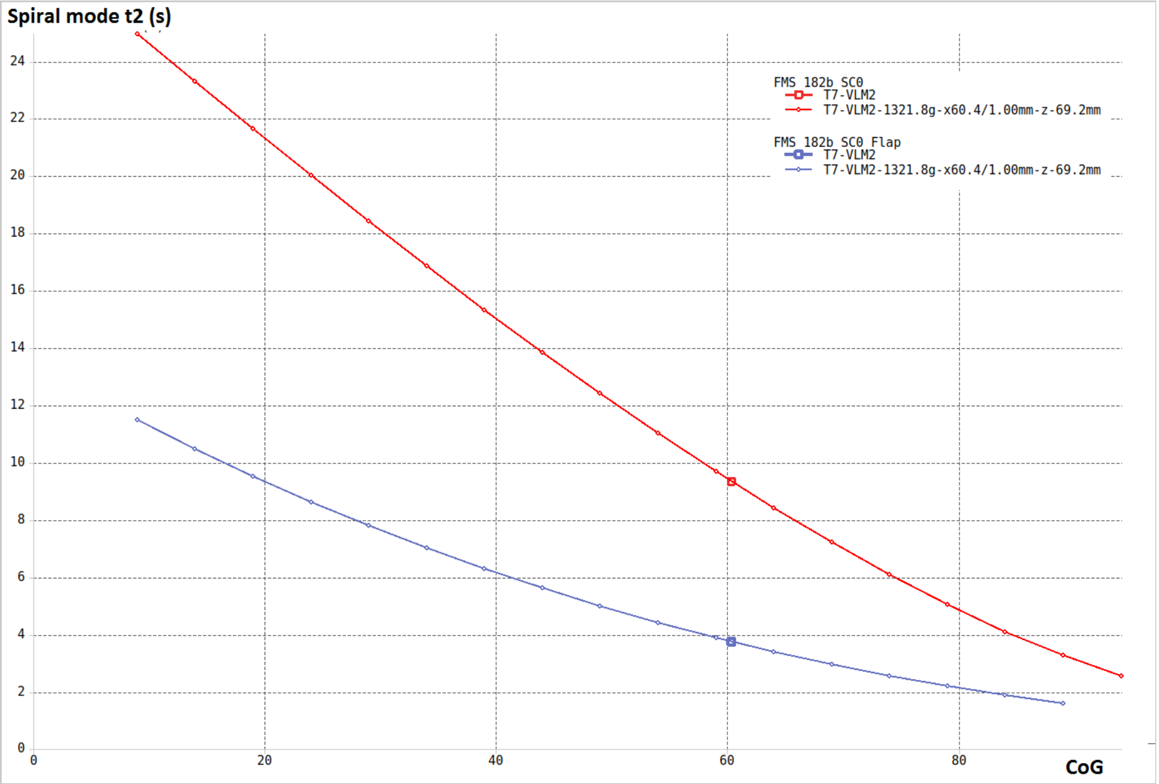


Figure 4.112 Changes to spiral model for C.G. variation for both models

Both configurations exhibit a similar pattern of change in spiral mode period where shifting CG towards the nose tends to increase it while it is reduced when shifting CG away from it.

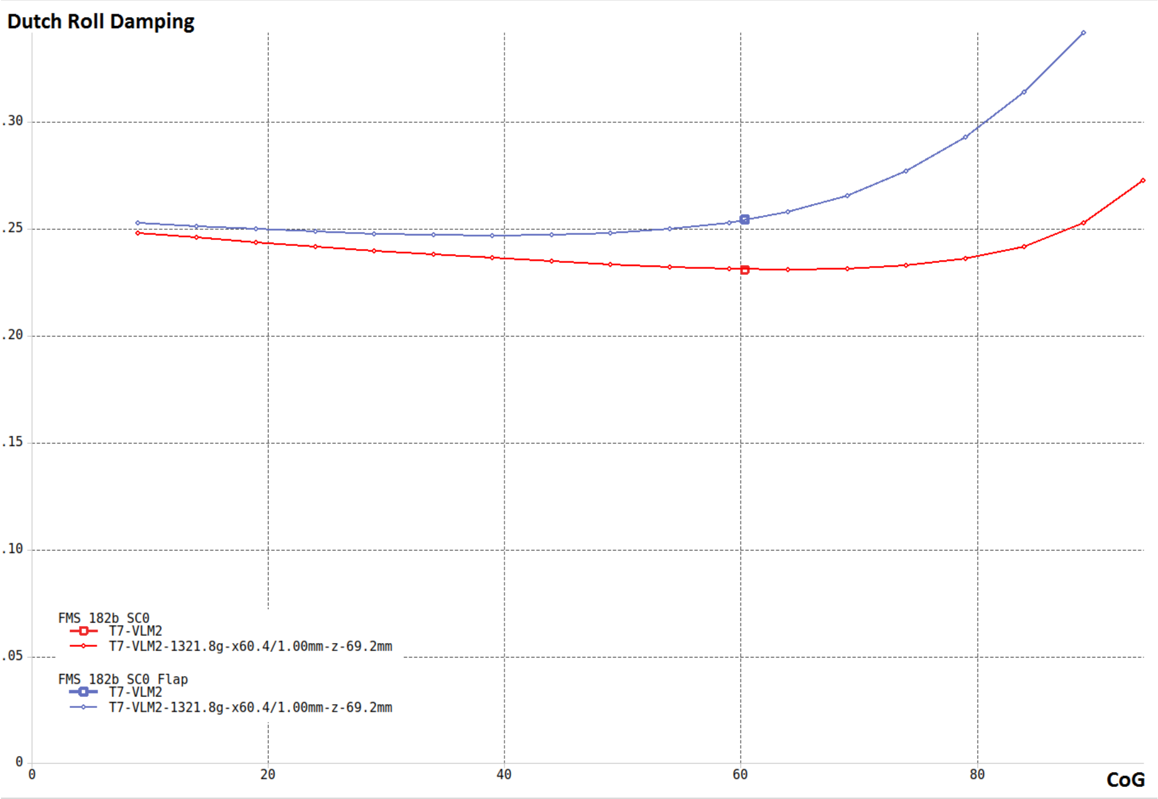


Figure 4.113 Changes to Dutch roll mode for C.G. variation for both models

For the base model, the ratio drops as the CG is shifted from the nose toward the tail until the reference point is reached. Beyond that, shifting the CG further causes the ratio to increase in magnitude. A similar pattern is observed for the flapped configuration, however, this change in direction happens before the reference CG point is reached.

4.1.8.5 Stability parameter estimations for SC0 and SC0_flap (base model and flapped model)

Due to the large volume of data generated, the results have been shifted to the Appendix (Chapter 7.4) for reference and documentation. They have been utilised for the X-Plane and Simulink modelling work.

4.1.8.6 Stability Derivatives for altered mass & CoG for base model and flapped model

Due to the large volume of data generated from the test, the results have been shifted to the Appendix (Chapter 7.4) for reference and documentation.

4.1.9 Control Test Result

The following results are for both model configurations. The ailerons, elevator and rudder were deflected from +15 to -15 with control gains to obtain the following results.

4.1.9.1 Aileron-

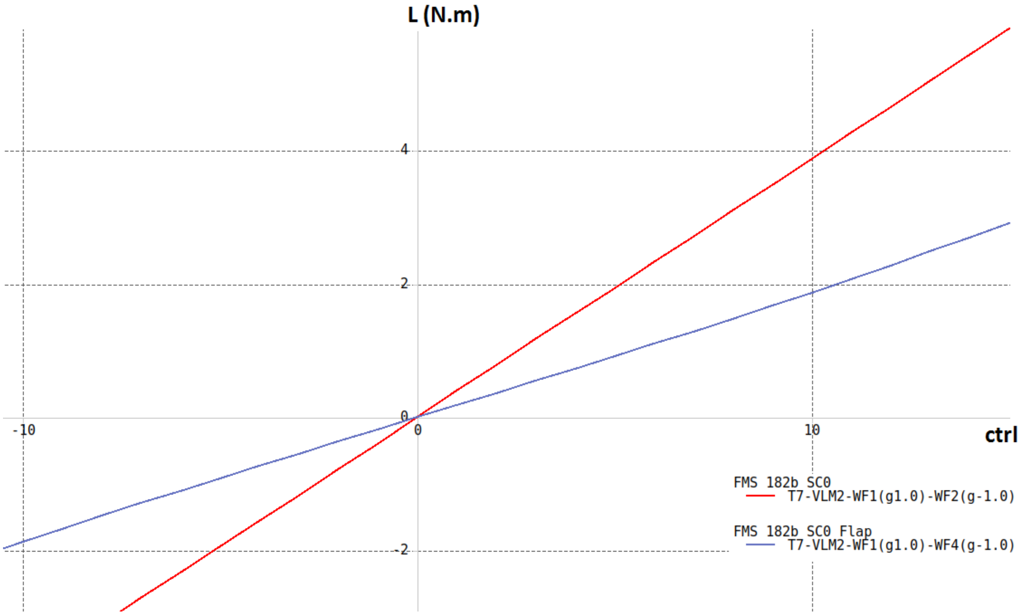


Figure 4.114 Rolling moment fluctuation with control gain for both models

The difference between base and flapped is most likely due to different trim airspeed. As can be seen, the magnitude of the rolling moment per unit deflection is reduced with the flaps deployed. This is expected as control effectiveness is proportional to dynamic pressure ($\frac{1}{2}\rho v^2$). Converting rolling moment (L in Figure 4.114) values to coefficients (*Cl* in Figure 4.115) removes most of the effects associated with airspeed, demonstrating that, especially at lower deflections (up to 10 degrees), the control effectiveness of the ailerons is not only linear, but also identical between configurations. Beyond 10 degrees a small nonlinearity is observed, with flaps increasing the control effectiveness slightly above baseline.

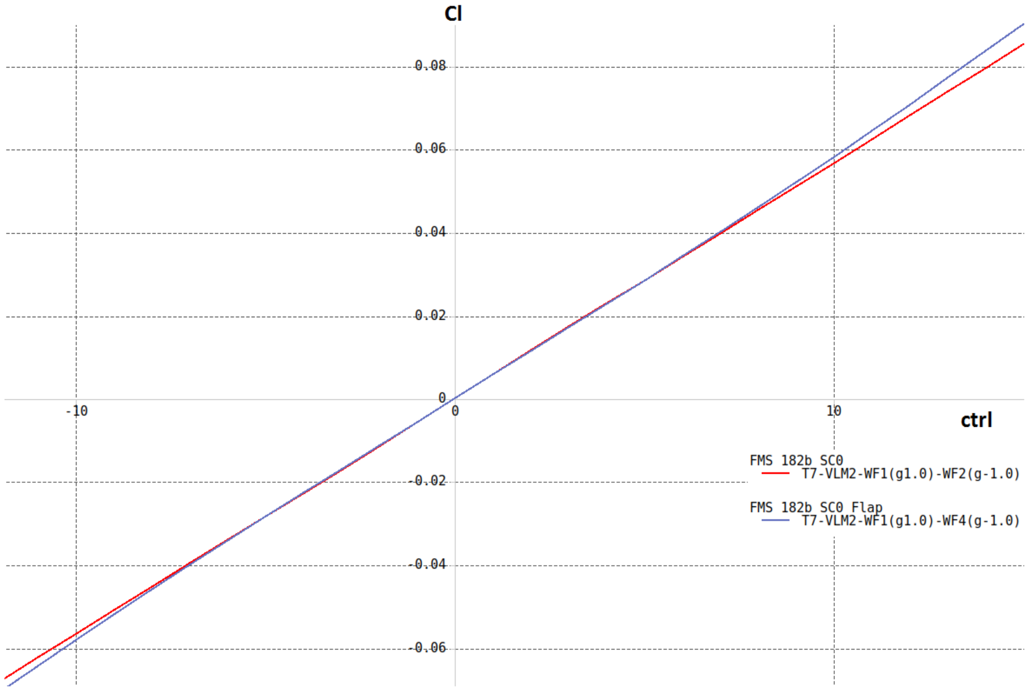


Figure 4.115 Rolling moment coefficient fluctuation with control gain for both models

All planned operations of the FMS 182 are to be kept at fairly low alpha and control deflections (<5°), thus predictions for high alpha and deflections are operationally irrelevant, however they remain important to the analysis.

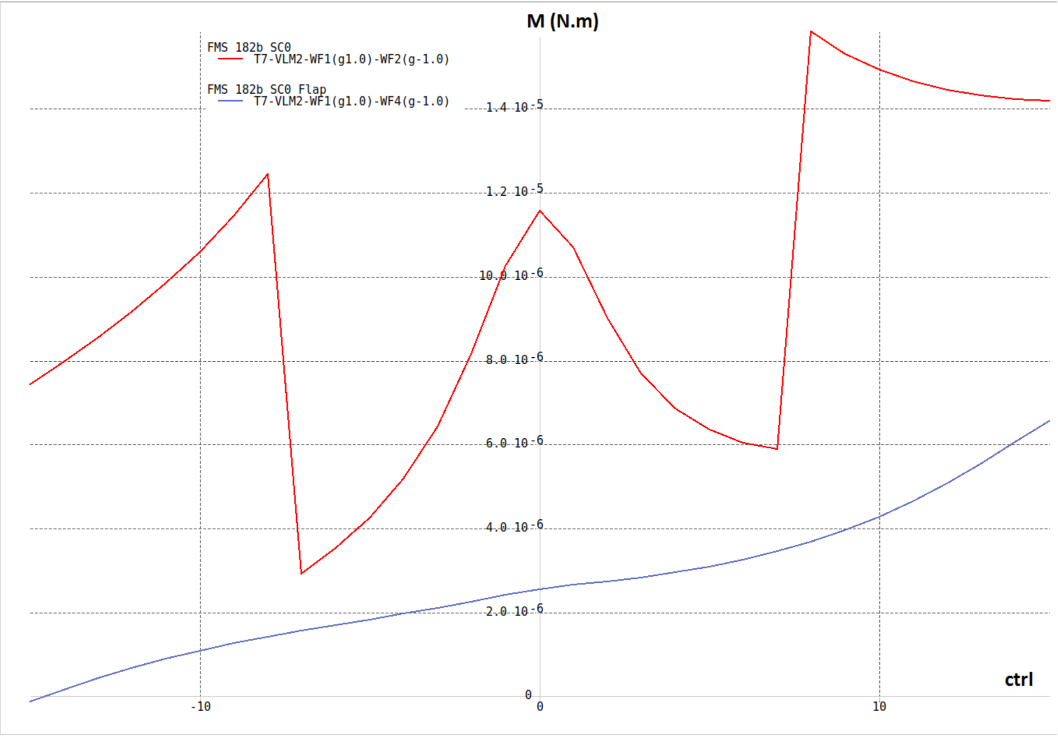


Figure 4.116 Pitching moment fluctuation with control gain for both models

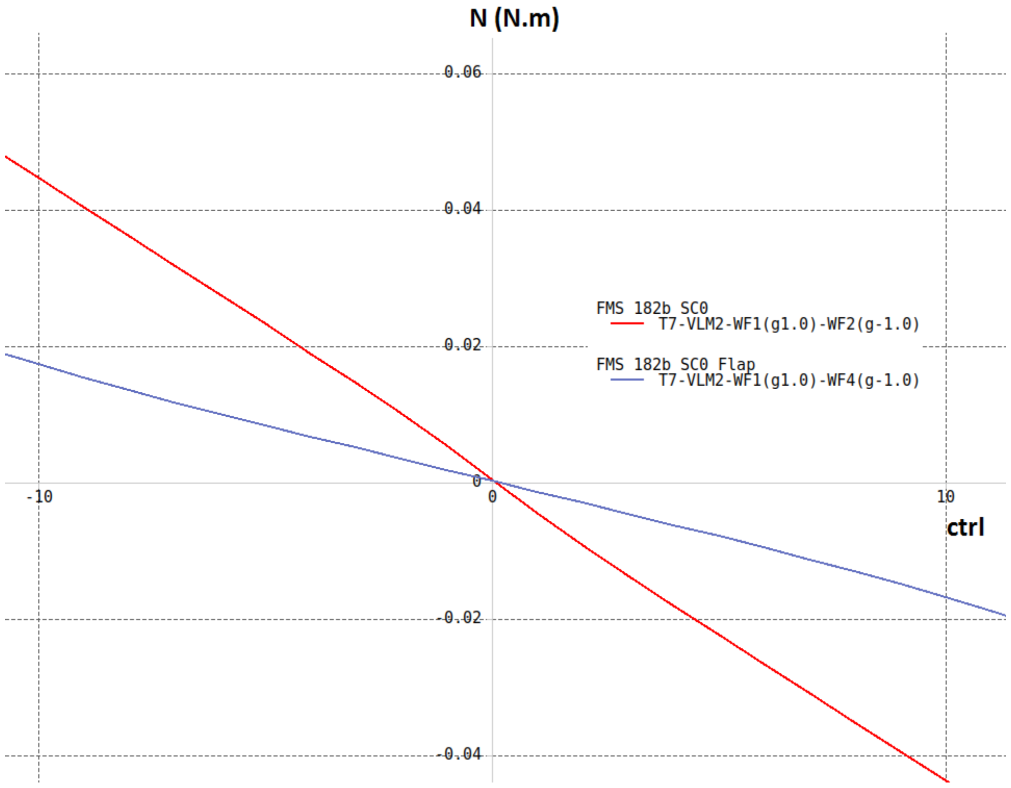


Figure 4.117 Yawing moment fluctuation with control gain for both models

Although there are coupled responses observed (per unit deflection of the control surface) for both the Pitching and Yawing moments, it can be seen from the graphs that their magnitudes are very small. In fact, the pitching moment graph resembles a simulation artifact/noise due to it's magnitude. Nevertheless, this is indicative of better controllability as the application of the ailerons isn't resulting in too much forces/moments in the unintended axises. Based on the obtained results, the average magnitudes for the Rolling moment and it's coefficient was calculated to be as shown below.

Average Lde	Average Clde
21.93906452	0.322493871

From data obtained for Flapped-configuration:

Average Lde (N)	Average Clde
9.9171	0.309181

4.1.9.2 Elevator

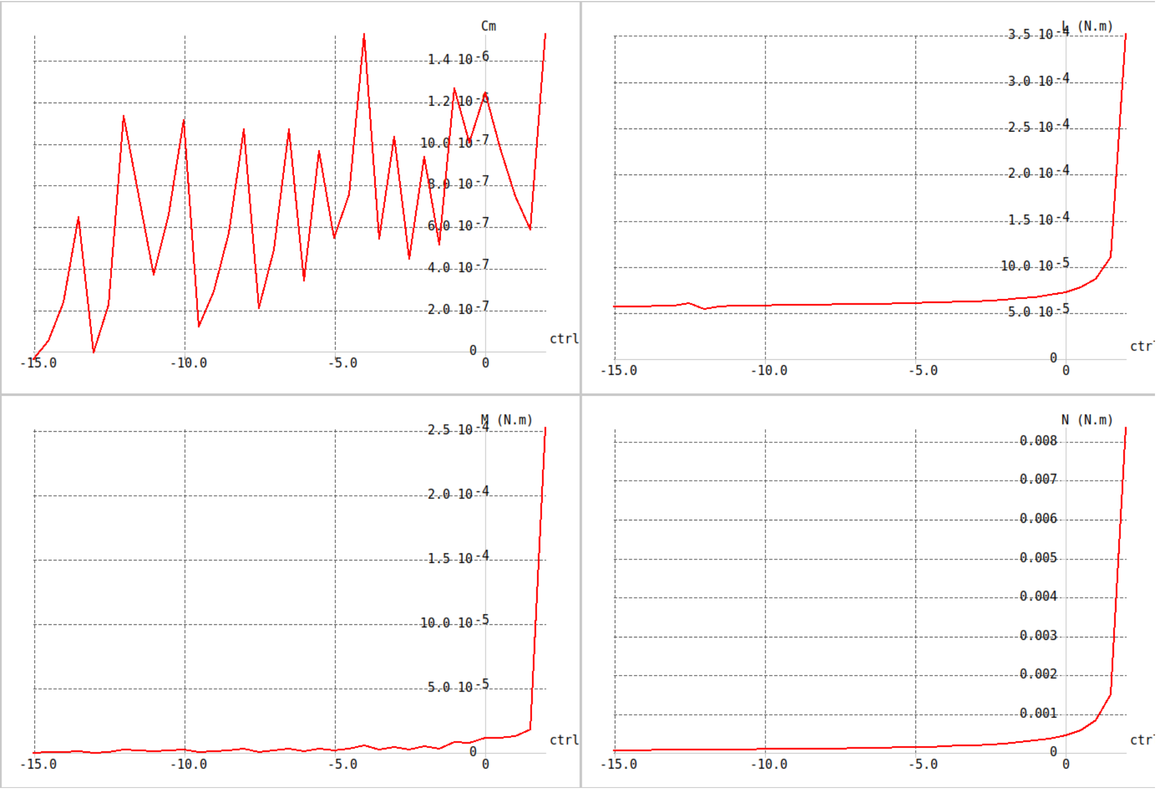


Figure 4.118 T7 test result for FMS 182b SC0 (T7-VLM2)

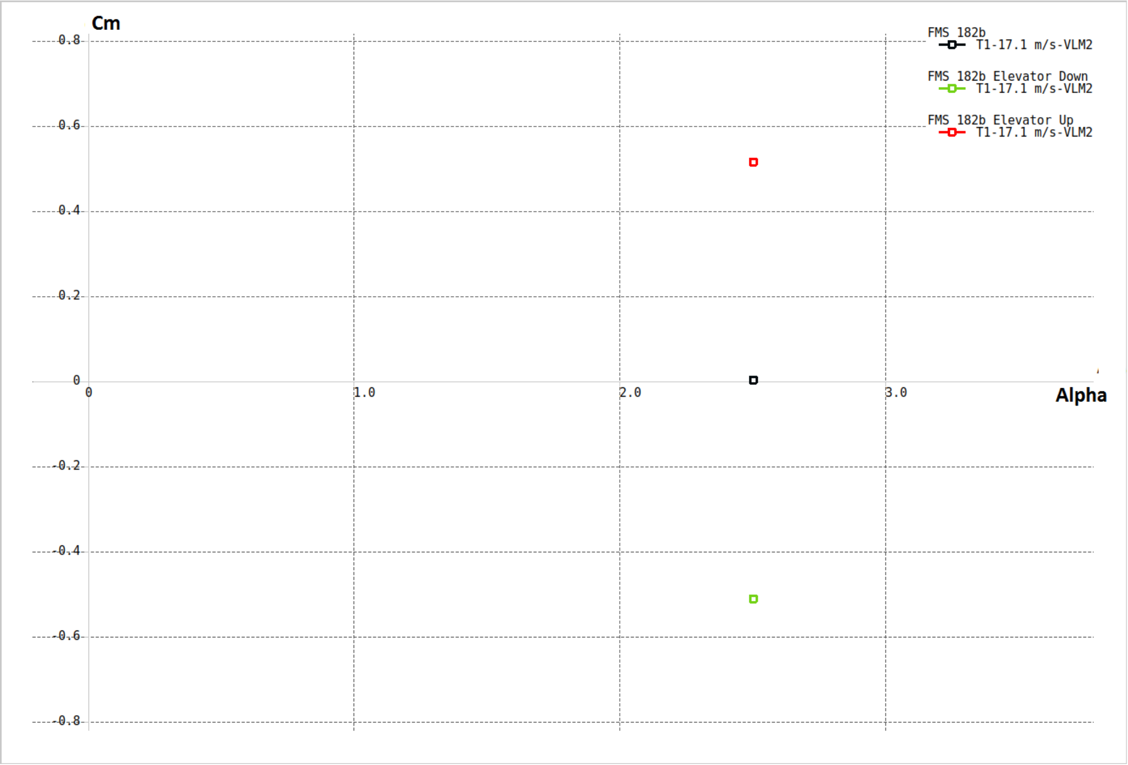


Figure 4.119 Cm against alpha T1 test result for FMS 182b and FMS 182b Elevator Up/Down

As can be seen in Figure 4.118, T7 type analyses run into a fundamental limitation when attempting to measure the control derivative of the elevator. A condition for this type of analysis is flight in pitch trim. As such, adding elevator deflection alters the parameters (airspeed, alpha) of the T7 analysis such that trim is maintained, resulting in a net zero pitching moment which, contradicting the test requirement for an overall pitching moment. XFLR5 offers a static analysis where airspeed and alpha are fixed. This is called the type 1 (T1) analysis. The type 1 analysis was configured to match the trim condition established during the baseline T7 analysis, such that without control deflection, the overall pitching moment coefficient would be zero. 15 degrees of elevator deflection was applied in either direction and the resulting pitching moment change can be seen on (Figure 4.119). For all other control surfaces, it is safe to assume that the pitching moment derivative is very close to linear. Thus, to reduce analysis time only the +15 and -15 degree deflections of the control surfaces were tested.

The plots depicting the rolling moment, pitching moment and yawing moment against the control gain (T1 test) have been generated and are presented in the Appendix (Chapter 7.6.3).

4.1.9.2.1 T1 test results:

Parameter	Value	Unit
XCP	-1124.414	mm
YCP	0.026	mm
ZCP	262.691	mm
CL	0.08499	
CD	0.03327	
VCD	0.02124	
ICD	0.01204	
CX	0.01204	
CY	0.00004	
Cl	0	
Cm	0.51573	
ICm	0.51449	
VCm	0.00124	
Cn	-0.00001	
ICn	-0.00001	
VCn	0	

Table 4-11 T1 Test results

Elevator down 15 degrees:

Parameter	Value	Unit
XCP	277.655	mm
YCP	0.004	mm
ZCP	-40.111	mm
CL	0.45424	
CD	0.03707	
VCD	0.0192	
ICD	0.01786	
CX	0.01786	
CY	0.00003	
Cl	0	
Cm	-0.51141	
ICm	-0.51317	
VCm	0.00176	
Cn	-0.00001	
ICn	-0.00001	
VCn	0	

Table 4-12 T1 test result for Elevator down 15 degrees

Cm per degree from 15 degree upward:

	15 degrees	per degree
Cm	0.51573	0.034382

Cm per degree from 15 degree downward:

	15 degrees	per degree
Cm	-0.51141	-0.03409

Because of how close the values are, the estimation was obtained from their average magnitude as 0.034238.

4.1.9.3 Rudder

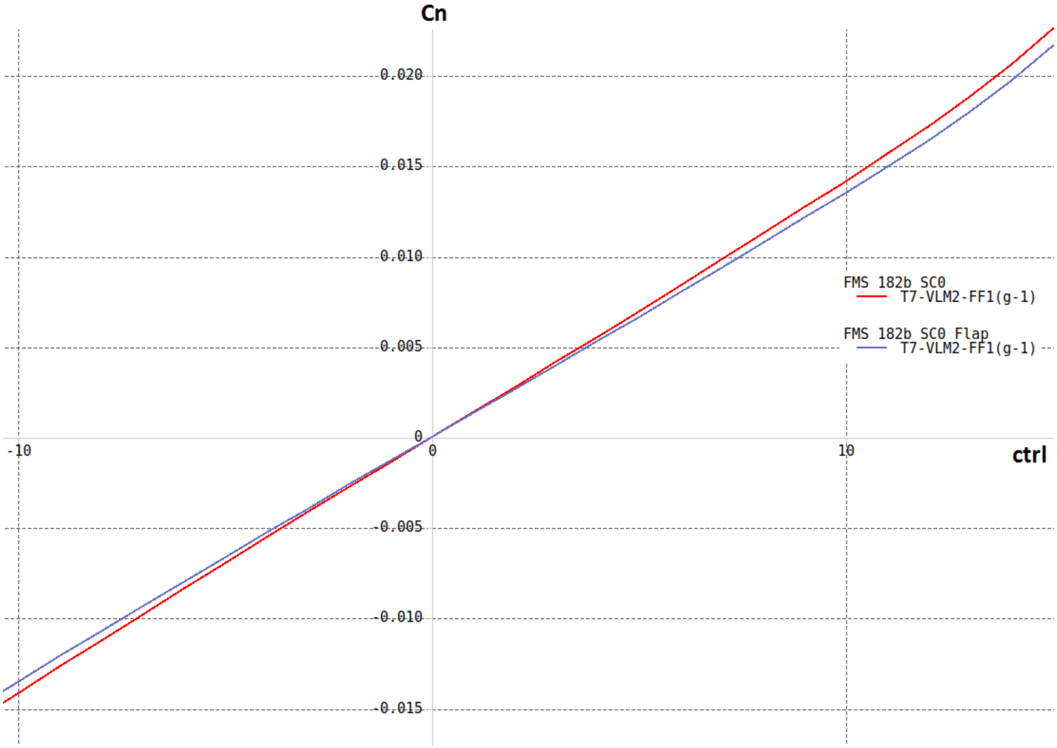


Figure 4.120 Yawing moment coefficient fluctuation with control gain for both models

The yawing moment coefficient demonstrates high degree of linearity for both configurations. The base model shows a slightly higher magnitude change per unit deflection.

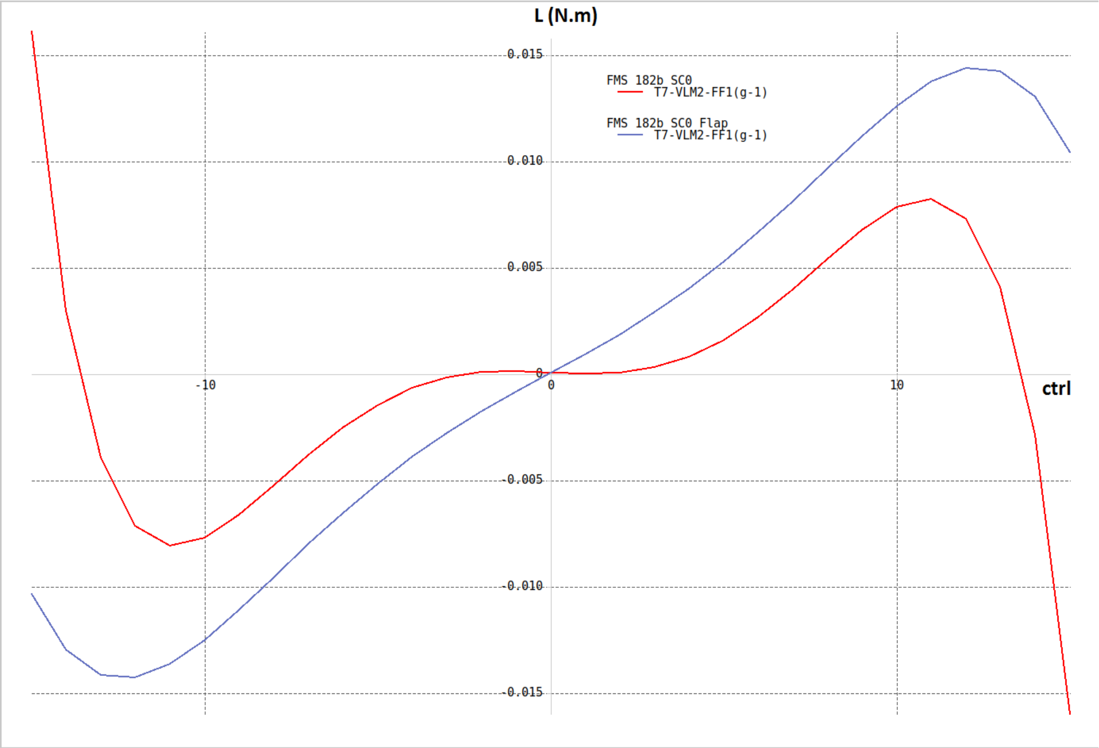


Figure 4.121 Rolling moment fluctuation with control gain for both models

The simulation outputs for rolling moments shows that there is a slight amount of roll response to be expected for rudder deflections. However, its relative magnitude is very small. This undesired coupling seems to be greater with flaps applied.

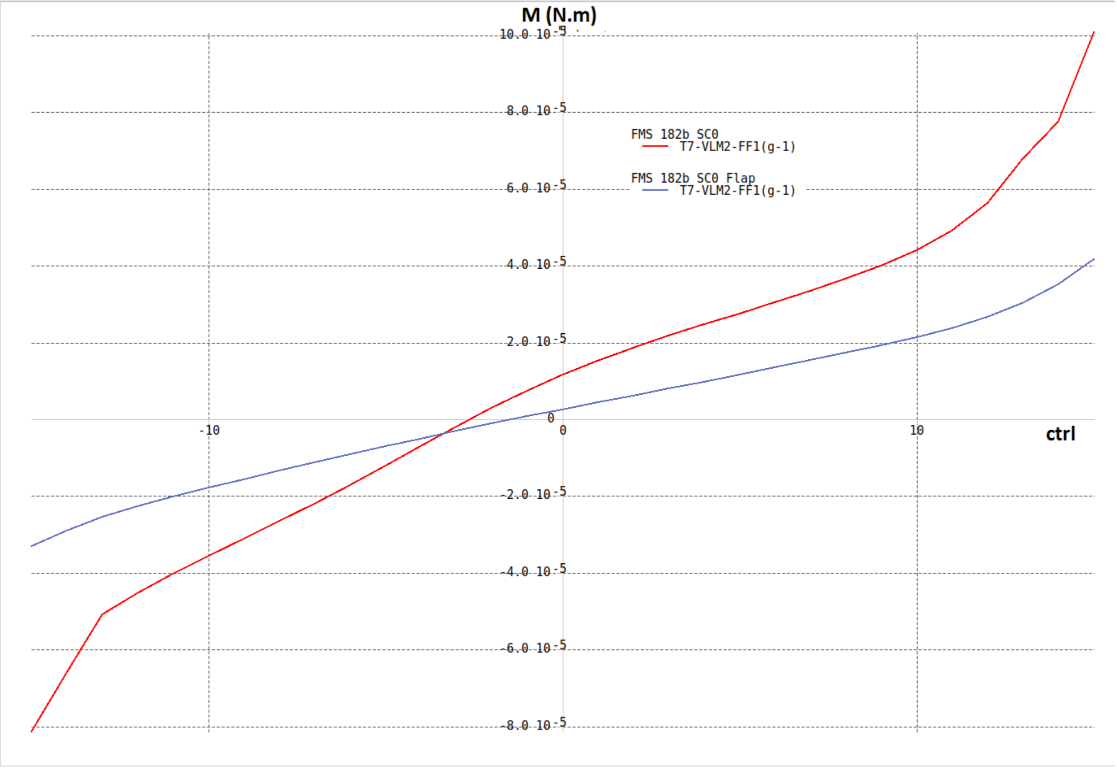


Figure 4.122 Pitching moment fluctuation with control gain for both models

Application of the rudder also induces a slight pitching moment (greater in case of the base model). However the magnitudes are relatively negligible.

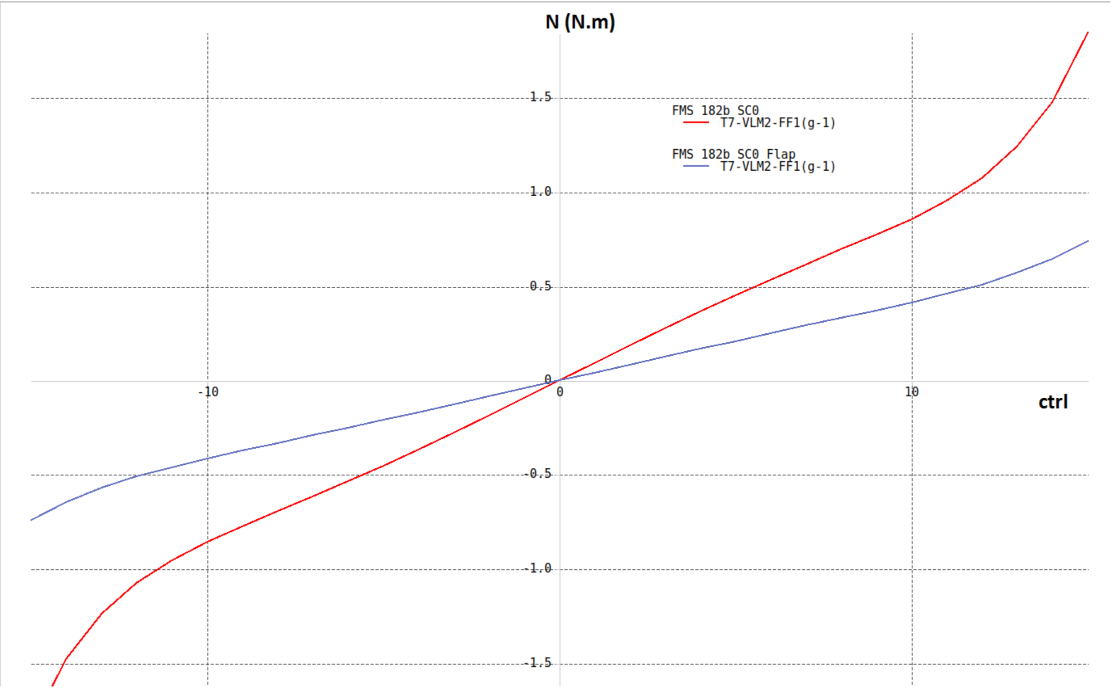
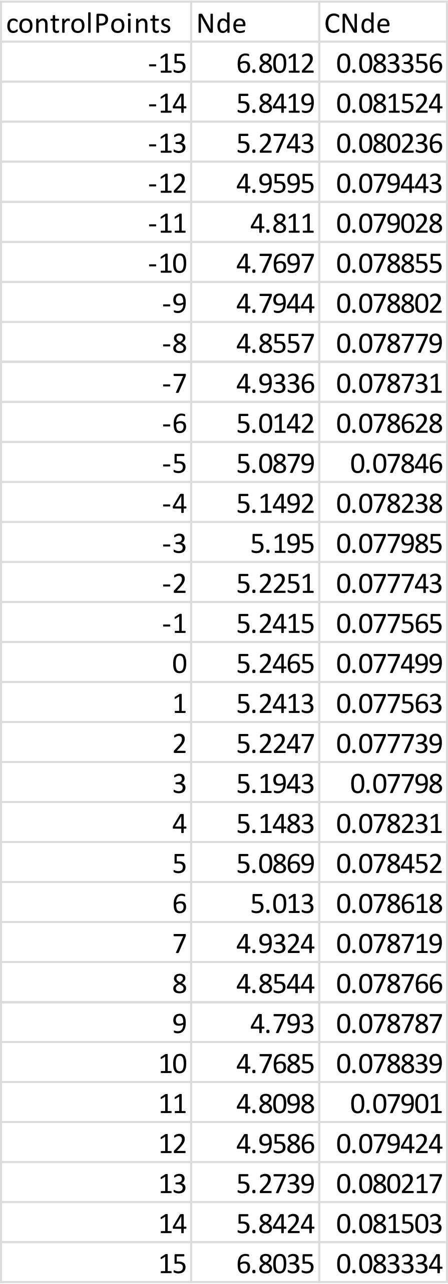


Figure 4.123 Yawing moment fluctuation with control gain for both models

As expected, the greatest difference made can be seen in the above graph for the yawing moment. Appropriate level of linearity is maintained for both configurations until 10 degrees of rudder deflection. Beyond that, the magnitude increases rapidly. The application of flaps is expected to reduce the magnitude of the yaw force generated per unit deflection of the rudder compared to the non-flapped configuration.

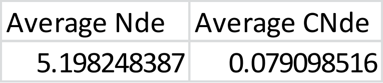
4.1.9.3.1 Control deflection test results:



controlPoints	Nde	CNde
-15	6.8012	0.083356
-14	5.8419	0.081524
-13	5.2743	0.080236
-12	4.9595	0.079443
-11	4.811	0.079028
-10	4.7697	0.078855
-9	4.7944	0.078802
-8	4.8557	0.078779
-7	4.9336	0.078731
-6	5.0142	0.078628
-5	5.0879	0.07846
-4	5.1492	0.078238
-3	5.195	0.077985
-2	5.2251	0.077743
-1	5.2415	0.077565
0	5.2465	0.077499
1	5.2413	0.077563
2	5.2247	0.077739
3	5.1943	0.07798
4	5.1483	0.078231
5	5.0869	0.078452
6	5.013	0.078618
7	4.9324	0.078719
8	4.8544	0.078766
9	4.793	0.078787
10	4.7685	0.078839
11	4.8098	0.07901
12	4.9586	0.079424
13	5.2739	0.080217
14	5.8424	0.081503
15	6.8035	0.083334

Table 4-13 Variation in derivatives for the control gains (SC0)

Based on the data set, the averages were calculated as follows.



Average Nde	Average CNde
5.198248387	0.079098516

4.1.9.3.2 Flapped

The same tests were conducted on the flapped model variant to allow for sufficient data to factor in the variations in the derivatives from the use of simulated flaps. However, that data is not being presented here but is made available in the Appendix (Chapter 7.6.3.1). The range of stability and control derivatives obtained here for a variety of conditions allow for improved modelling and simulation opportunities for the MATLAB/Simulink model because any future mission altering the payload and/or CG can benefit from this analysis and the MATLAB/Simulink can be updated with more appropriate estimations for these important parameters.

4.2 X-Plane Modelling & Simulation

4.2.1 Aerofoil graphs

The following are the graphs for the aerofoil file created by the method described earlier in the X-Plane methods section. The results were visualised through the Airfoil Maker software that's included with X-Plane. The horizontal axis represents -20 to +20 angle of attack. In the following screenshots, the parameters on the left should all be ignored as they only matter if this file is edited and saved inside Airfoil Maker. The emphasis should be on the graphs themselves and their identifiers on the bottom left.

For the sake of demonstrating the success of the methods used, only a handful of graphs depicting the generated foils are being shown here. They are the low and high (100k & 400k) Reynolds number foils generated for the main wing. It is also important to note that the x-axis on the Airfoil Maker plots is always range from -20 degrees to + 20 degrees from the centreline.

4.2.1.1 Aerofoil polars for FMS 1214 low Reynolds number:

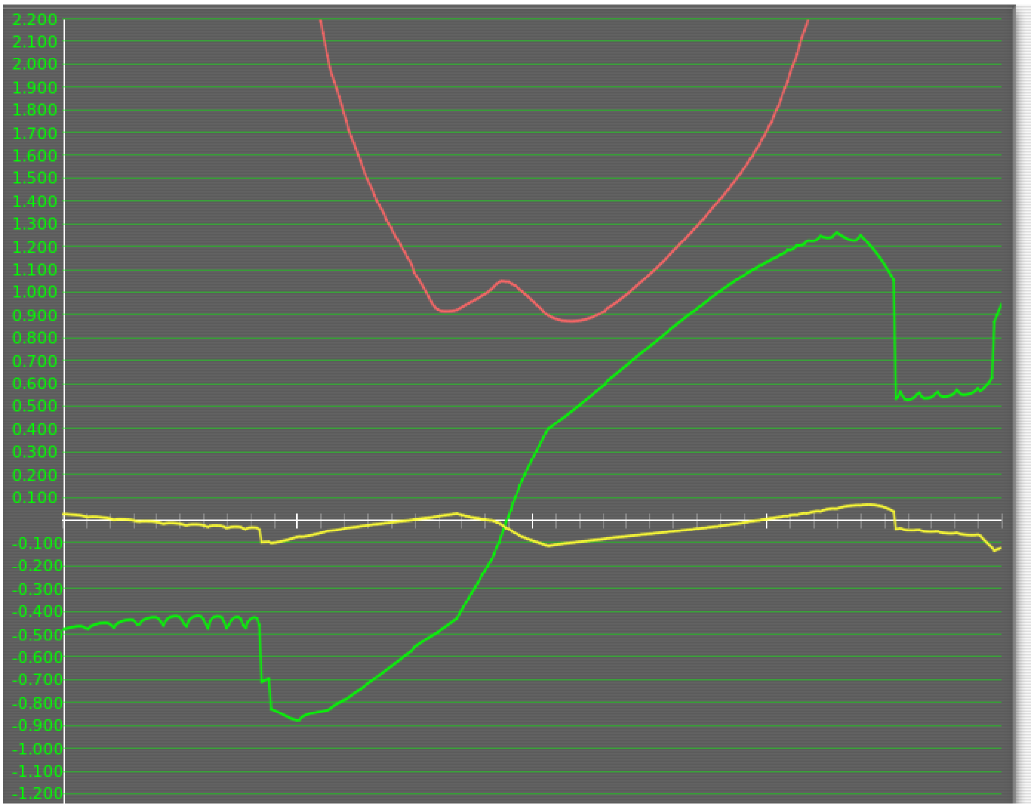


Figure 4.124 Aerofoil polar FMS 1214 Re 100k.afl (green = Cl, red = Cd, yellow = Cm)

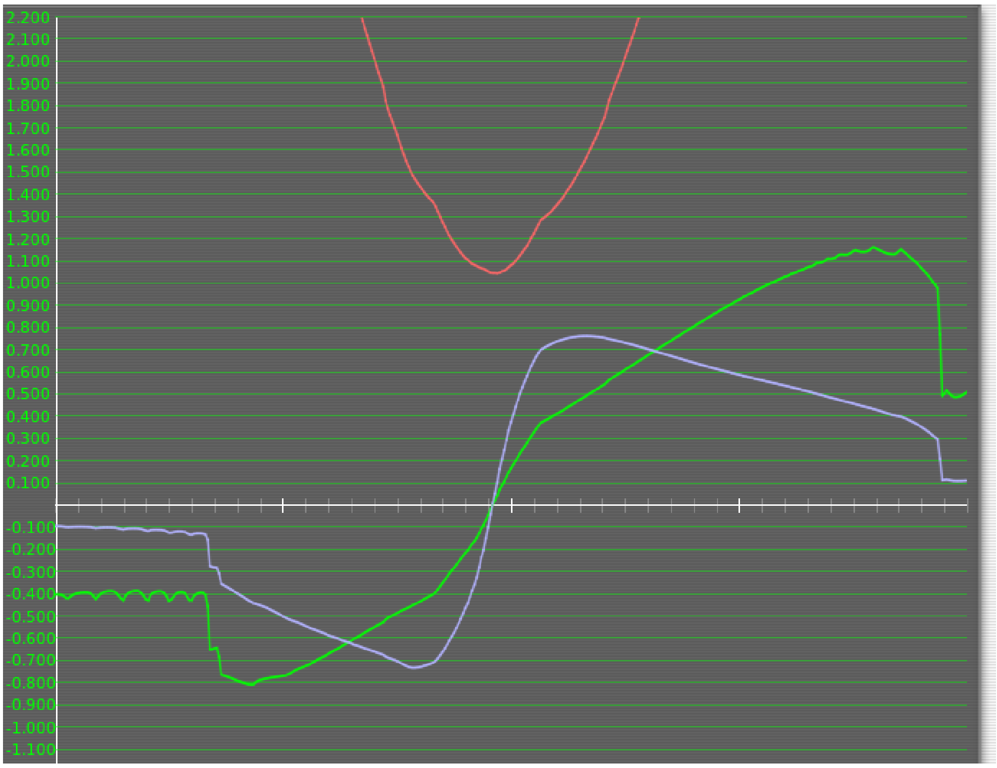


Figure 4.125 Aerofoil polar FMS 1214 Re 100k.afl (green = Cl, red = Cd, purple = L/D)

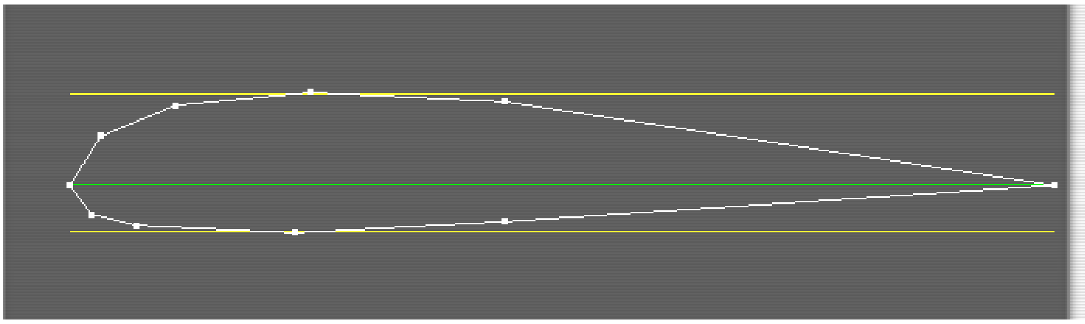


Figure 4.126 Aerofoil FMS 1214 Re 100k.afl foil

4.2.1.2 Aerofoil polars for FMS 1214 high Reynolds number:

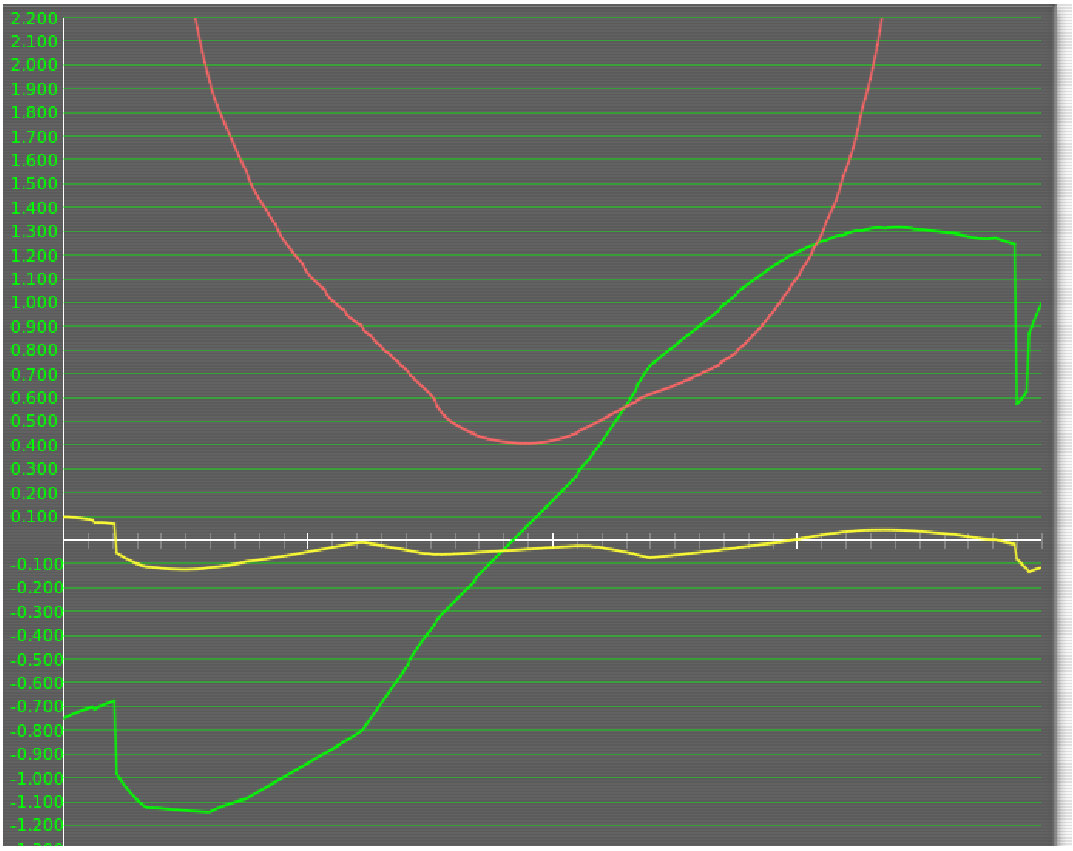


Figure 4.127 Aerofoil polar FMS 1214 Re 400k.afl (green = Cl, red = Cd, yellow = Cm)

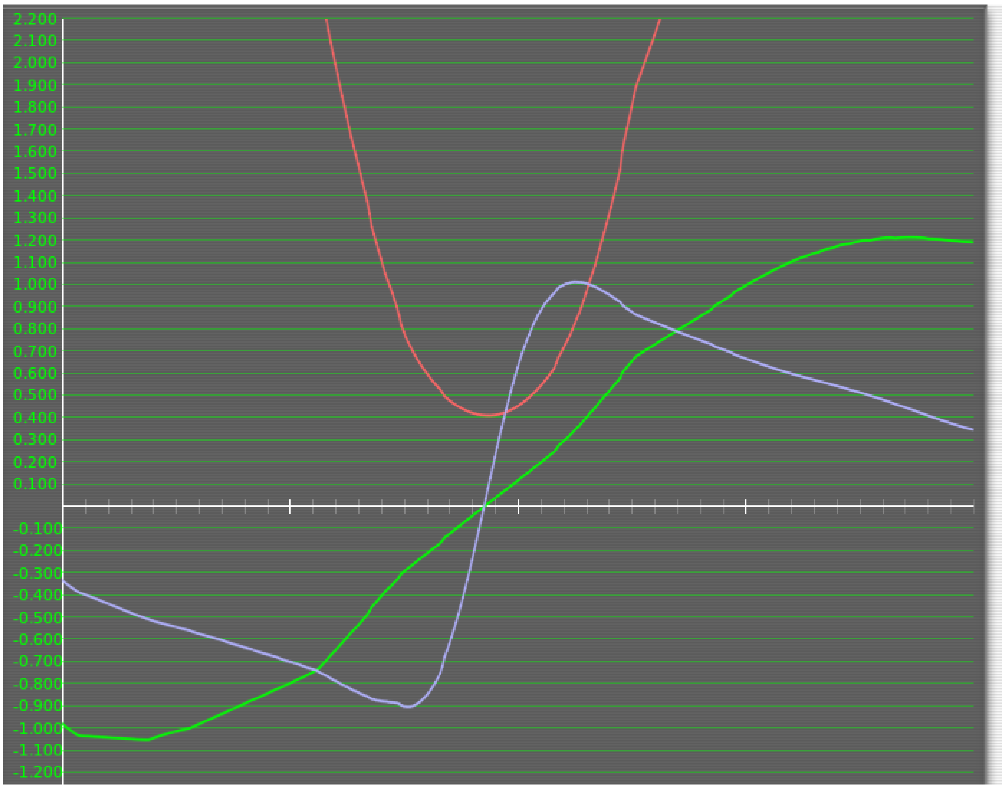


Figure 4.128 Aerofoil polar FMS 1214 Re 400k.afl (green = Cl, red = Cd, purple = L/D)

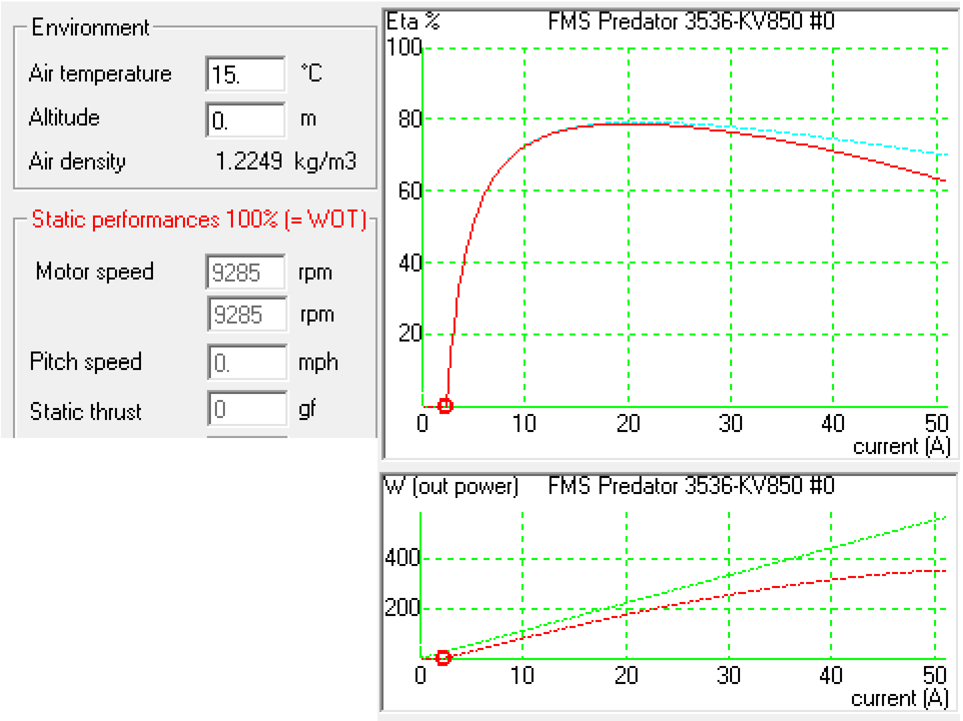
For every section of the X-Plane model requiring a custom aerofoil, the proposed methods were utilised and appropriate foils were generated and used. A range of such foils that are not shown above are presented in the Appendix (Chapter 7.6.4).

4.2.2 Power plant results

The following are the results obtained from Scorpion_calc and physical testing of the power plant setup.

4.2.2.1 Scorpion_calc estimations:

In the absence of any propeller loading acting on the motor, based on our setup, the software predicts a peak max power output of 400 Watts around 50 amp current supplied. This is reasonably close to the 380 Watts peak power mentioned in the datasheet for the FMS Predator 3536-KV850 motor. For the loaded situation, the following results were obtained.



Environment

Air temperature 15. °C

Altitude 0. m

Air density 1.2249 kg/m3

Static performances 100% (= WOT)

Motor speed 9285 rpm

9285 rpm

Pitch speed 0. mph

Static thrust 0 gf

Figure 4.129 Scorption_calc power output prediction

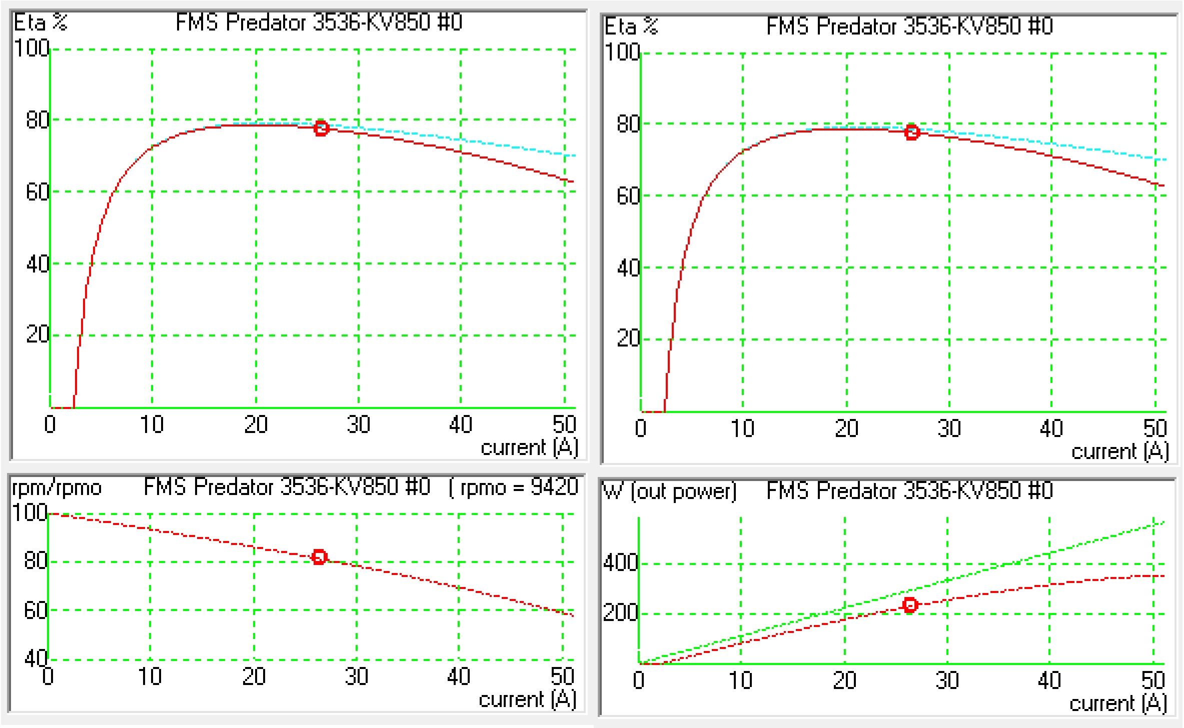
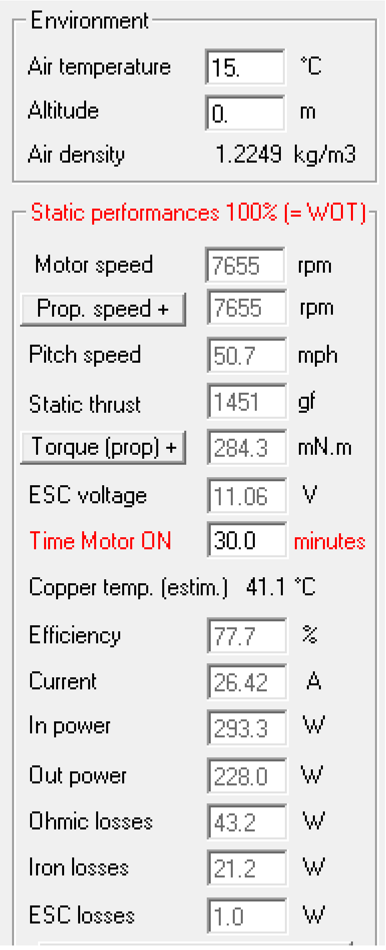


Figure 4.130 Scorption_calc result for RPM and power output



Environment

Air temperature 15. °C

Altitude 0. m

Air density 1.2249 kg/m3

Static performances 100% (= WOT)

Motor speed 7655 rpm

Prop. speed + 7655 rpm

Pitch speed 50.7 mph

Static thrust 1451 gf

Torque (prop) + 284.3 mN.m

ESC voltage 11.06 V

Time Motor ON 30.0 minutes

Copper temp. (estim.) 41.1 °C

Efficiency 77.7 %

Current 26.42 A

In power 293.3 W

Out power 228.0 W

Ohmic losses 43.2 W

Iron losses 21.2 W

ESC losses 1.0 W

Figure 4.131 Scorpion_cal static test result

The above pictures display the results for static test situations for the setup in mean sea level conditions. The sustained current of 26.42 amps is close to the datasheet provided with the motor and the obtained results therefore were used to get an estimation of the horsepower value needed to be entered into Plane Maker (0.3-0.4 HP).

4.2.2.2 Physical test results:

While all the parameters obtained from Scorpion_calc could not be checked against the power plant physical tests, it wasn't a problem for this modelling as most of them cannot be modelled in X-Plane.

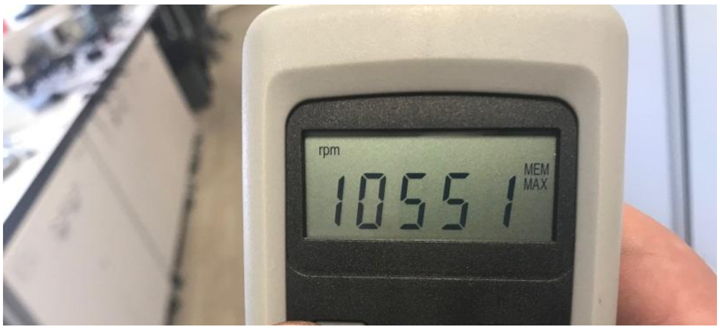


Figure 4.132 Tachometer reading for static test (no load)

The static test result produced a peak RPM of 10551. Sciorpion_cal estimated this value at 9285 (no load test).

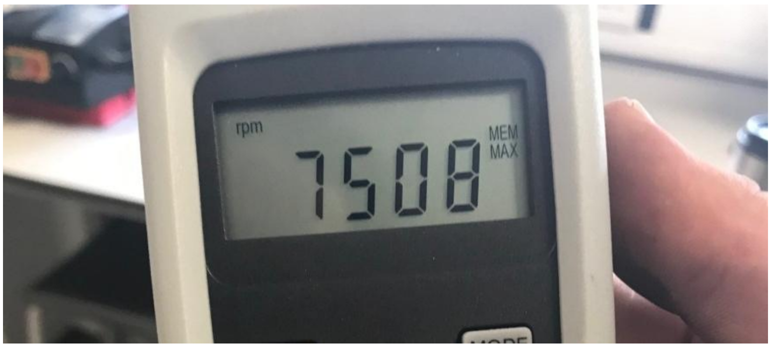


Figure 4.133 Tachometer reading for loaded situation

The peak RPM measured for the static test with propeller load was 7508. Scorpion_calc estimated this value to be 7655. These results that are in reasonable agreement with each other are what was used to specify the max RPM and engine power (converted to horsepower) in Plane Maker for X-Plane modelling. It should also be noted that the static thrust of 1451 gram-force is also reasonably close to the 1203 gram-force (1.203 kg) measured by the scale on which the entire setup was mounted.

4.2.3 Component modelling results

The process of calculating the 11 stations with the leading and trailing edge offsets and individual incidence angles described in the methods section led to reasonably accurate modelling of the propeller and spinner geometry.

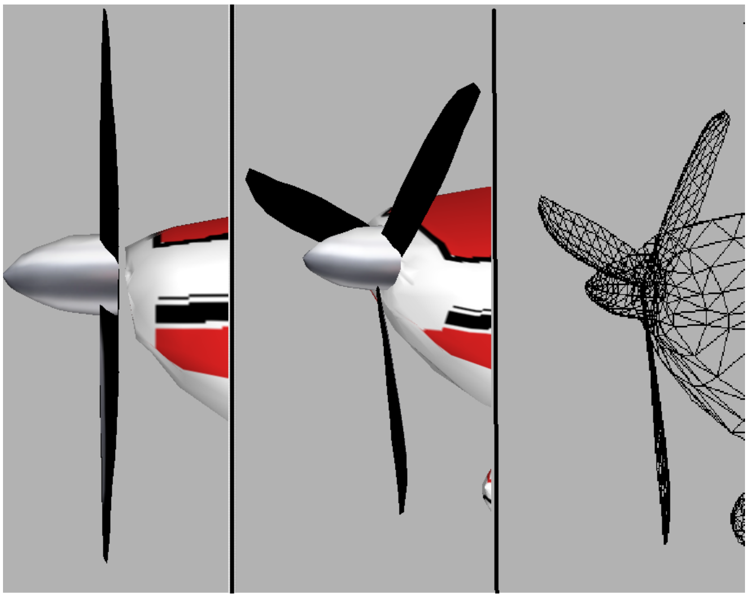


Figure 4.134 Propeller and spinner designed in Plane Maker

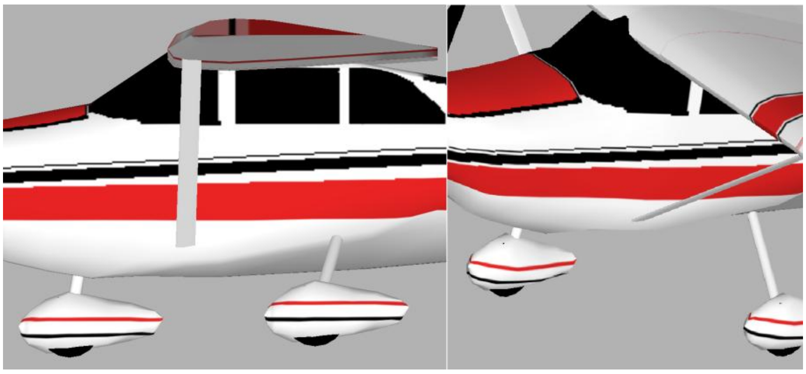


Figure 4.135 Wing struts and landing gears designed in Plane Maker

The wing struts and landing gear (along with gear-legs) were modelled based on the procedure explained in the methods section.

4.2.4 Finished model

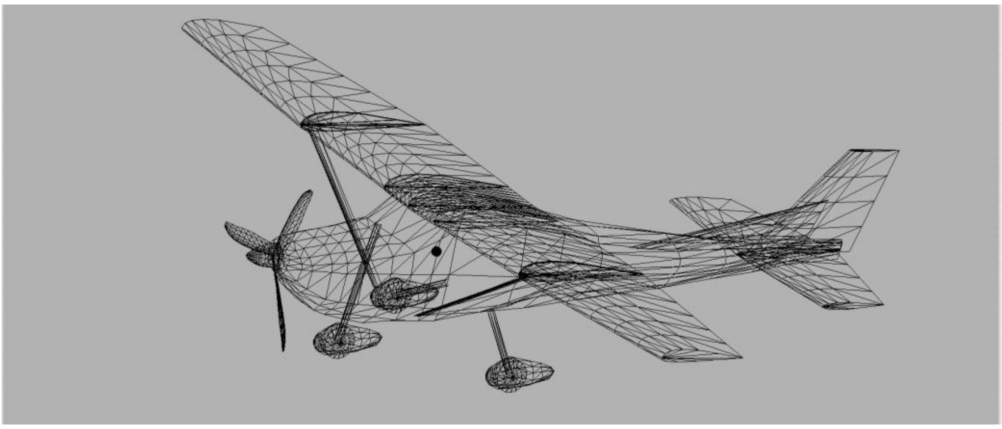


Figure 4.136 Wire-mesh view of the FMS 182 Plane Maker model

Following the procedures outline in the corresponding X-Plane methods section, the entire aircraft was successfully modelled in plane maker. The following images show the finished final model fully textured in Plane Maker.

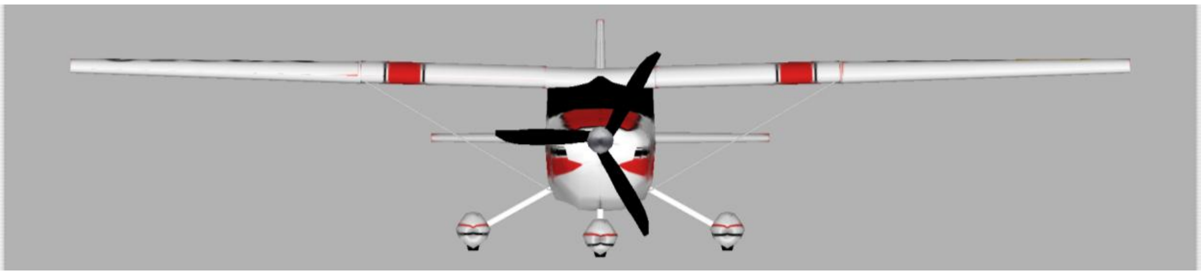


Figure 4.137 FMS 182 Plane Maker model (front view)

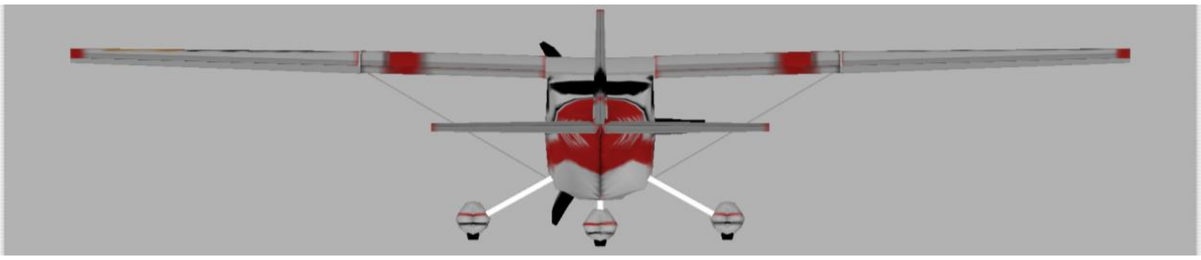


Figure 4.138 FMS 182 Plane Maker model (rear view)

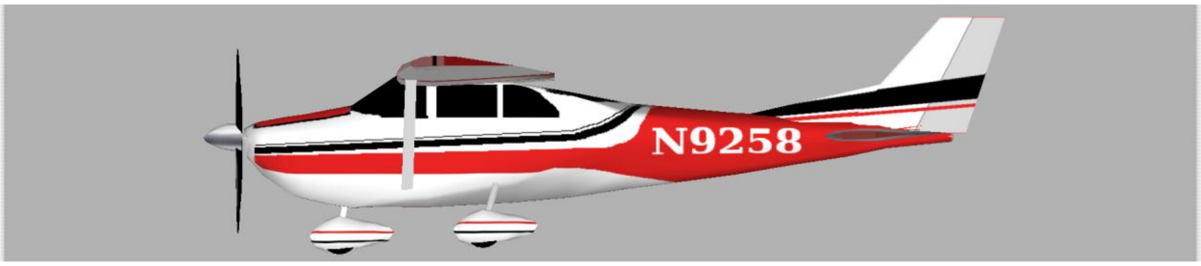


Figure 4.139 FMS 182 Plane Maker model (side view)

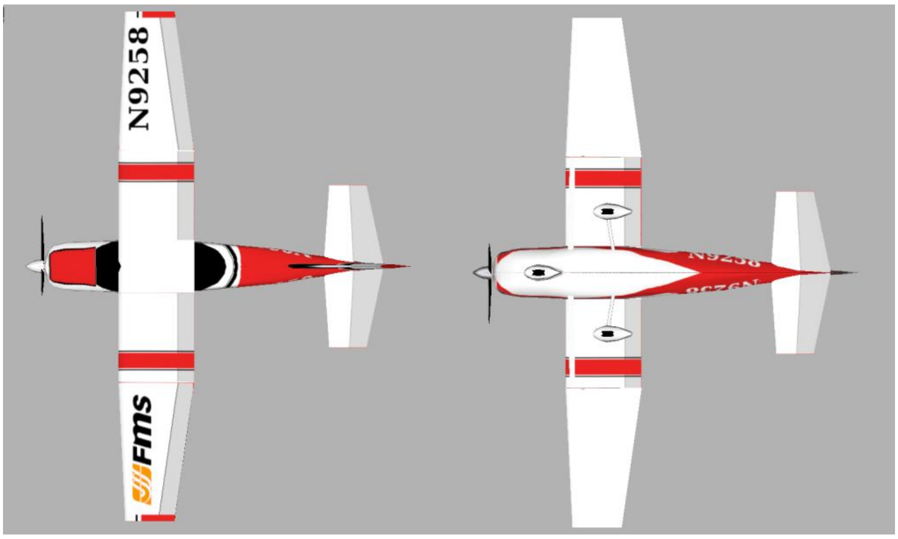


Figure 4.140 FMS 182 Plane Maker model top(left) and bottom(right) views

The modelled aircraft loads successfully into the X-Plane environment without any errors.

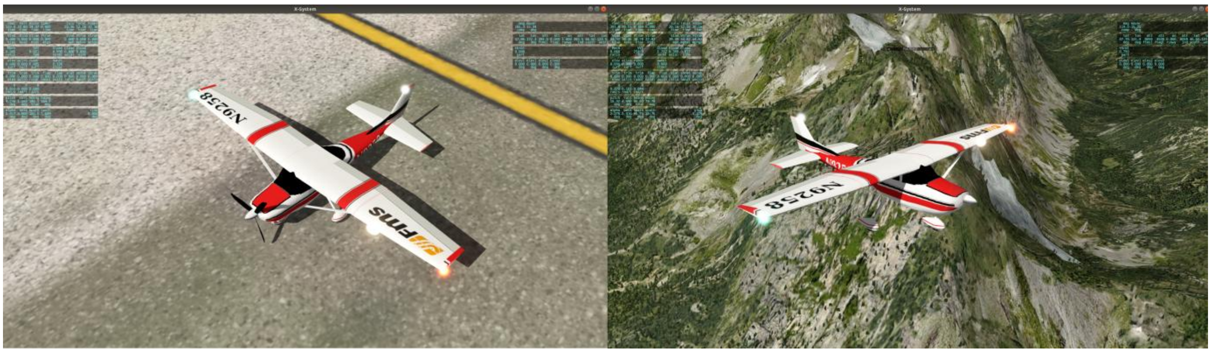


Figure 4.141 FMS 182 X-Plane model operating in X-Plane simulation environment.

The following shows the Plane Maker model operating inside the X-Plane simulated environment with the flight models made visible.

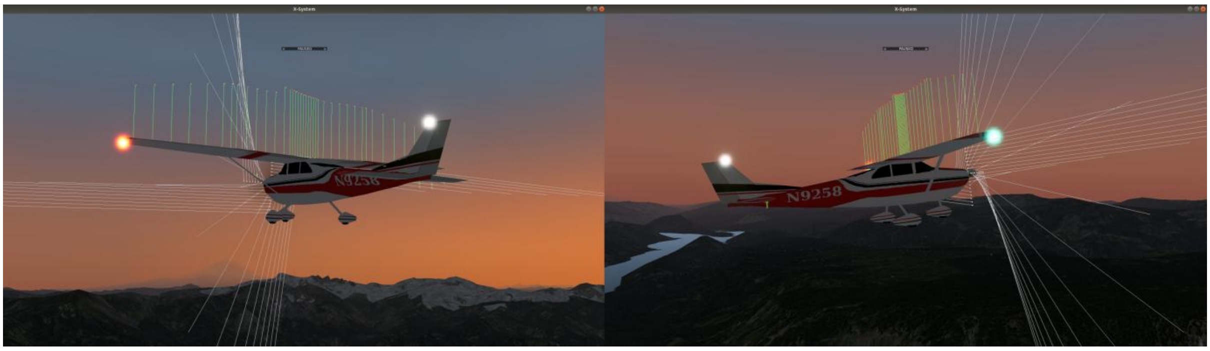


Figure 4.142 X-Plane flight model depicting forces on the wings and propellers

This is a visual way of checking the current flight model's simulation outputs. The white lines depicted in front of the aircraft are force lines relating to the propeller and the green lines on top of the wing are the resultant lift lines.

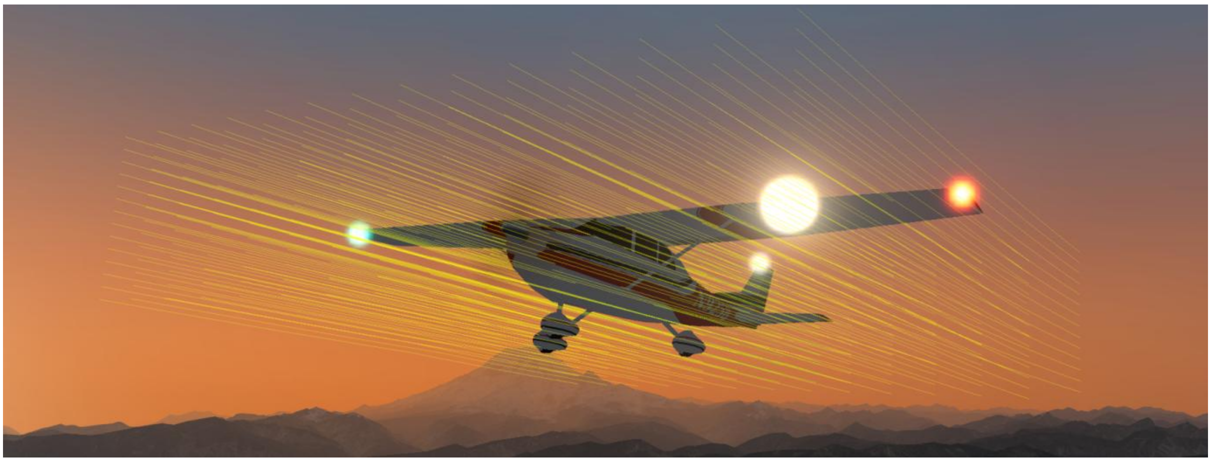


Figure 4.143 X-Plane flight model depicting airflow

Due to the test flight conditions using a steady environment (no wind), the simulated airspace is modelled as steady uniform airstream as depicted in the above image.

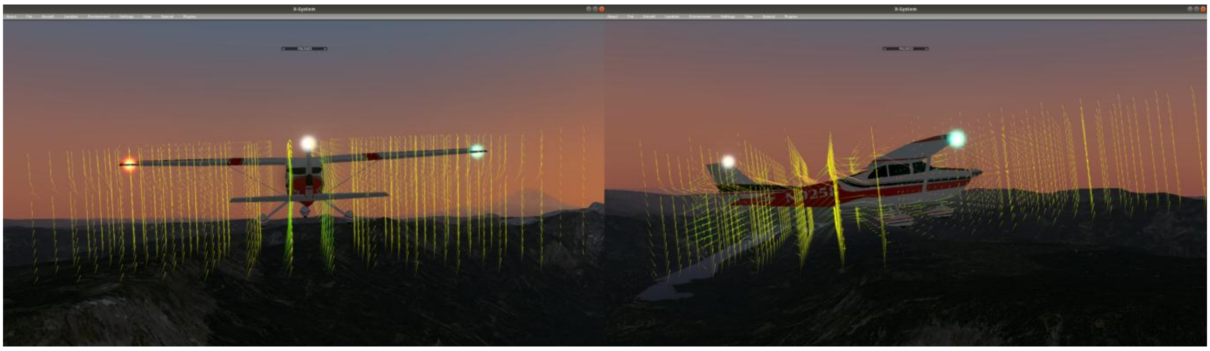


Figure 4.144 X-Plane flight model depicting flow field and model interaction

The field of simulated air around the vehicle however interacts with the modelled airplane as shown in Figure 4.144. The types of forces and moments generated as a result go on to determine the dynamics of the airplane. The following section shows the type of aerodynamic data this flight model uses in graphical form.

4.2.5 Flight Tests

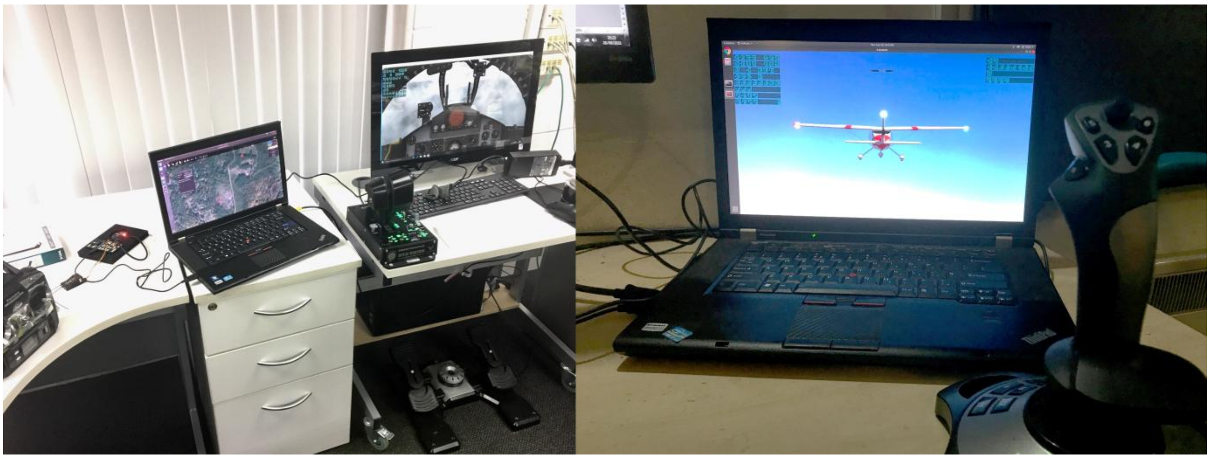


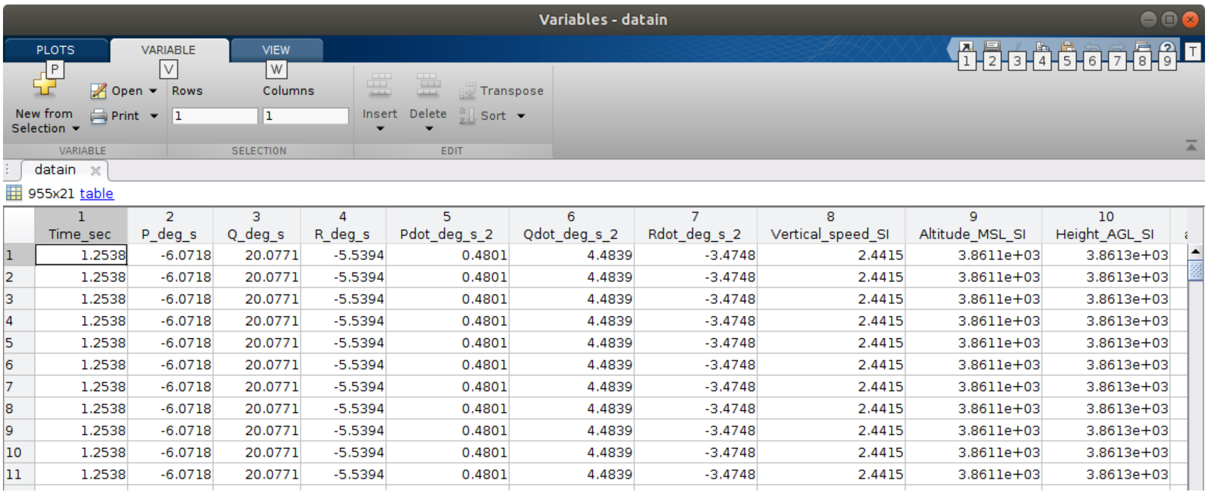
Figure 4.145 X-Plane flight tests at two different test setups

The completed model was flown around in X-Plane both on the Linux based mobile workstation (ThinkPad W520 running X-Plane 10) and Windows based dedicated X-Plane simulator situation at Sheaf 4114 lab at Sheffield Hallam University. The following general observations were made:

- The aircraft requires very little take-off runway.
- The throttle needs reduction and persistent low inputs for most flight operations otherwise controllability becomes a problem.
- Most large moments predicted by XFLR5 for full control deflection holds true in the X-Plane simulation.
- The FMS SkyTrainer 182 model is highly responsive to control inputs. While this is a good sign of correct modelling for this type of an RC platform, this also results in hardship for precise manoeuvring.
- The finished model is unstable in roll. Despite tweaking the aerofoils, wing designs, dihedral, etc., all efforts failed in getting the aircraft to sustain a trimmed level flight without eventually rolling. Multiple different test platforms (different OS's and versions of X-Plane) were used to check the consistency of this problem and it proved to not be solvable. It should be noted that this problem is not limited to this model in X-Plane. Various other full-scale fixed-wing aircraft models do exhibit this problem to varying degrees while their real world counterparts don't have that instability. This may be a limitation of the simulator that is to be factored in until the developers fix the issue.

4.2.6 MATLAB data import results

The following are the results from the methods described earlier.

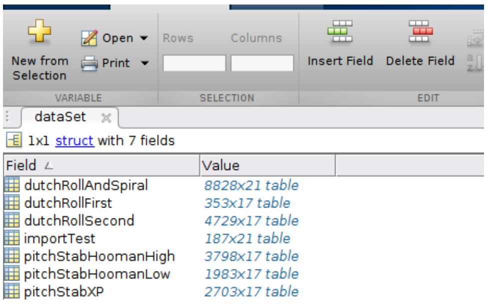


955x21 table

	1 Time_sec	2 P_deg_s	3 Q_deg_s	4 R_deg_s	5 Pdot_deg_s_2	6 Qdot_deg_s_2	7 Rdot_deg_s_2	8 Vertical_speed_SI	9 Altitude_MSL_SI	10 Height_AGL_SI
1	1.2538	-6.0718	20.0771	-5.5394	0.4801	4.4839	-3.4748	2.4415	3.8611e+03	3.8613e+03
2	1.2538	-6.0718	20.0771	-5.5394	0.4801	4.4839	-3.4748	2.4415	3.8611e+03	3.8613e+03
3	1.2538	-6.0718	20.0771	-5.5394	0.4801	4.4839	-3.4748	2.4415	3.8611e+03	3.8613e+03
4	1.2538	-6.0718	20.0771	-5.5394	0.4801	4.4839	-3.4748	2.4415	3.8611e+03	3.8613e+03
5	1.2538	-6.0718	20.0771	-5.5394	0.4801	4.4839	-3.4748	2.4415	3.8611e+03	3.8613e+03
6	1.2538	-6.0718	20.0771	-5.5394	0.4801	4.4839	-3.4748	2.4415	3.8611e+03	3.8613e+03
7	1.2538	-6.0718	20.0771	-5.5394	0.4801	4.4839	-3.4748	2.4415	3.8611e+03	3.8613e+03
8	1.2538	-6.0718	20.0771	-5.5394	0.4801	4.4839	-3.4748	2.4415	3.8611e+03	3.8613e+03
9	1.2538	-6.0718	20.0771	-5.5394	0.4801	4.4839	-3.4748	2.4415	3.8611e+03	3.8613e+03
10	1.2538	-6.0718	20.0771	-5.5394	0.4801	4.4839	-3.4748	2.4415	3.8611e+03	3.8613e+03
11	1.2538	-6.0718	20.0771	-5.5394	0.4801	4.4839	-3.4748	2.4415	3.8611e+03	3.8613e+03

Figure 4.146 MATLAB data import

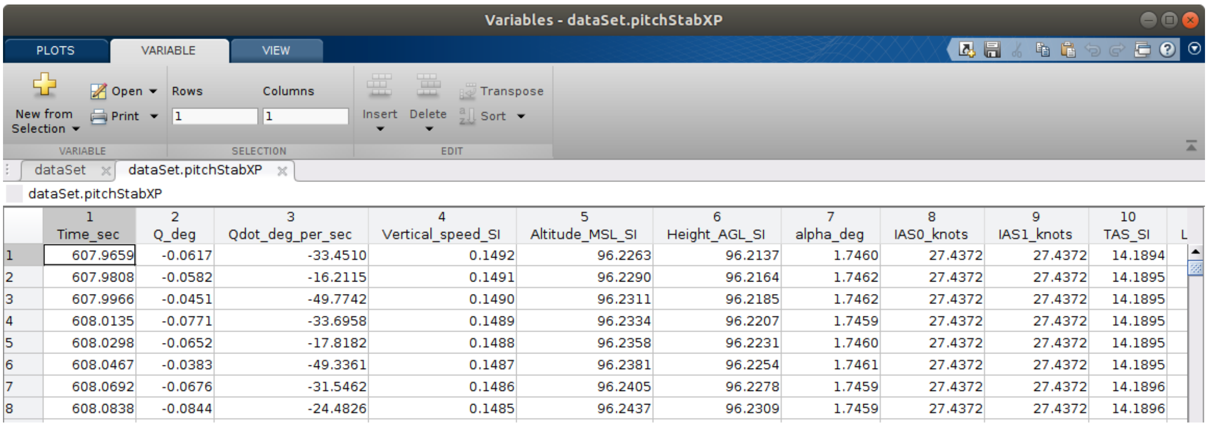
As stated earlier, the modifications and extensions to the JavaScript allowed for flight data from X-Plane to not only be captured but imported right into MATLAB's working memory in the format of a table with headers identifying the parameter type.



1x1 struct with 7 fields

Field	Value
dutchRollAndSpiral	8828x21 table
dutchRollFirst	353x17 table
dutchRollSecond	4729x17 table
importTest	187x21 table
pitchStabHoomanHigh	3798x17 table
pitchStabHoomanLow	1983x17 table
pitchStabXP	2703x17 table

Figure 4.147 MATLAB flight data storage in structures



dataSet.pitchStabXP

	1 Time_sec	2 Q_deg	3 Qdot_deg_per_sec	4 Vertical_speed_SI	5 Altitude_MSL_SI	6 Height_AGL_SI	7 alpha_deg	8 IAS0_knots	9 IAS1_knots	10 TAS_SI
1	607.9659	-0.0617	-33.4510	0.1492	96.2263	96.2137	1.7460	27.4372	27.4372	14.1894
2	607.9808	-0.0582	-16.2115	0.1491	96.2290	96.2164	1.7462	27.4372	27.4372	14.1895
3	607.9966	-0.0451	-49.7742	0.1490	96.2311	96.2185	1.7462	27.4372	27.4372	14.1895
4	608.0135	-0.0771	-33.6958	0.1489	96.2334	96.2207	1.7459	27.4372	27.4372	14.1895
5	608.0298	-0.0652	-17.8182	0.1488	96.2358	96.2231	1.7460	27.4372	27.4372	14.1895
6	608.0467	-0.0383	-49.3361	0.1487	96.2381	96.2254	1.7461	27.4372	27.4372	14.1895
7	608.0692	-0.0676	-31.5462	0.1486	96.2405	96.2278	1.7459	27.4372	27.4372	14.1896
8	608.0838	-0.0844	-24.4826	0.1485	96.2437	96.2309	1.7459	27.4372	27.4372	14.1896

Figure 4.148 Inside one of the MATLAB flight data structures

As the above images show, the flight test results were all recorded and neatly organised in MATLAB workspace as data tables with each table containing all the relevant data from the respective flight test.

4.2.7 Parameter Identification process

As described in the methods section, the runScript.m file was used to generate the following outputs for initialisation of the process.

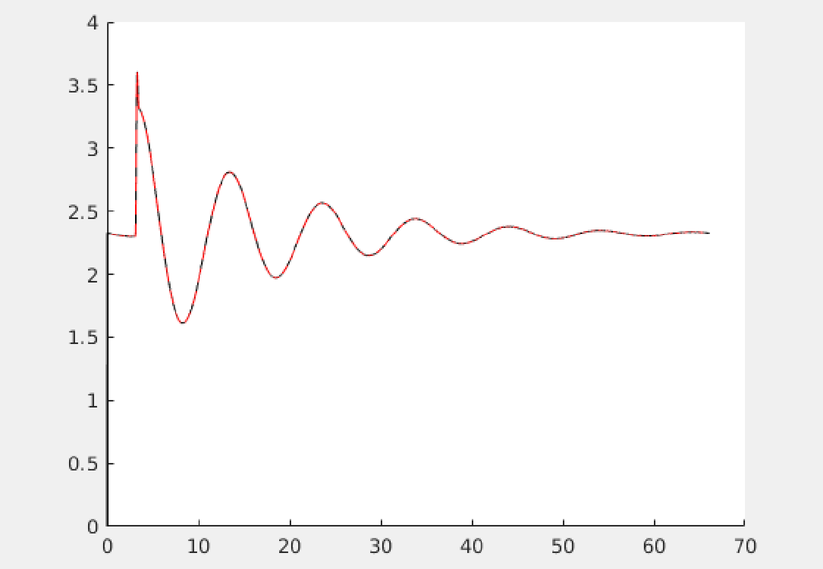


Figure 4.149 Sample of MATLAB resampling result

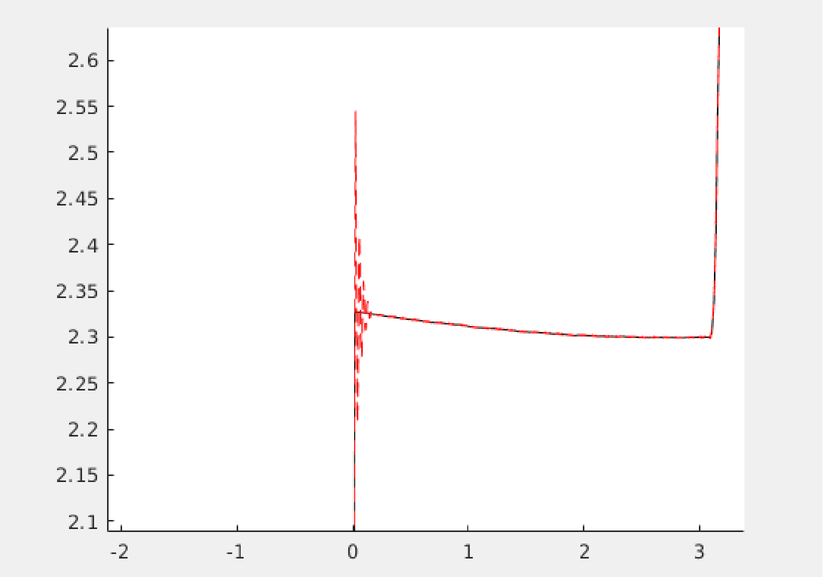


Figure 4.150 Possible FFT noise in the start of the resampled data

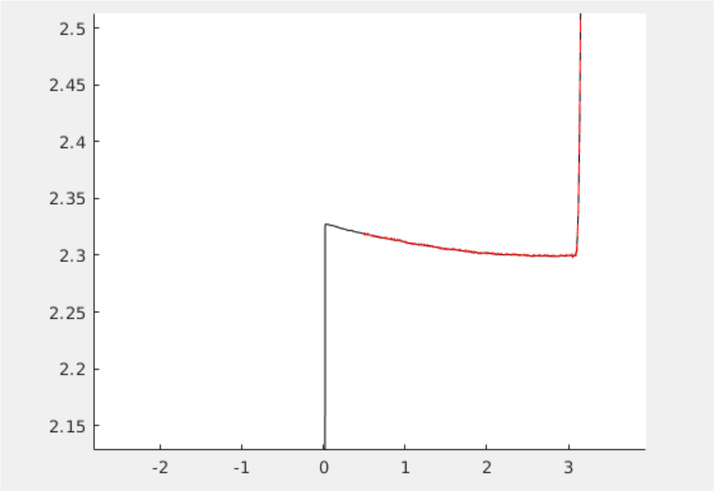
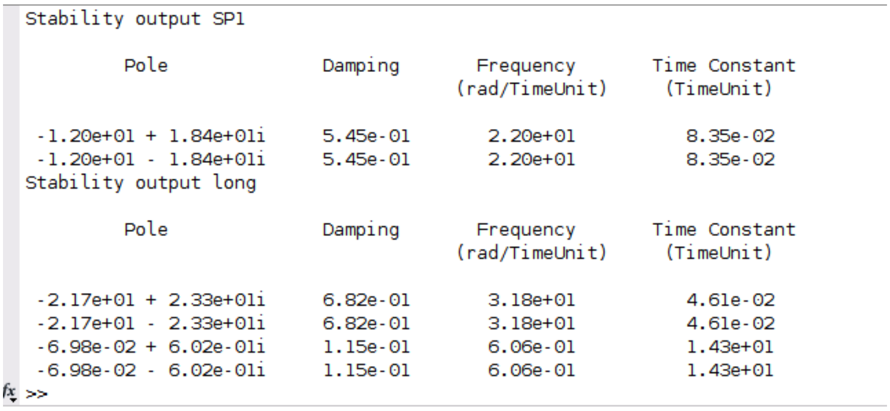


Figure 4.151 Resampled data showing good fit post processing

As it can be seen from the above images, the resampling process is able to track the original data set while faithfully interpolating it to provide the data points missing between every two time step. As discussed in the methods section, the custom scripts and functions are utilised to resample the dataset and remove unwanted noise.

4.2.7.1 Stability Output Results:

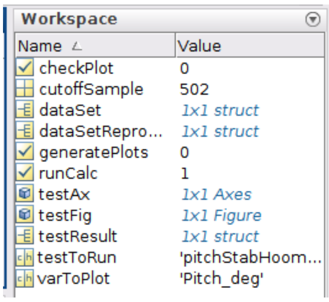
The following results were obtained from the flight test conducted for pitch stability. The specific dataset used for this one was the one with manual pitch input provided to the X-Plane model in trimmed level flight to excite the appropriate dynamic modes.



```
Stability output SP1

         Pole              Damping       Frequency       Time Constant
                                       (rad/TimeUnit)     (TimeUnit)

 -1.20e+01 + 1.84e+01i     5.45e-01       2.20e+01          8.35e-02
 -1.20e+01 - 1.84e+01i     5.45e-01       2.20e+01          8.35e-02
Stability output long

         Pole              Damping       Frequency       Time Constant
                                       (rad/TimeUnit)     (TimeUnit)

 -2.17e+01 + 2.33e+01i     6.82e-01       3.18e+01          4.61e-02
 -2.17e+01 - 2.33e+01i     6.82e-01       3.18e+01          4.61e-02
 -6.98e-02 + 6.02e-01i     1.15e-01       6.06e-01          1.43e+01
 -6.98e-02 - 6.02e-01i     1.15e-01       6.06e-01          1.43e+01
>>
```

Figure 4.152 Stability modes results

The above image shows the short and long period mode results obtained from the flight test (pitch stability). More detailed breakdown of the results were stored in the MATLAB workspace as show below.

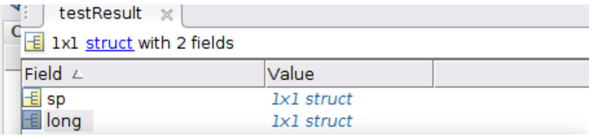


Workspace

Name	Value
☑ checkPlot	0
cutoffSample	502
dataSet	*1x1 struct*
dataSetRepro...	*1x1 struct*
☑ generatePlots	0
☑ runCalc	1
testAx	*1x1 Axes*
testFig	*1x1 Figure*
testResult	*1x1 struct*
testToRun	'pitchStabHoom...
varToPlot	'Pitch_deg'

Figure 4.153 Data and test results stored into MATLAB structures

As shown in Figure 4.153, a structure is created within MATLAB’s workspace which contains the following results based on the processed flight data.



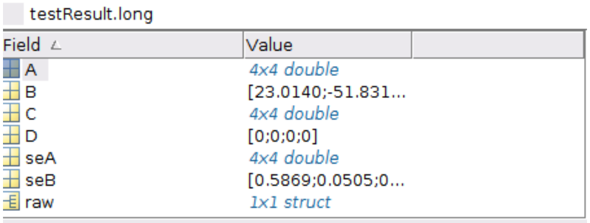
testResult

1x1 struct with 2 fields

Field	Value
sp	*1x1 struct*
long	*1x1 struct*

Figure 4.154Structures within the structures store specific test data set

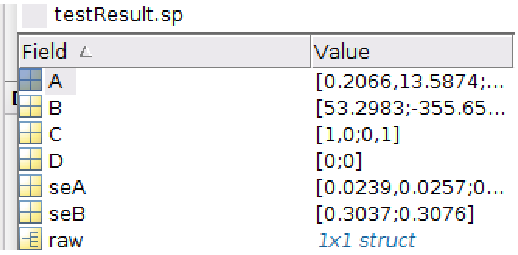
The structure “testResult” contains two structures inside it. The structure “sp” stores the information from the reduced order state space short period model while the structure “long” stores the information on the long period longitudinal state space model results.



testResult.long

Field	Value
A	*4x4 double*
B	[23.0140;-51.831...
C	*4x4 double*
D	[0;0;0;0]
seA	*4x4 double*
seB	[0.5869;0.0505;0...
raw	*1x1 struct*

Figure 4.155 Extracted matrices (A,B,C & D) and the standard errors (seA & seB) stored in MATLAB structure

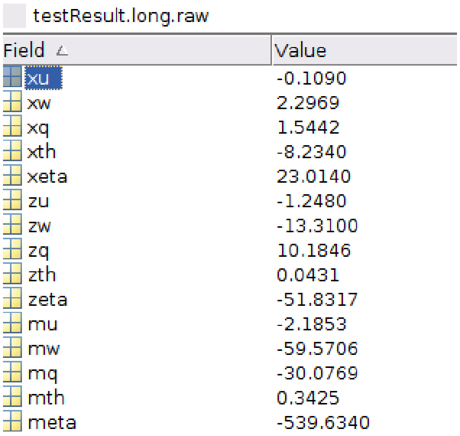


testResult.sp

Field	Value
A	[0.2066,13.5874;...
B	[53.2983;-355.65...
C	[1,0;0,1]
D	[0;0]
seA	[0.0239,0.0257;0...
seB	[0.3037;0.3076]
raw	*1x1 struct*

Figure 4.156 Extracted matrices (A,B,C & D) and the standard errors (seA & seB)

Within these structures, there is also sub-structures created (based on the custom scripts) to store the unprocessed results (Field label: raw) separately for convenience. The A-D matrices are the State Matrices and the SeA and SeB matrices contain the standard errors.



testResult.long.raw

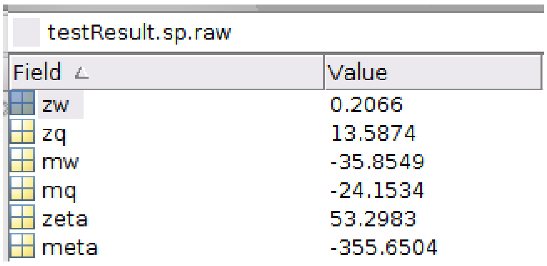
Field	Value
xu	-0.1090
xw	2.2969
xq	1.5442
xth	-8.2340
xeta	23.0140
zu	-1.2480
zw	-13.3100
zq	10.1846
zth	0.0431
zeta	-51.8317
mu	-2.1853
mw	-59.5706
mq	-30.0769
mth	0.3425
meta	-539.6340

Figure 4.157 Derivatives stored inside the “raw” structure

4.2.7.2 Result from ls_sp:

$$\begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0.2066(0.0239) & 13.5874(0.0257) \\ -35.8549(0.0242) & -24.1534(0.0260) \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} 53.2983(0.3037) \\ -355.6504(0.3076) \end{bmatrix} \eta$$

[Eq. 4.1]



testResult.sp.raw

Field	Value
zw	0.2066
zq	13.5874
mw	-35.8549
mq	-24.1534
zeta	53.2983
meta	-355.6504

Figure 4.158 Derivatives stored inside the “raw structure”

The above are the results from execution of the modified ls_sp function. They are the resulting State and Input matrices along with the standard errors in brackets (Eq. 4.1). As the standard error estimates are significantly smaller than the corresponding parameters, it can be safely concluded that the parameter identification process has been successful. It can be noted that the z_q is approximately equal to the trim airspeeds from both the X-Plane model and the XFLR5 models.

The runScript.m file had to be modified further (as shown in Chapter 7.6.5) to derive the transfer function. The following are the MATLAB outputs:

```
nQDeltaE1 =

  1.0e+03 *

      0   -0.3557   -1.8375
```

```
dQDeltaE1 =

   1.0000   23.9468  482.1819


th =
Discrete-time OE model:  y(t) = [B(z)/F(z)]u(t) + e(t)
  B(z) = -5.801 + 5.827 z^-1 - 0.02532 z^-2

  F(z) = 1 - 1.125 z^-1 + 0.127 z^-2

Sample time: 0.02 seconds

Parameterization:
   Polynomial orders:   nb=3   nf=2   nk=0
   Number of free coefficients: 5
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Model modified after estimation.
More information in model's "Report" property.
>> [dnum,dden] = th2tf(th);
   [qnum2,qden2] = d2cm(dnum,dden,0.02,'tustin')


qnum2 =

 -513.0295   -0.1551


qden2 =

   1.0000   77.5444    9.7695
```

These outputs are meant to populate the numerators and denominators of the following transfer function equation:

$$\frac{Q(s)}{DeltaE(s)} = \frac{k_q\left(s + {}^{1}/_{t_{\theta 2}}\right)}{s^2 + 2\varsigma_s\omega_s s + {\omega_s}^2} \qquad \text{[Eq. 4.2]}$$

$$\frac{Q(s)}{DeltaE(s)} = \frac{-513.0295(s - 0.1551)}{s^2 + 77.5444s + 9.7695} \qquad \text{[Eq. 4.3]}$$

Given the size and scale of the aircraft, it is difficult to reliably sense whether the numbers are good/bad. Only a proper real flight test could reveal the metric by which these numbers are to be judged for their accuracy. Because of all the limitations relating to the way the data was obtained, if these values are available through other means (e.g. XFLR5), those values are preferred.

4.2.7.3 Result from ls_long:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.1090(0.0023) & 2.2969(0.0502) & 1.5442(0.0473) & -8.2340(0.0210) \\ -1.2480(0.0001982) & -13.3100(0.0043) & 10.1846(0.0041) & 0.0431(0.0018) \\ -2.1853(0.00038698) & -59.5706(0.0084) & -30.0769(0.0080) & 0.3425(0.0035) \\ -1.5266e-16(0.00015891) & -5.2874e-15(0.0035) & 1.0000(0.0033) & -2.2204e-16(0.0015) \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 23.0140(0.5869) \\ -51.8317(0.0505) \\ -539.6340(0.0987) \\ -5.3957e4-14(0.0405) \end{bmatrix} \eta \qquad \text{[Eq. 4.4]}$$

Based on this obtained model, the following graphs were plotted to check the model's prediction against the measured data.

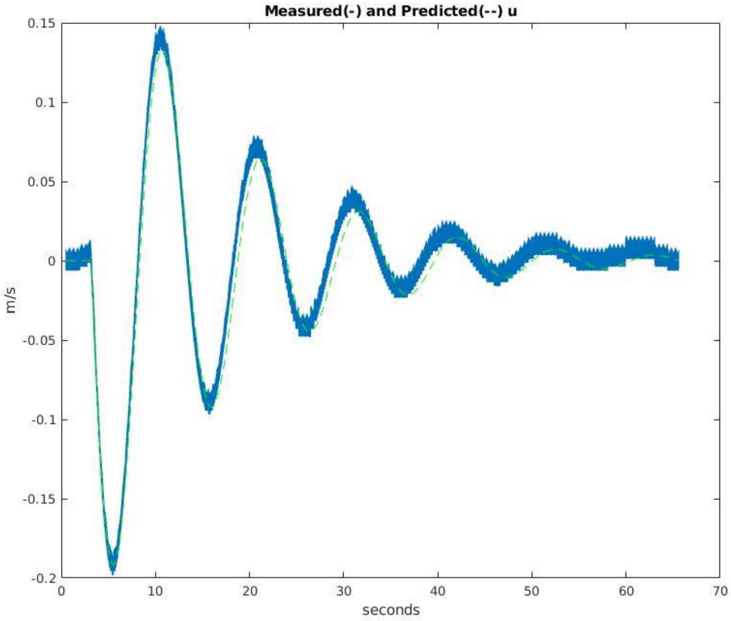


Figure 4.159 X-Plane data(measured) and model predictions for u against time

The processed flight data for the forward velocity (u) exhibited noisy artefacts from the algorithms used to resample it. This can be seen in the form of thickening of the blue curve in the above graph. However, the general pattern is still clearly evident and comparable with the predicted output values (--).

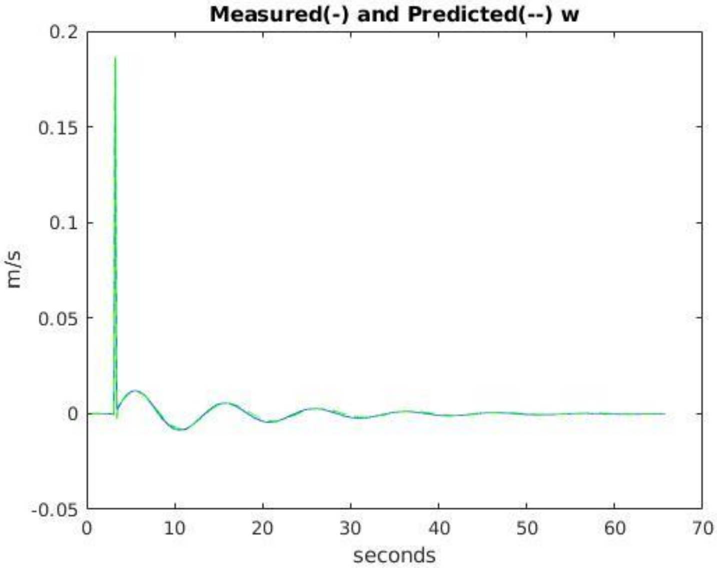


Figure 4.160 X-Plane data (measured) and model predictions for w against time

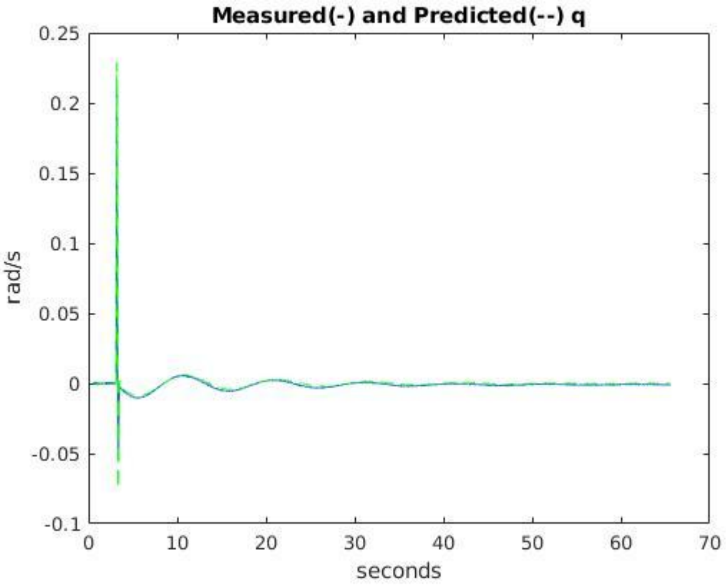


Figure 4.161 X-Plane data (measured) and model predictions for q against time

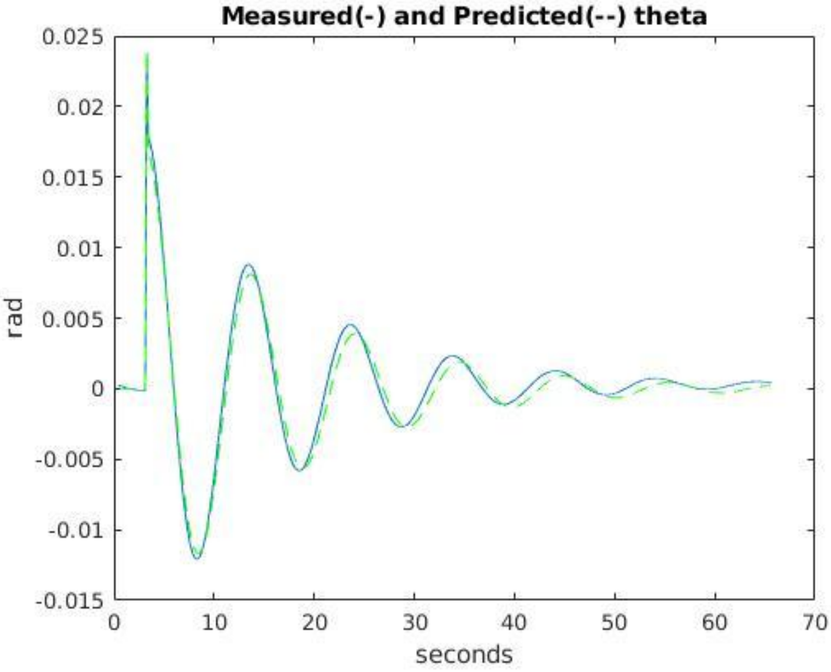


Figure 4.162 X-Plane data (measured) and model predictions for rad against time

As it can be seen from the above plots, the identified model’s prediction (--) very closely tracks the pattern of the measured data (-) from the flight test.

4.2.7.4 Problems encountered:

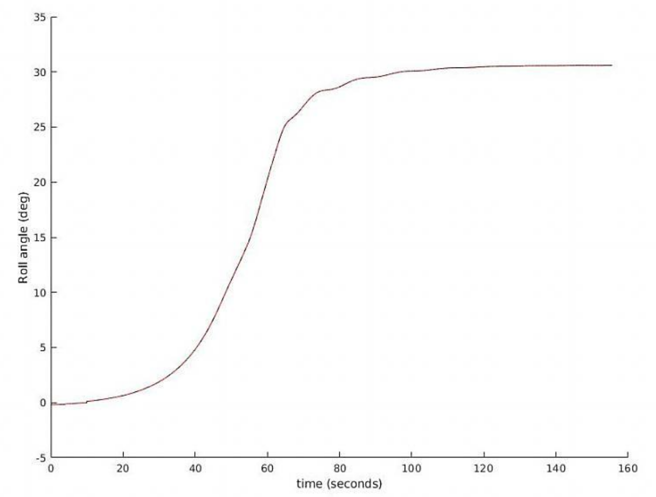


Figure 4.163 Roll instability in X-Plane model

Even though various different flight tests were conducted to excite all possible dynamic modes of the FMS 182 X-Plane model, the aileron trim issue stated earlier made the obtained flight data unusable until that issue has been resolved. As seen in the XFLR5 predictions, there are restoring roll moments that are meant to counter disturbances in that axis. While the results obtained here for the pitch stability tests have reasonable agreement with the XFLR5 test results, the roll and yaw stability test results simply cannot be compared.

As the above graph shows, instead of the aircraft restoring it’s trim, it tends to keep rolling slightly for a while until at some point when the roll rate increases and the test condition is compromised. As a result, in the MATLAB/Simulink model, the stability and control derivatives obtained from XFLR5 is heavily relied on.

4.3 Simulink Results

4.3.1 Modelling Aerofoil Stall

The sets of equations obtained from the literature that are discussed in the methods section were tested for the type of outputs. The following are the results. The MATLAB code written to test the output from these models has been documented in the Appendix (Chapter 7.3).

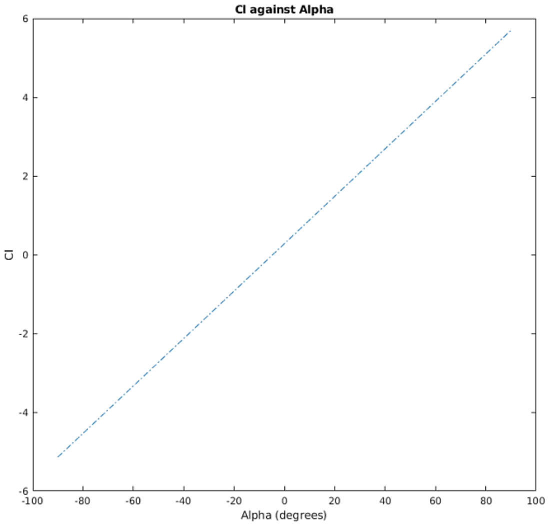


Figure 4.164 Linear lift model

As the above figure shows, the linear modelling produces unrealistic prediction for the lift coefficient for increasing/decreasing angle of attacks.

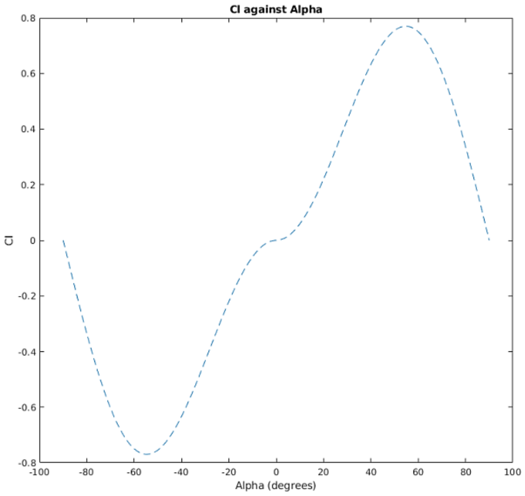


Figure 4.165 Flat plate model

As it can be seen in the above plot for the equation for the flat plate, the flat-plate equation captures the drastic changes in lift (via a change in C_L) beyond a certain range of α. This is more in line with traditional expectation of aerofoils as well as the results obtained from XFLR5 for this research.

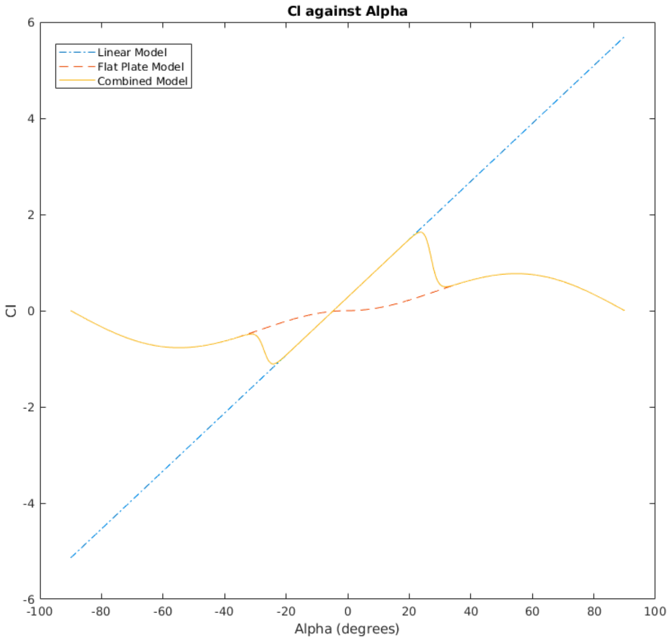


Figure 4.166 Linear (blue), Flat-Plate (red) and combined (yellow) models implemented from literature(see Chapter 3.4.1)

As the above figure illustrates, the combination of the two models via a sigmoid function enables us to derive a more realistic model for the lift generation of the aerofoil. It is possible to manipulate the transition rate (M) to generate profiles (for C_L/α) that matches a particular aerofoil more accurately.

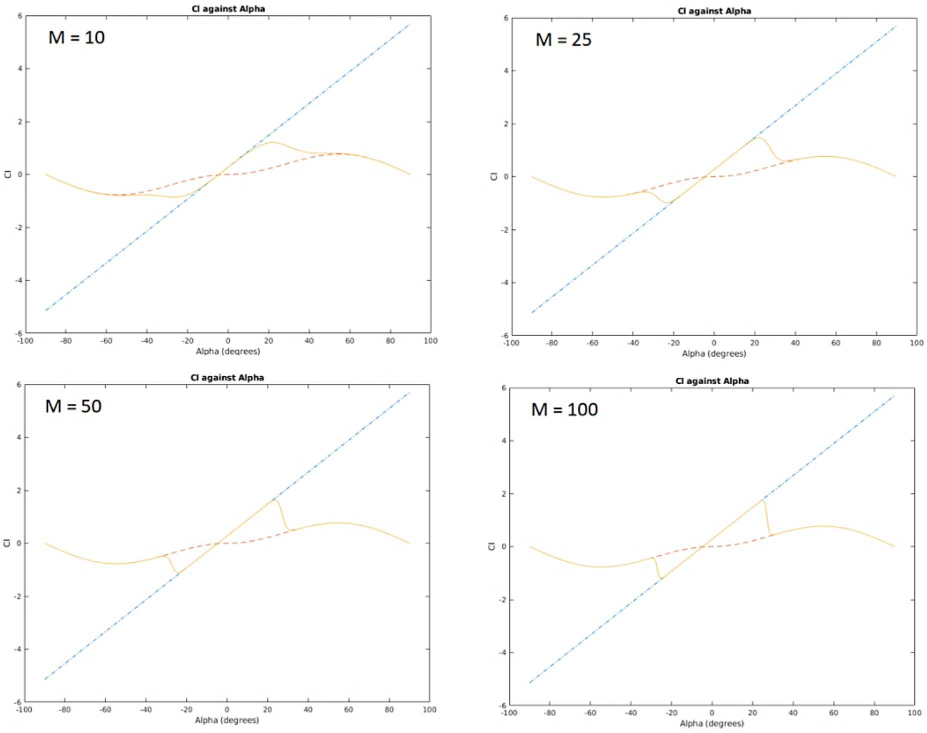


Figure 4.167 Model variations for different transition rates (M)

As it can be seen in the trend of the above image (Figure 4.167), this technique for the modelling of the lift is not only more realistic, it offers a greater deal of control over the output. However, such initial test results were not enough to accurately model the behaviour of FMS 182 target airframe. The reason being that the results represent a polar/curve that is realistic but not the realistic representation of the polar/curve for the target airframe. This prompted subsequent development of the method leading to the results in the next subsection.

4.3.2 Extension of the sigmoid function utility

As stated previously, the lift modelling process followed for this model ran into uncertainties involving the specific aerodynamic behaviour of the target airframe. As such, further development work (detailed in the methods section) was undertaken to resolve it. The following are the results from the methods and process described in the corresponding methods section.

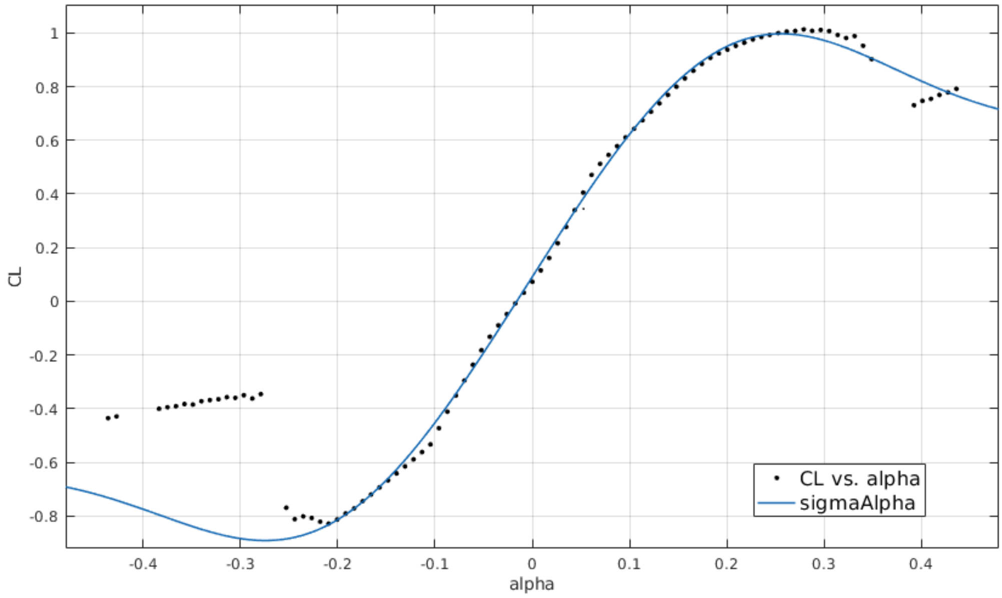
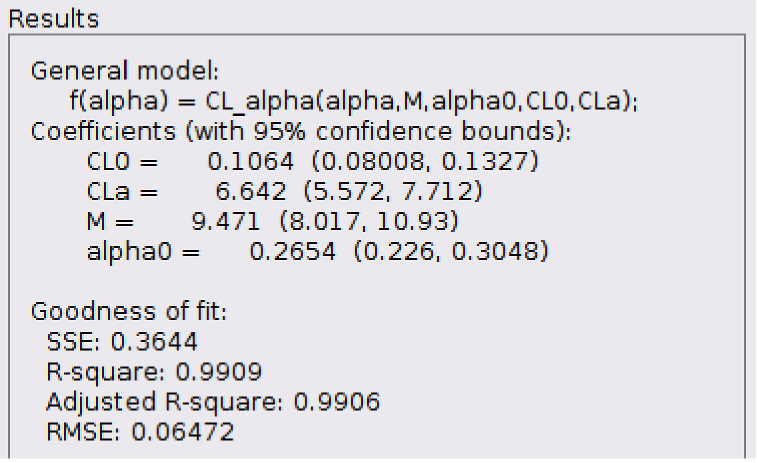


Figure 4.168 XFLR5 example data (CL vs alpha) compared to fitted curve (sigmaAlpha)

As the above graph shows, the method can successfully be used for generating the type of non-linear behaviour of the lift generated by the aircraft as was obtained by the use of the sigmoid function to blend the flat-plate approximation equation with the non-linear one. The process dropped certain data points to generate this smooth approximation but they are functionally irrelevant as the modelled aircraft is meant to maintain angles of attack that lie well within the bounds of the proper fit.



```
Results

General model:
    f(alpha) = CL_alpha(alpha,M,alpha0,CL0,CLa);
Coefficients (with 95% confidence bounds):
       CL0 =      0.1064  (0.08008, 0.1327)
       CLa =       6.642  (5.572, 7.712)
       M =       9.471  (8.017, 10.93)
       alpha0 =      0.2654  (0.226, 0.3048)

Goodness of fit:
  SSE: 0.3644
  R-square: 0.9909
  Adjusted R-square: 0.9906
  RMSE: 0.06472
```

Figure 4.169 Sigmoid curve fitting results

Another advantage of this process, as shown in Figure 4.169 is that the values for the CL0, CLa, M and alpha0 that would reproduce this model is now known. As such, they have been incorporated into the m-file for the simulation which contains values for all the constants that are needed for the initialisation of the simulation process in Simulink.

4.3.3 Modelling of Drag

Both of the models from the literature described in the methods section were tested with dummy inputs for their suitability.

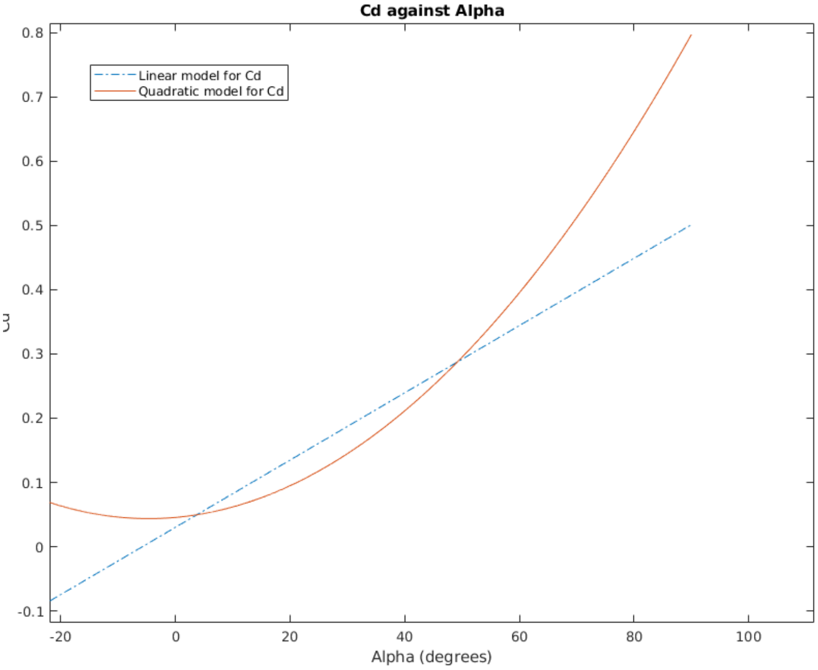


Figure 4.170 Linear (blue) and Quadratic drag (red) models from literature (Chapter 3.4.1) tested

The simplistic linear model does not produce a curve at all. On top of that inaccuracy, it falsely predicts negative values for C_D under a certain angle of attack, implying that the drag is negative within that range. The non-linear model's output behaves similar to the drag profile predicted by XFLR5 and as such this model was used.

4.3.4 Modelling of Cm:

The parameters extracted from XFLR5 and X-Plane was advantageous in the modelling of the moment coefficient (see Appendix). Their use resulted in the following curve for Cm:

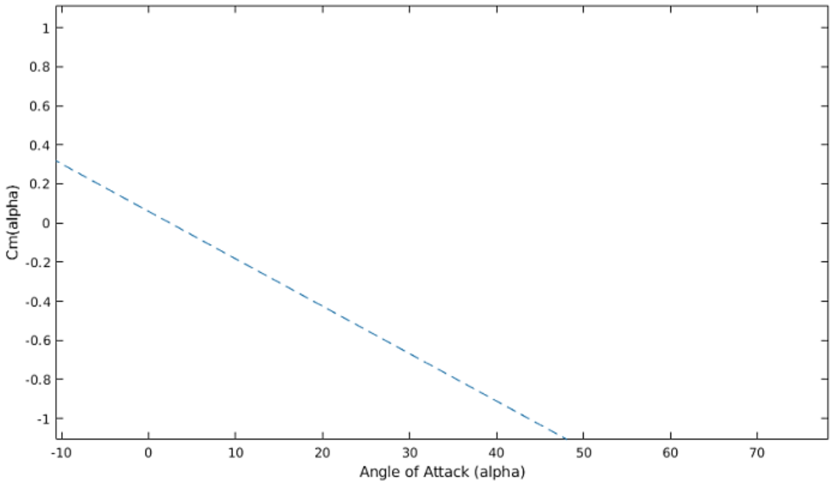
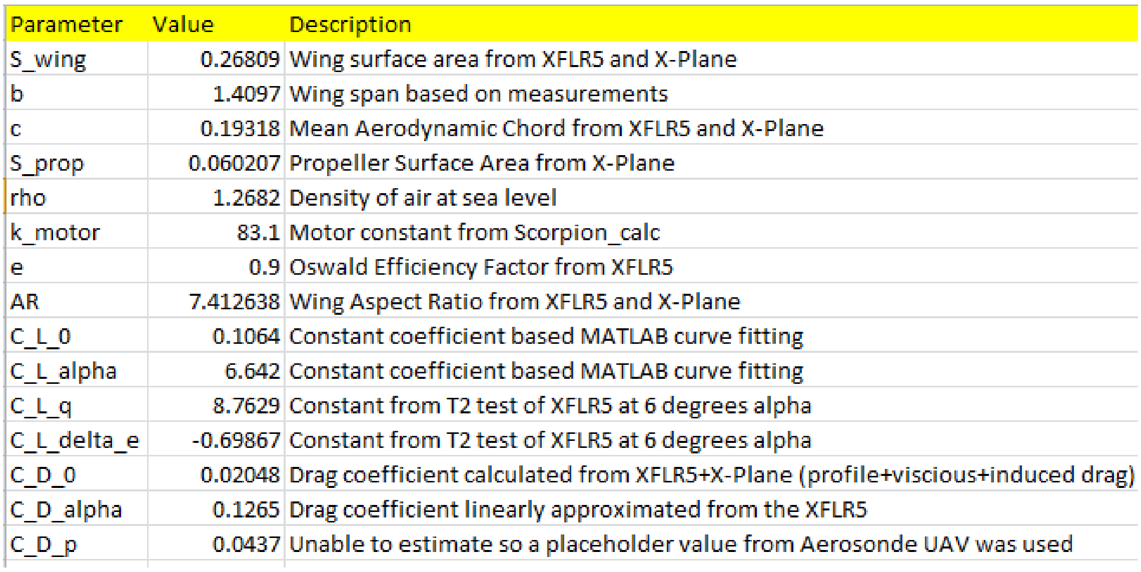


Figure 4.171 Simple linear lift model

The behaviour predicted by this linear model based on the FMS 182 datasets (from XFLR5 and X-Plane) resembles the standard pattern of restoring pitching moment for increasing alpha. Sample code depicting implementation is made available in the Appendix (Chapter 7.6.6).

4.3.5 Simulation Initialisation

The following are the list of parameters extracted from XFLR5 and X-Plane alongside physical measurements of the FMS 182 aircraft. This forms a dataset that is specific to this aircraft and allows for accurate modelling of it. However, due to the length of the actual data table, only a snippet of it is presented here.



Parameter	Value	Description
S_wing	0.26809	Wing surface area from XFLR5 and X-Plane
b	1.4097	Wing span based on measurements
c	0.19318	Mean Aerodynamic Chord from XFLR5 and X-Plane
S_prop	0.060207	Propeller Surface Area from X-Plane
rho	1.2682	Density of air at sea level
k_motor	83.1	Motor constant from Scorpion_calc
e	0.9	Oswald Efficiency Factor from XFLR5
AR	7.412638	Wing Aspect Ratio from XFLR5 and X-Plane
C_L_0	0.1064	Constant coefficient based MATLAB curve fitting
C_L_alpha	6.642	Constant coefficient based MATLAB curve fitting
C_L_q	8.7629	Constant from T2 test of XFLR5 at 6 degrees alpha
C_L_delta_e	-0.69867	Constant from T2 test of XFLR5 at 6 degrees alpha
C_D_0	0.02048	Drag coefficient calculated from XFLR5+X-Plane (profile+viscious+induced drag)
C_D_alpha	0.1265	Drag coefficient linearly approximated from the XFLR5
C_D_p	0.0437	Unable to estimate so a placeholder value from Aerosonde UAV was used

Figure 4.172 Snippet from the bigger parameters file for simulation initialisation

The actual complete dataset is much bigger and therefore presented in full form only within the Appendix (Chapter 7.6.7).

4.3.6 Visualisation Results

The following are the results of the attempts to exploit various freely available MATLAB functions to improve the quality of the visual rendering for the simulation. As it can be seen from the figure below, the simplistic rendering shown previously (in the methods section) can be replaced with a more complex one of a representative airframe visualisation. This method of geometry importing is prone to sizing/scaling problems as the STL files do not preserve the units of measurement. By converting the geometric information to the V and F matrices, that problem can be resolved by simply using linear scaling factors as shown below.

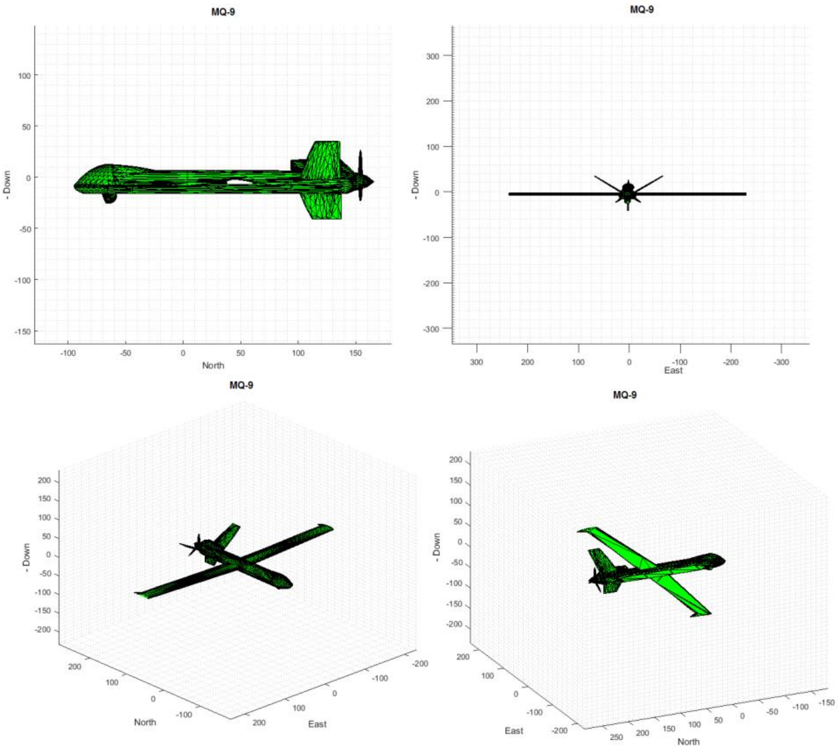


Figure 4.173 Rendering of the MQ-9 from STL file inside the visualisation module

This allows for correction of the mismatch of the scale between the 3D coordinate system used in the simulation and that of the rendered aircraft. Based on the design specifications of the MQ-9 aircraft, an appropriate scaling factor was used and the V matrix was scaled down accordingly. The F matrix is kept intact as it merely stores information on which vertices are to be connected together to form any given triangular polygon (face/shape).

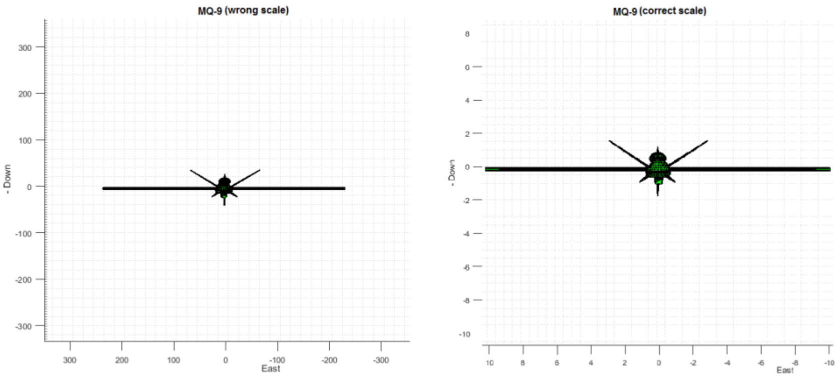


Figure 4.174 Scaling the rendered aircraft with simple matrix manipulation in MATLAB

As it can be seen from the above images, this process allows the import of and manipulation of complex aircraft shape right into the Simulink Environment.

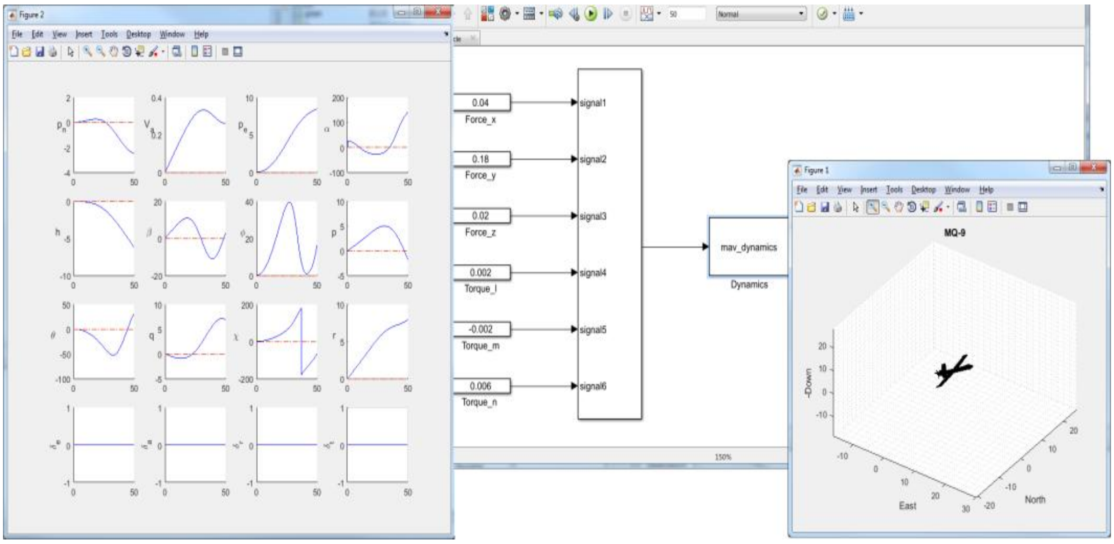
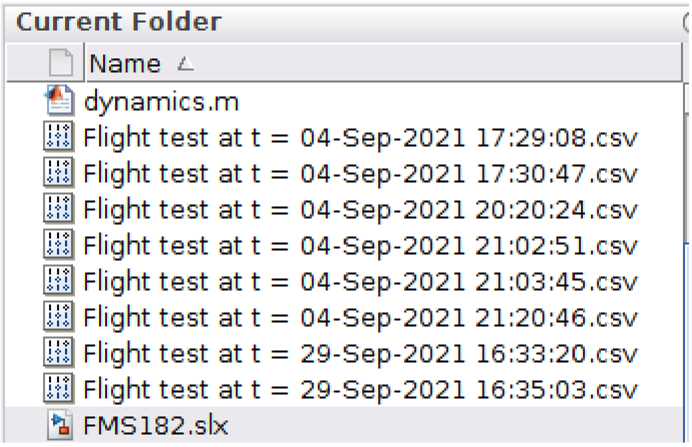


Figure 4.175 A test of the MQ9 rendering complete with dynamic inputs provided

As the above screenshot shows, the method has been tested and shown to work with the type of Simulation constructed for the FMS 182. While STL files are imported into MATLAB for various image processing applications, they are seldom used for such visualisation when dealing with the complexity of mapping a dynamic system's outputs on to the rendered objects to visualise accurate kinematics. The test put together for the MQ-9 representative airframe model shows that the method is capable of simulating the in-flight dynamic responses based on arbitrary inputs and was used to animate the dynamics of the MQ-9 geometry inside the figure window for visualisation. The dynamic responses / kinematic outputs of another previously modelled aircraft have been tested successfully for integration with the three dimensional complex representative airframe visualisation of the MQ-9 by following the method described here and earlier.

4.3.7 Data Output

The data output in the graphical form described in the methods section is shown along with the simulation output for better context. The following are the results of the extensions to the code to generate flight data recordings for later use.



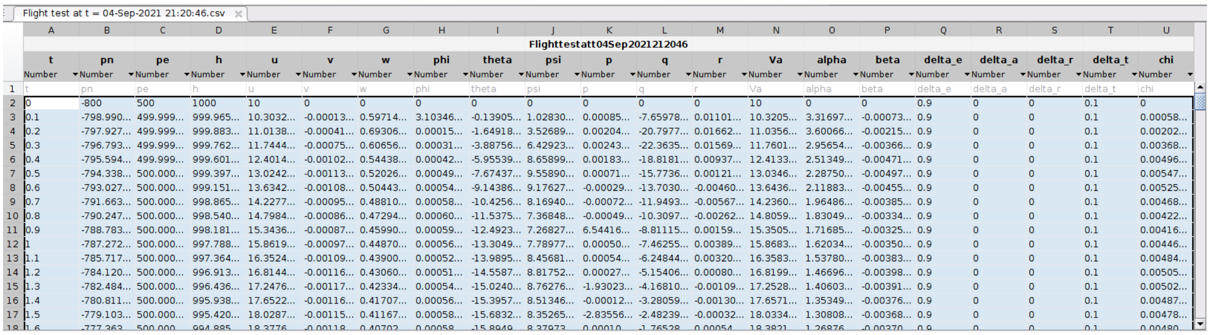
Current Folder

Name

- dynamics.m
- Flight test at t = 04-Sep-2021 17:29:08.csv
- Flight test at t = 04-Sep-2021 17:30:47.csv
- Flight test at t = 04-Sep-2021 20:20:24.csv
- Flight test at t = 04-Sep-2021 21:02:51.csv
- Flight test at t = 04-Sep-2021 21:03:45.csv
- Flight test at t = 04-Sep-2021 21:20:46.csv
- Flight test at t = 29-Sep-2021 16:33:20.csv
- Flight test at t = 29-Sep-2021 16:35:03.csv
- FMS182.slx

Figure 4.176 Creation and storage of flight test runs

As it can be seen in the above image, every simulation run results in the automatic creation of flight data for post-processing.



Flight test at t = 04-Sep-2021 21:20:46.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	Flighttestatt04Sep2021212046																				
	t	pn	pe	h	u	v	w	phi	theta	psi	p	q	r	Va	alpha	beta	delta_e	delta_a	delta_r	delta_t	chi
	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number
1	t	pn	pe	h	u	v	w	phi	theta	psi	p	q	r	Va	alpha	beta	delta_e	delta_a	delta_r	delta_t	chi
2	0	-800	500	1000	10	0	0	0	0	0	0	0	0	10	0	0	0.9	0	0	0.1	0
3	0.1	-798.990...	499.999...	999.965...	10.3032...	-0.00013...	0.59714...	3.10346...	-0.13905...	1.02830...	0.00085...	-7.65978...	0.01101...	10.3205...	3.31697...	-0.00073...	0.9	0	0	0.1	0.00058...
4	0.2	-797.927...	499.999...	999.883...	11.0138...	-0.00041...	0.69306...	0.00015...	-1.64918...	3.52689...	0.00204...	-20.7977...	0.01662...	11.0356...	3.60066...	-0.00215...	0.9	0	0	0.1	0.00202...
5	0.3	-796.793...	499.999...	999.762...	11.7444...	-0.00075...	0.60656...	0.00031...	-3.88756...	6.42923...	0.00243...	-22.3635...	0.01569...	11.7601...	2.95654...	-0.00366...	0.9	0	0	0.1	0.00368...
6	0.4	-795.594...	499.999...	999.601...	12.4014...	-0.00102...	0.54438...	0.00042...	-5.95539...	8.65899...	0.00183...	-18.8181...	0.00937...	12.4133...	2.51349...	-0.00471...	0.9	0	0	0.1	0.00496...
7	0.5	-794.338...	500.000...	999.397...	13.0242...	-0.00113...	0.52026...	0.00049...	-7.67437...	9.55890...	0.00071...	-15.7736...	0.00121...	13.0346...	2.28750...	-0.00497...	0.9	0	0	0.1	0.00547...
8	0.6	-793.027...	500.000...	999.151...	13.6342...	-0.00108...	0.50443...	0.00054...	-9.14386...	9.17627...	-0.00029...	-13.7030...	-0.00460...	13.6436...	2.11883...	-0.00455...	0.9	0	0	0.1	0.00525...
9	0.7	-791.663...	500.000...	998.865...	14.2277...	-0.00095...	0.48810...	0.00058...	-10.4256...	8.16940...	-0.00072...	-11.9493...	-0.00567...	14.2360...	1.96486...	-0.00385...	0.9	0	0	0.1	0.00468...
10	0.8	-790.247...	500.000...	998.540...	14.7984...	-0.00086...	0.47294...	0.00060...	-11.5375...	7.36848...	-0.00049...	-10.3097...	-0.00262...	14.8059...	1.83049...	-0.00334...	0.9	0	0	0.1	0.00422...
11	0.9	-788.783...	500.000...	998.181...	15.3436...	-0.00087...	0.45990...	0.00059...	-12.4923...	7.26827...	6.54416...	-8.81115...	0.00159...	15.3505...	1.71685...	-0.00325...	0.9	0	0	0.1	0.00416...
12	1	-787.272...	500.000...	997.788...	15.8619...	-0.00097...	0.44870...	0.00056...	-13.3049...	7.78977...	0.00050...	-7.46255...	0.00389...	15.8683...	1.62034...	-0.00350...	0.9	0	0	0.1	0.00446...
13	1.1	-785.717...	500.000...	997.364...	16.3524...	-0.00109...	0.43900...	0.00052...	-13.9895...	8.45681...	0.00054...	-6.24844...	0.00320...	16.3583...	1.53780...	-0.00383...	0.9	0	0	0.1	0.00484...
14	1.2	-784.120...	500.000...	996.913...	16.8144...	-0.00116...	0.43060...	0.00051...	-14.5587...	8.81752...	0.00027...	-5.15406...	0.00080...	16.8199...	1.46696...	-0.00398...	0.9	0	0	0.1	0.00505...
15	1.3	-782.484...	500.000...	996.436...	17.2476...	-0.00117...	0.42334...	0.00054...	-15.0240...	8.76276...	-1.93023...	-4.16810...	-0.00109...	17.2528...	1.40603...	-0.00391...	0.9	0	0	0.1	0.00502...
16	1.4	-780.811...	500.000...	995.938...	17.6522...	-0.00116...	0.41707...	0.00056...	-15.3957...	8.51346...	-0.00012...	-3.28059...	-0.00130...	17.6571...	1.35349...	-0.00376...	0.9	0	0	0.1	0.00487...
17	1.5	-779.103...	500.000...	995.420...	18.0287...	-0.00115...	0.41167...	0.00058...	-15.6832...	8.35265...	-2.83556...	-2.48239...	-0.00032...	18.0334...	1.30808...	-0.00368...	0.9	0	0	0.1	0.00478...
18	1.6	-777.363	500.000	994.885	18.3776	-0.00118	0.40702	0.00058	-15.8949	8.37973	0.00010	-1.76528	0.00054	18.3821	1.26876	-0.00370	0.9	0	0	0.1	0.00480

Figure 4.177 Properly labelled and organised flight test data in CSV format

The end result is that CSV files containing data from every time step of the simulation is automatically stored in the neatly organised manner as shown in the above image. All the values are in the right order and the headers are automatically populated using the variable names for convenience.

4.3.8 Complete Simulink Model

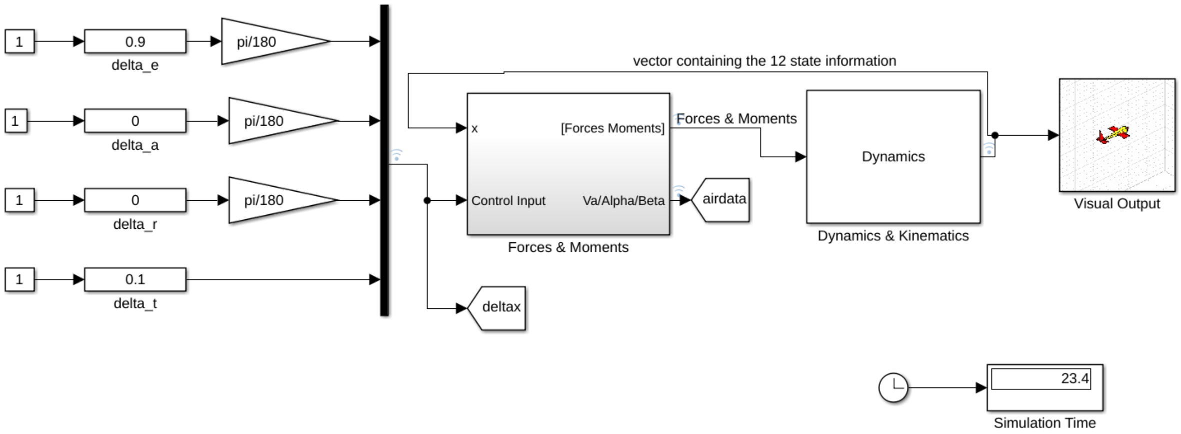


Figure 4.178 Screenshot from Simulink of the blocks put together for the simulation of the FMS 182

The individual components of this overall model are explained in the methods section. The templates mentioned earlier were utilised and modified and extended in functionality as explained in the methods section. In the above image, from the left are the control blocks, Forces & Moments block, Dynamics & Kinematics block and the Visual Output block in order. The control blocks (present from the templates) are just default Simulink slider gains with constants for unit conversion. The black bar is the multiplexer used to collate the individual signals (elevator/rudder/aileron/throttle inputs) and pass them to the Forces & Moments block and a Goto block. The Goto block passes the signals it to the Visual Output block without requiring the physical connections being shown for convenience. The outputs from the Forces & Moments block the inputs to the Dynamics & Kinematics block (the Forces and Moments) and the Goto block labelled “airdata”, which again is passed to the Visual Output block. The output of the Dynamics & Kinematics block is the input to both the Visual Output and the Forces & Moments block. The simulation is configured to run on fixed-step of 0.1 seconds.

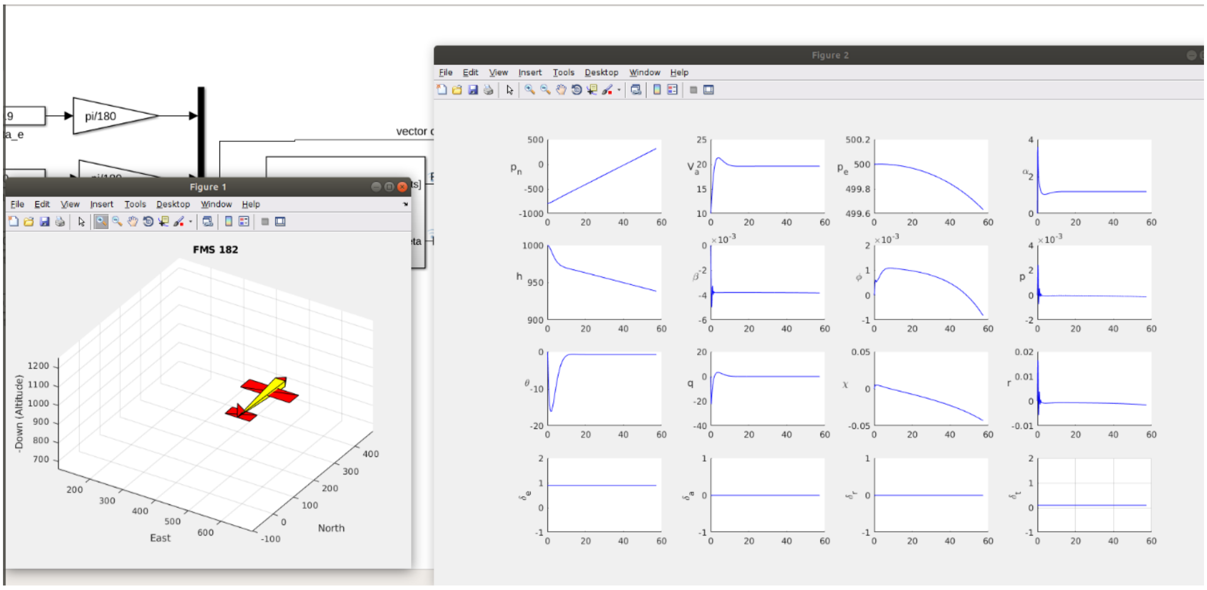


Figure 4.179 Visualisation and plotted variables based on the methods stated earlier (Chapter 3.4.6)

As shown in Chapter 4.3.5 and the above image, the full setup with all modifications and customisation works and outputs the 12 state information. The following are some tests results done to check the responses of this simulation of the FMS 182. The following are visual outputs from a test conducted with a slight elevator input and throttle set to 10%. The simulation was left running for a while and the responses recorded.

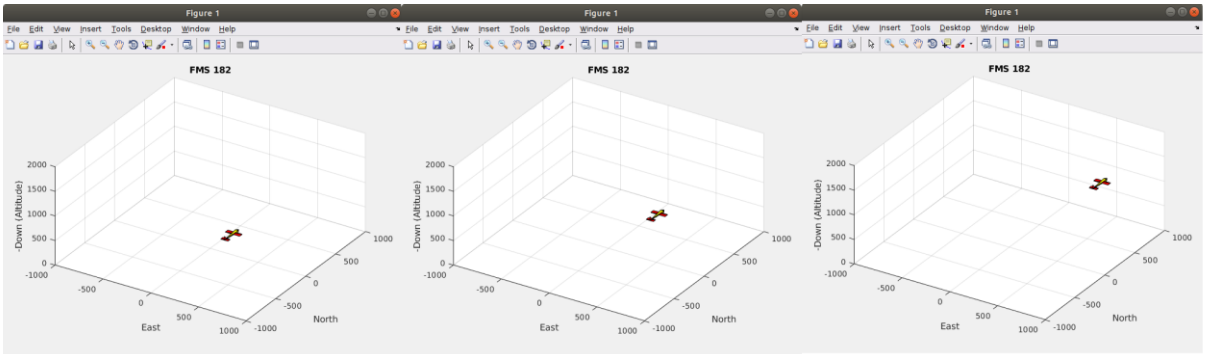


Figure 4.180 Visualisation of the aircraft following the approaches detailed in Chapter 3.4.6

From the visual rendering it can clearly be seen that the aircraft is in motion and moving due North.

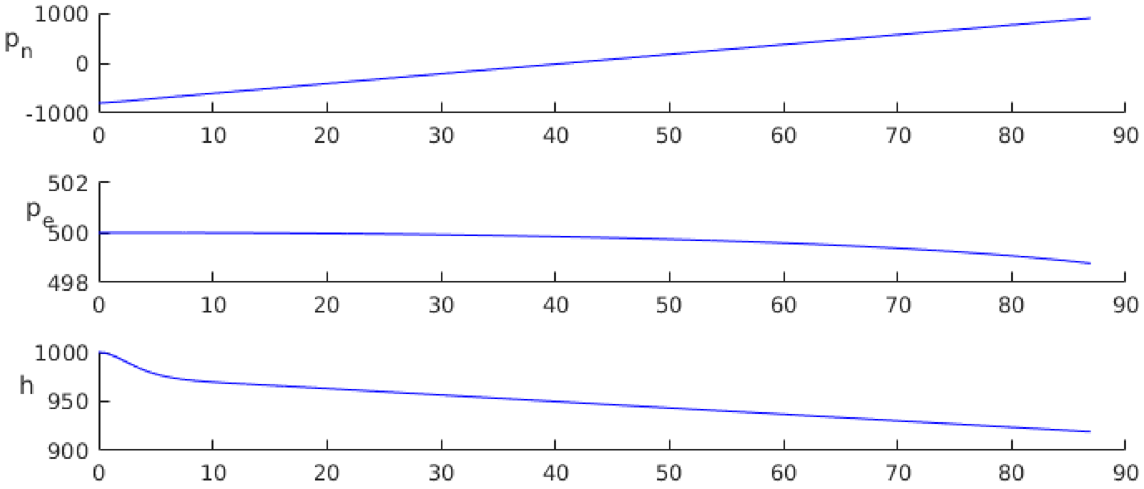


Figure 4.181 Sample outputs from the test for North, East and Down positions

As expected, the graphs for the north, east and altitude (down) show the changes in the positions. It is interesting to note that despite no wind present and in the absence of any aileron or rudder input, the aircraft is drifting gradually east. The graph for altitude reflects what’s expected for the elevator input.

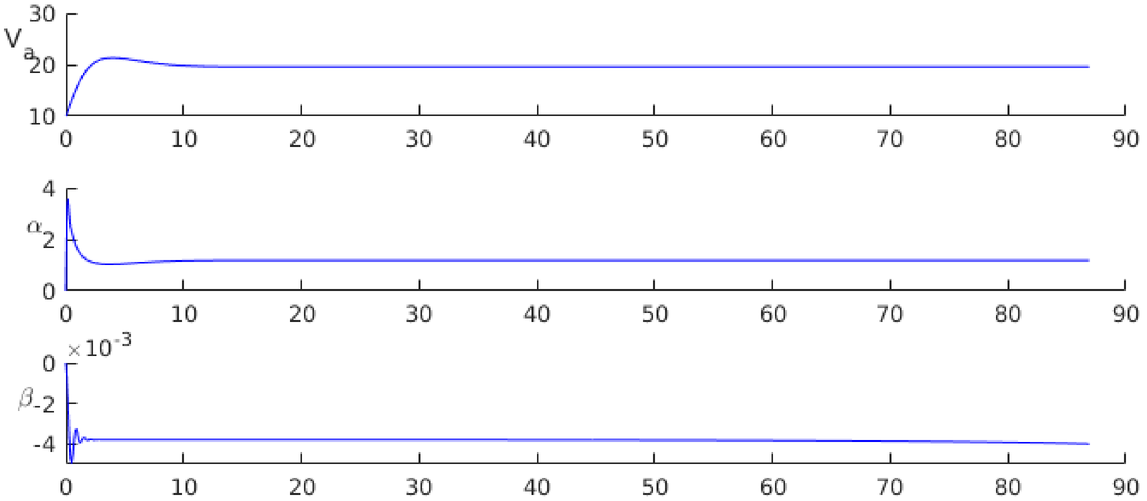


Figure 4.182 Sample output from the test showing airspeed, angle of attack and side slip angles

The aircraft seems to have stabilised in forward velocity at around 19 m/s. In previous rudimentary tests with the linear lift modelling, this wasn’t the case. The aircraft’s airspeed gets progressively over-estimated and leads to highly unusual large numbers which often end up crashing the simulation. The angle of attack is shown to have dropped from the initial condition to the stable run. There is a constant side slip (beta angle) present throughout the test envelope.

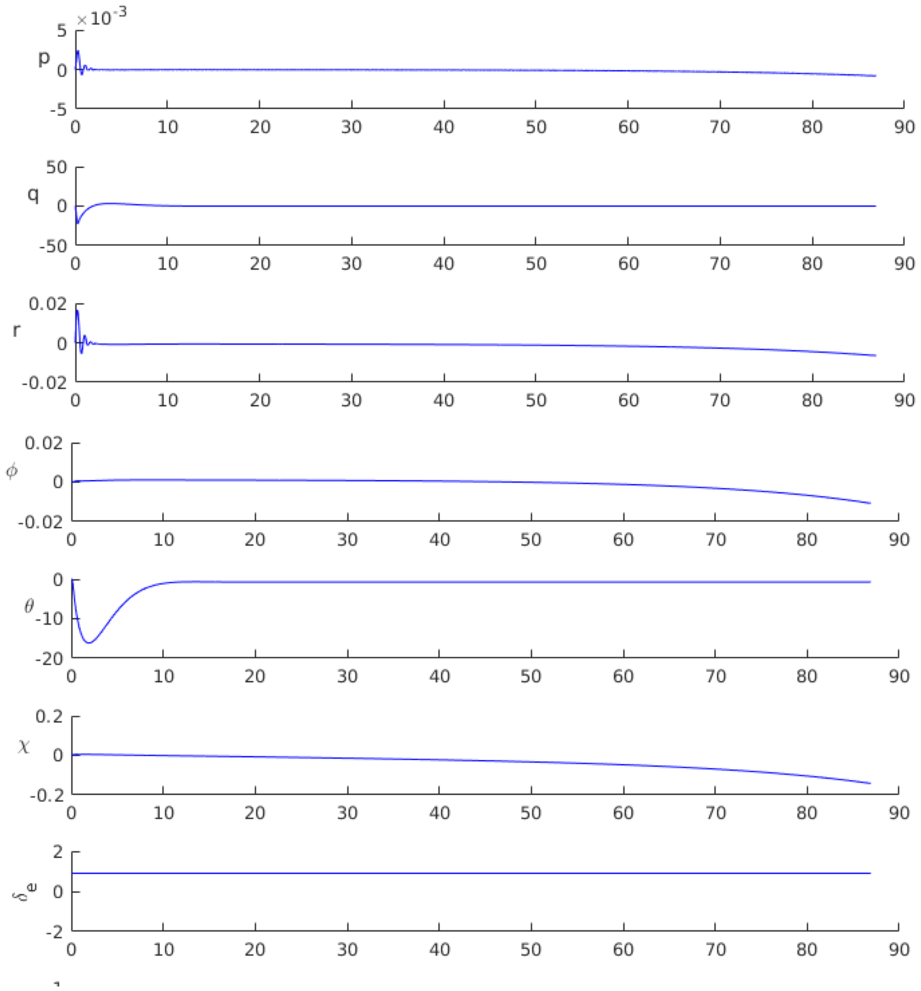


Figure 4.183 Sample test output showing roll(p), pitch (q), yaw (r) rates along with the roll, pitch and yaw angles and delta_e (elevator input)

It can be seen that from the slight elevator input, even though the roll and pitch and their rates are initially disturbed, they eventually settle. The yaw and yaw-rate however steers away from the starting point and keeps drifting. Similar drifting problems have been encountered in the X-Plane flight tests where there was a roll instability. There can be cross-coupling between roll and yaw due to yaw motion pushing one wing into the airflow while making the other move away from the airflow. This causes asymmetric lift production of the left and the right wings leading to a rolling moment.

The MATLAB/Simulink simulation of the FMS 182 however can successfully simulate the dynamic behaviour of the fixed-wing plane and provide a foundation for further development. During the test runs of this simulation setup (based on the parameters measured in real life or extracted from XFLR5/X-Plane), various shortcomings were identified which will be discussed in the end of this thesis. But this should not detract from the high utility of this platform and future development potentials now that this test-bench has already been created.

4.4 Flight Test

While various simulated flight tests (Chapter 4.1.3-4.1.9 & 4.2.5) were carried out within the simulation space of XFLR5, X-Plane and MATLAB/Simulink (Chapters 4.1-4.3), the work relating to the real-world flight tests of the physical platform (FMS 182) remains incomplete due to the COVID-19 pandemic related restrictions. This is addressed in more detail in Chapter 6.2.

5 Discussion

The following subsections are discussions based on the findings in the previous chapter. They are organised based on topic.

5.1 Measurement Issues

As mentioned in Chapter 3, a fixed-wing platform (FMS 182) had to be selected for the proposed research. Even though the purchase of that aircraft helped in reducing workload of the alternative process of designing from scratch, it resulted in challenges of measurement.

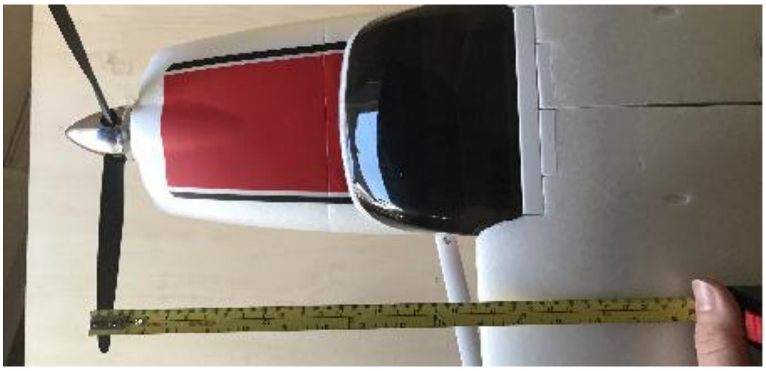


Figure 5.1 Measuring with common hand tools

A digital scale was used for weighing components. Ordinary rules, measuring tape and callipers were used for measuring length. Angles were measured using a protractor. They introduced sources of error of their own. The following are a summary of measurement challenges encountered:

- Rounding off error due to measurement accuracy of the tool.
- Parallax error in measurement of distances/angles.
- Difficulties in pinpointing the C.G. location of the modules that required to be balanced on a point. This can be potentially mitigated by suspending the aircraft from two points along the fuselage and hanging a plumb line from the same hanger location which points to the cg position.

The measured data obtained from these methods were used for the subsequent XFLR5, X-Plane and MATLAB/Simulink work. As a result, the data obtained from those simulations and modelling work suffer from modelling errors originating at this measurement phase. For example, an incorrectly measured C.G. location of a component will shift the overall inertial tensor estimation by XFLR5 and thereby introduce some degree of error into the stability/control results; measurement noises such as these are unavoidable due to every tool having its finite precision/accuracy. They can however be improved using more sophisticated measurement systems. The advantage of the method used is that it relies on ordinary hand tools which make the process more widely adoptable.

5.2 XFLR5

This freely available and open-source tool proved highly effective in a number of ways. It showed the greatest amount of utility in the following works, with reference to the FMS 182:

- Obtaining aerofoil analysis data.
- Construction of model for aerodynamic performance tests.
- Construction of model for stability and control tests.
- Creation of custom aerofoil files for the X-Plane simulation model.
- Providing inertial, stability and control parameters for both X-Plane and Simulink models.

While XFLR5 proved to have great utility, certain features and bugs in this software have created problems in the research process. Some of these problems are due to unclear documentation or mismatch of terminology. For example, XFLR5 labels all control surfaces as “flaps”. The user interface for the control tests is not intuitive and would benefit from being made more intuitive. Analysis results are not always saved in the project files unless certain boxes are ticked voluntarily by the user. This resulted in the need to repeat a lot of the analysis.

The general approach for the XFLR5 work was to first create all the aerofoils for the FMS 182. These aerofoils were subjected to appropriate low Reynolds number analysis. The dataset created from that were used for the performance analysis. For the performance analysis, a range of model variants were needed to be created as XFLR5 doesn't allow one model with multiple configurations. Every one of these models needed to be assigned aerofoils specific to the variant (e.g. positive aileron deflection). Inertial modelling based on measured data also needed to be prepared for the Stability and Control tests. Overall, a large number of aerofoils and models needed to be created and a long list of tests were performed on them. This resulted in a large number of repeated trials and an equally large dataset of analysis and results. The software has no scripting feature and as such virtually every part of this process had to be manually repeated and the dataset manually gathered and recorded. This led to the writing of custom MATLAB functions that are capable of handling data acquisition processing and storage from XFLR5.

Once all the procedures mentioned in Chapter 3 were followed and all the models were generated, the software (XFLR5) proved its usefulness as detailed in Chapter 4. As discussed within the results section, all the performance analysis visualisation and polars were consistent with general expectations. The stability and control analysis results also showed reasonable similarities with expected dynamic behaviour of the aircraft. But these results have to be interpreted in the light of the limitation of the program. For example, due to the fuselage not being modelled, the overall performance estimations can be considered to be over-estimated. The stability and control tests however are not impacted as much by this as the dynamic tests are dependent on the inertial modelling based on point masses (see Chapter 3).

Considering that the FMS 182 is expected to be an autonomous platform carrying various payloads, the tests involving changing weight and C.G. location produced a good collection of useful data for any future work on this platform with currently unknown/undecided payloads. Once the methods shown are followed and these datasets are generated, they can be used to inform model changes in both X-Plane and Simulink. The results obtained from these mass and C.G. variations are useful for any future plans

involving sensors, controllers, etc. which will inevitably disturb these parameters and result in altered dynamic responses.
Many of the results obtained from XFLR5 proved useful for improving the modelling accuracy of both the X-Plane and MATLAB/Simulink modelling and simulation.

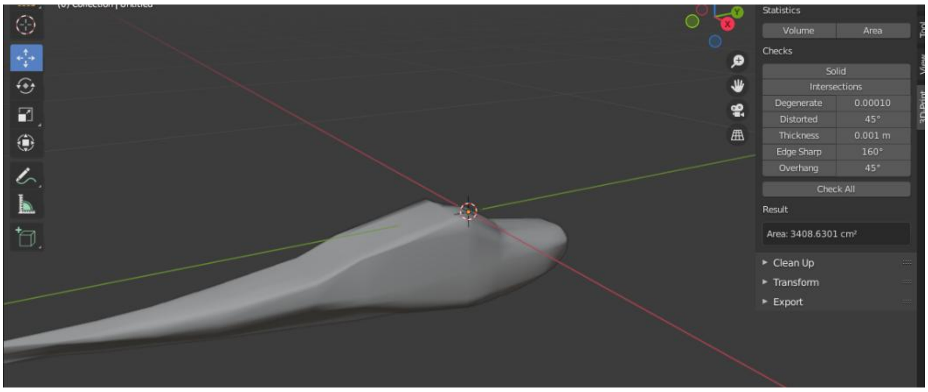
5.3 X-Plane

Despite the limitations, lack of appropriate documentation and uncertainties discussed earlier in the thesis about the application of X-Plane for modelling an aircraft at such a small scale, it was possible to successfully create an X-Plane model of the FMS 182. This was achieved by taking a mixed approach where the XFLR5 data and custom MATLAB functions were used to make the model more representative of the FMS 182.

There were a number of challenges that need to be highlighted. Plane Maker was used to create the model. It lacks any feature to import 3D models of entire designs and the user needs to manually create the geometry following the methods described earlier in the thesis (Chapter 3.3). The problem isn’t limited to just the difficulty in using that tool to generate custom geometries as shown in the documentation of the construction method. It doesn’t allow universally changing all units to conform to one single standard of unit of measurement. Furthermore, the user interface itself was designed with full-scale aircraft in mind as various parameters had to be rounded. Despite these drawbacks, it was possible to successfully model the FMS 182 and confirm its viability for modelling at this scale.

The default collection of aerofoil files (.afl format) lacked data in the low Reynolds number regime (10^3-10^5) that an aircraft the size of the FMS 182 is meant to operate in. This required the development of a process to successfully generate these aerofoil files for the FMS 182 (or any other UAV). Already having the aerofoil data from XFLR5, the biggest challenge was the lack of documentation on the file format and structure of the aerofoil files. The method shown in this thesis to do this and the subsequent results were not as straight forward as this documentation might suggest. In practice, it required an extensive series of trial and error to understand all the unlabelled numbers in the .afl file. The data from XFLR5 needed to be imported on to MATLAB, followed by further post-processing to interpolate a sample .afl file. Once that was ready, the model had to be opened in Plane Maker and this test aerofoil assigned to the right element. If it accepted it, then the next test was to load the aircraft in X-Plane and check. As such, repeated modifications were carried out until the aerofoil file was finally accepted by X-Plane simulator as valid. This resulted in a satisfactory solution to the problem of the lack of aerofoils for aircraft like the FMS 182, as discussed in the literature review section.

There were additional challenges faced in the drag modelling of the X-Plane aircraft model. X-Plane expects the zero-lift drag coefficient to be supplied by the user for components such as the fuselage, propeller spinner and wheel fairings. Although classical sources of estimations for these values, such as Hoerner et al [1][2][3][4], were consulted to improve simulation accuracy, it proved unsuccessful. The primary reason being that most published data refer to Reynolds number regimes that are much higher than the operational range of the FMS 182. As such, to improve the modelling accuracy, these parameters needed to be calculated based on the recommended formulations in the previously mentioned sources. The formulas however required good estimates of wetted surface areas that are not easily measured.



Statistics

Checks

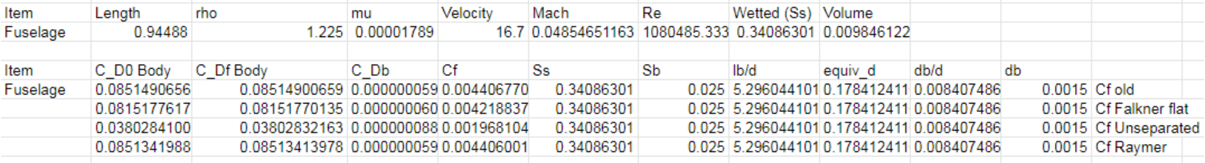
Degenerate	0.00010
Distorted	45°
Thickness	0.001 m
Edge Sharp	160°
Overhang	45°

Result

Area: 3408.6301 cm²

Figure 5.2 FMS182 fuselage imported to blender

As a result, the model was exported as OBJ format and split apart in the free and open-source 3D modelling software Blender [4] to obtain these estimates as shown in Figure 5.2.



Item	Length	rho	mu	Velocity	Mach	Re	Wetted (Ss)	Volume			
Fuselage	0.94488	1.225	0.00001789	16.7	0.04854651163	1080485.333	0.34086301	0.009846122			
Item	C_D0 Body	C_Df Body	C_Db	Cf	Ss	Sb	lb/d	equiv_d	db/d	db	
Fuselage	0.0851490656	0.08514900659	0.000000059	0.004406770	0.34086301	0.025	5.296044101	0.178412411	0.008407486	0.0015	Cf old
	0.0815177617	0.08151770135	0.000000060	0.004218837	0.34086301	0.025	5.296044101	0.178412411	0.008407486	0.0015	Cf Falkner flat
	0.0380284100	0.03802832163	0.000000088	0.001968104	0.34086301	0.025	5.296044101	0.178412411	0.008407486	0.0015	Cf Unseparated
	0.0851341988	0.08513413978	0.000000059	0.004406001	0.34086301	0.025	5.296044101	0.178412411	0.008407486	0.0015	Cf Raymer

Figure 5.3 Analytical estimation of fuselage zero-lift drag coefficient by various methods

Various approximation formulas were used with the same variables (per object) to estimate the zero-lift drag coefficient. They all calculate the skin friction drag coefficient and add it to the base-drag of the body to arrive at the final drag coefficient. The final values used for improving the drag modelling by X-Plane were 0.081, 0.34, 0.103 for the fuselage, propeller spinner and wheel fairings respectively. This is the best approximate arrived at and the X-Plane simulation results are based on the use of these figures. In the case of the spinner, the simplifying assumption of no axial rotation was used because no method of analytically predicting the combined effect of the spin and forward motion was found.

Although the overall modelling process was largely successful, there were difficulties experienced (refer to Chapter 4) upon flight testing. Particularly the aircraft having a tendency to roll and as such fail to stay in the trimmed condition. X-Plane’s inability to stay in roll trim hold for certain models was a known issue that has no fixes that could be applied. Before the results were obtained, a lot of modifications to the simulation setup and the model itself were carried out to isolate the source of the lack of roll trim hold. A range of tests including a very unrealistic high dihedral angle wing setting to improve stability was carried out. Yet the roll trim destabilisation still remained. Therefore the less than desirable roll trim function remain unresolved. It was still possible to extract some flight data for testing out some parameter estimation techniques as detailed in the Chapter 4.

The extraction of flight data for post-processing and analysis, from X-Plane, revealed some concerns worthy of note. For example, it was hard to obtain all the necessary parameters from the default flight data output menu. As such, alternative means had to be explored. Originally, the MATLAB code from NASA mentioned in the literature review was used but it proved cumbersome for the limited hardware capacity to simultaneously run both platforms (X-Plane & MATLAB). This required the use a freely available JavaScript [6] which was modified as shown in the relevant section of the thesis (Chapter 3.3). The imported data however revealed a low sampling frequency. The lack of high enough sampling frequency for the estimation algorithms and inconsistent frame rates were observed. A simple Parameter Estimation MATLAB code

[15] was modified to solve these inconsistencies of sampling rate and frequency to operate successfully on the X-Plane flight data and produce the results reported.

Despite the challenges, a fully functional X-Plane model with good fidelity (accurate design, custom aerofoils, etc.) was produced. It demonstrates the usefulness of the simulation platform as an independent modelling and simulation platform for small-scale fixed-wing UAVs such as the FMS 182.

5.4 Simulink

As detailed in the Results section, a 12-state six-degree-of-freedom Simulink model of the FMS 182 has been successfully constructed by following the methods detailed previously in the thesis. Utilisation of pre-existing MATLAB template files[14] for the various components of this simulation has not only helped construct this simulation, it's transparency and extendibility allowed for a range of modifications in MATLAB code to further develop the simulation platform. The experimentation result based on the 3D visualisation module proved useful.

	MATLAB/ Simulink	X-Plane	FlightGear
RAM	8 GB	16-24 GB RAM or more	6-8 GB
Processor	Any Intel or AMD x86-64 processor with four logical cores and AVX2 instruction set support	Intel Core i5 6600K at 3.5 ghz or faster	A quad core processor with ~ 2 GHz each, 64 bit architecture
Graphics	No specific graphics card is required.	A DirectX 12-capable video card from NVIDIA, AMD or Intel with at least 4 GB VRAM (GeForce GTX 1070 or better or similar from AMD)	1024-2048MB of dedicated DDR3+ (DDR5 preferred) VRAM (i.e. 512 Mb VRAM minimum)

Table 5-1 Comparison of hardware requirements for MATLAB/Simulink, X-Plane and FlightGear[9][10][11]

While it's possible to integrate MATLAB/Simulink with X-Plane/FlightGear type external simulators for more complex visualisations, it still ends up requiring a lot more resource than running MATLAB/Simulink alone. While the visualisation technique makes excellent improved renderings representative of the desired complex airframes and allows for manipulation of the airframe model using simple matrix manipulation techniques, it exhibits certain limitations. The aircraft is rendered by stitching together many triangles which can end up creating many orders of magnitude difference in the sizes of the V and F matrices for both the simplistic and the more complex rendering. The simplistic representation required 16 vertices and 13 faces while the MQ-9 required 28830 vertices and 9610 faces. The rendering is not meant to take place in isolation. Within the simulation framework, it will be connected to a data stream from other Simulink blocks providing it with updates on position and orientation state values. For every single update, MATLAB erases the previous rendering and redraws a new rendering based on the V and F matrices. It can therefore be concluded that the computational demand on the machine running the simulation goes up significantly when this type of complex shapes are rendered. The function patchslim [12] can be used alongside stlread [8] to reduce some of this load. These functions filter out duplicates of the vertex points and associated faces from the V and F matrices and thus reducing their size.

While not being entirely free of drawbacks, the visualisation method presented here is a lower-cost and less resource intensive alternative to the more usual methods of complex visualisation such as X-Plane. No presuppositions related to user familiarity or expertise in integration of multiple simulation platforms, exporting of visual data, CAD processes/packages, etc. are required to utilise this modification. STL files for a range of fixed-wing aircraft designs are widely available throughout various online CAD repositories (e.g. GRABCAD [13]) .Even in situations where only the 3D model file is available, there exists various means of exporting that to STL files. The ability to manipulate the model using simple matrix manipulations allows for better flexibility compared to using a CAD package. Being able to perform the visualization within the MATLAB/Simulink environment this way also reduces the need for additional computers and/or multiple software packages, which are difficult to keep in synchronisation with the simulation steps. These lock-step problems that may occur in simulation spaces, especially Real-Time, are outside the scope of this research but well documented [16]. The potential for losing synchronisation however can be minimised by reducing the number of software packages/machines comprising the holistic simulation suite.

The stall modelling using the sigmoid function [13], as shown in the literature review and methods section, allowed for incorporating the non-linear nature of aerodynamic forces. There was no existent way prescribed to determine the correct parameter settings for this model. The method of using the sigmoid curve fitting tool to import the XFLR5 polar for the FMS 182 allowed for better utilisation of the available aerodynamic data while further increasing this Simulink simulation's fidelity. The same can be said about the incorporation of automated flight data recording by the use of standard MATLAB functions. The flight data is now neatly labelled, organised and available for any further analysis. By implementing all the formulations of the models discussed in Chapter 2 and 3, it was possible to put together the complete simulation of the FMS 182 without reliance on the premium blocksets mentioned before.

5.5 References

[1] Advanced Aircraft Design: Conceptual Design, Analysis and Optimization of Subsonic Civil Airplanes, First Edition, Egbert Torenbeek, 2013 by Egbert Torenbeek, Published 2013 by John Wiley & Sons, Ltd.

[2] Hoerner SF, Fluid-Dynamic Drag, 1965.

[3] Carichner, Grant and Nicolai, Leland Malcolm. (2010). Fundamentals of Aircraft and Airship Design (vol 1). American Institute of Aeronautics and Astronautics.

[4] Schlichting H, Boundary Layer Theory, 1955.

[5] Blender, https://www.blender.org/, retrieved 05/05/2021

[6] Javascript for dref, https://github.com/havber/X-Plane-Datarefs-with-nodejs, retrieved 01/01/2021

[8] Francis Esmonde-White (2021). Binary STL file reader (https://www.mathworks.com/matlabcentral/fileexchange/29906-binary-stl-file-reader), MATLAB Central File Exchange. Retrieved 17/05/2021.

[9] MATLAB System Requirements, https://uk.mathworks.com/support/requirements/matlab-system-requirements.html, last accessed 18/05/2021

[10] X-Plane System Requirements, https://www.x-plane.com/kb/x-plane-11-system-requirements/, last accessed 18/05/2021

[11] FlightGear Hardware Recommendations, https://wiki.flightgear.org/Hardware_recommendations, last accessed 18/05/2021

[12] Francis Esmonde-White (2021). Patch Slim (patchslim.m) (https://www.mathworks.com/matlabcentral/fileexchange/29986-patch-slim-patchslim-m), MATLAB Central File Exchange. Retrieved 17/05/2021.

[13] GRABCAD, https://grabcad.com/, last accessed 17/05/2021

[14] R.l W. Beard and T. W. McLain. Small Unmanned Aircraft: Theory and Practice. Princeton

[15] G. J. Mullen, Aircraft Parameter Identification Using Matlab, Volume 11 of Cranfield College of Aeronautics report, Cranfield University, 2000.

[16] Stefan Poledna (1996). Fault-Tolerant Real-Time Systems: The Problem of Replica Determinism. p. 80. ISBN 9780585295800.

.

6 Conclusions & Future Work

6.1 Conclusions

Within the Introduction and Literature Review sections of this thesis, a range of concerns relating to the modelling and simulation of small-scale fixed wing Autonomous Aerial Vehicles have been discussed. To fulfil the ultimate aim of this research project, extensive work was conducted to meet the stated research objectives. The challenges of applying the traditional means and methods of aircraft modelling and simulation to such small platforms were studied. Potential solutions were explored within the context of accessibility of resources and user/developer expertise. The emphasis is on utilizing and developing both freely available and custom code. The limitations of existing simulation platforms discussed in this thesis were all addressed and resolved by this approach. The work undertaken resulted in the creation of multiple independent models and simulations. Combined with the large number of datasets generated for the target airframe, this converts an off-the-shelf RC fixed-wing aircraft (FMS 182) into a suitable research platform. Despite the methods/tools/techniques utilised being independent, it was found that they can be integrated into a collective platform capable of methodically studying, modelling and simulating small-scale Autonomous Aerial Vehicles. This lowers entry cost into this multidisciplinary and often expensive field of research, in terms of access to both resources and developer expertise. The following summarises the outcomes of the overall research project:

- Deploying a simple suite of measurement tools and techniques utilised during this research successfully overcame any challenges without loss of functionality or accuracy throughout the subsequent modelling and simulation work conducted.
- The range of appropriately low Reynolds number aerodynamic data obtained for the various aerofoils of the FMS 182 allowed for more accurate model creation within XFLR5 for performance testing. Thus providing a means of predicting the performance limits of the aircraft long before flight test.
- The flow visualisation outputs from XFLR5 enabled a detailed study of the FMS 182’s aerodynamic performance in various flight configurations without the

requirement for quantitative data output. Vortex formations/interference patterns informed subsequent modelling and control system design considerations. At higher angle of attack, for example, significant disruption to the airflow over the tailplane was observed, which would inform a control systems designer to set appropriate limits to retain effective control authority throughout the flight envelop. It could feasibly replace any requirement for a wind tunnel testing phase.

- The simple but detailed geometric measurements of the FMS 182, combined with the inertial modelling in XFLR5, provided output estimation of stability and control derivatives and revealed key dynamic characteristics for a range of flight configurations without risking the airframe during expensive flight test campaigns.
- Despite the challenges outlined in the Literature Review, a high fidelity model of the FMS 182 was created in X-Plane through sequential coding solutions and matrix manipulations to transcribe the XFLR5 dataset into X-Plane formats, in order to satisfactorily represent the aerodynamic performance of small scale UAVs.
- Successful validation of the propeller and power plant modelling approach suggests the Scorpion_calc simulation tool is a valuable resource capable of yielding representative performance characteristics of small electric motors/ESC/battery system combinations typical of those driving small scale UAVs, like the FMS 182, without the need for experimental bench top testing setups.
- Modified freely available Parameter Estimation MATLAB code, was successfully integrated and tested. Dynamic mode outputs from Simulink compared very well with identical data traces from X-Plane simulations, verifying state space models of the FMS 182.
- Similarly, modified freely available coding coupled with custom MATLAB functions improved flight data recording and storing processes developed for System Identification/Parameter Estimation techniques.
- Freely available MATLAB/Simulink templates and mathematical models from the literature were successfully integrated with data output from both XFLR5

and X-Plane to develop a Simulink model of the FMS 182 without the need for Aerospace Blocksets.

- Additional MATLAB functionality, using the Sigmoid function to represent an input dataset to model aerodynamic characteristics of any given aircraft, was integrated to improve the Simulink simulation's capabilities.
- Integrating standard MATLAB functions and freely available custom functions with CAD models, a solution for complex aircraft geometry visualisation was devised, reducing the hardware and software demand of any given simulation setup.
- Multiple independent models and simulation outputs were created and generated from the collective suite of simulation tools developed. Once actual flight test data becomes available, these models can be further developed and assessed for their relative advantages and drawbacks.

This research project started with the aim of exploring and developing accessible methods for modelling and simulation of small fixed-wing UAVs. As such, various unintended but impactful restrictions and limitations had to be resolved. As the above bullet points show, the work undertaken has resulted in promising solutions that might help reduce some of these limitations of accessibility. Hopefully, this will inspire a generation of prospective researchers around the world, who are currently held back by the lack of access to expensive and sophisticated resources, to contribute to this field of research.

6.2 Future Work

The outset goals were set with the intension of culminating this research with a flight test campaign to provide real time in-flight data for an analogous airframe. This would have then provided key validation data to enable the tuning of the simulation based models, to more optimally represent the performance and response of the target airframe. However, operating under pandemic restrictions during the final year of this research, forced the flight test phase to be reluctantly eliminated from the programme. Many of the following subsections therefore in-particular seek to provide a detailed outline of the originally proposed flight test programme and the intended simulation model tuning loops. Also, the individual sections elucidate how specific modules of the overall project could be improved upon.

6.2.1 Wind Tunnel Testing

Figure 6.1 An operational Wind Tunnel at Sheffield Hallam University

As discussed within the thesis, the traditional method used for obtaining the aerodynamic dataset for a given aircraft involves the use of Wind Tunnels. Computational means (e.g. XFoil/XFLR5) were utilised for this research work to generate the necessary aerodynamic data. Although care was taken to rely only on the range of data that could be validated against published literature, the vast dataset produced should allow for further validation work. Sheffield Hallam University has two operational wind tunnels. They could be utilised to create aerofoil dataset for validation and verification of the dataset created through the use of XFLR5. Due to the unavailability of experimental data for low Reynolds number (below 500k), wind tunnel testing results are required for validation of the models. This however will require the manufacture of appropriate wing sections. The following are some useful considerations:

- The test section walls are far enough away to reduce the influence on results
- Achieving appropriate flow Reynolds numbers
- Ensuring the flow remains incompressible
- Blockage area ratio remains less than 10%

The wind tunnels could be utilised to then measure the Lift force, Drag force and the Pitching moment. This should allow for the creation of a new and unique dataset for the wings considered. It would also be possible to drill pressure tappings in the wing section in order to measure the pressure profiles. The pressure plots from XFLR5 or any further CFD solutions of the wings could be compared against the results obtained from this experimental process for validation. The flight models however benefit the most from the lift, drag and moment curves. Practical considerations, such as size constraints of the wind tunnel, may make this test challenging. Care should also be taken when selecting the manufacturing process and material for the wing section such that the boundary layer behaviour is comparable to the material of the target airframe wing to achieve analogous surface roughness. The wing sections should be set up from side to side (eliminating tip vortices from each panel) and setting up a near flat, span-wise, lift distribution. This allows for the results to be approximately equivalent to a 2D infinite wing/aerofoil that can be converted to the actual target airframe’s wing and stabilisers using Prandtl’s lifting line theory. Since the aerodynamic dataset generated for this research was based on the main wing, horizontal stabiliser and vertical stabiliser, the foils used for the wind tunnel testing must be the same aerofoils.

As explained in the Discussion (Chapter 5) of the thesis, the aerodynamic characteristics of the rotating spinner of the FMS 182 remained unresolved. Such components are typically present in such small-scale aircraft and not unique to the FMS 182 airframe. Wind tunnel testing can also be utilised to generate useful data for the spinner. Similar to traditional sphere drag experiments in wind tunnels, the spinner could be mounted on a sting mount sitting in the spinner wake to reduce the interference. The sting can be set up to rotate, and instrumented to measure the torque along with the drag force. Care should be taken to ensure that the rotational speed and the airflow speed in the tunnel match that of the relevant simulation conditions described in Chapters 3 & 4. The data obtained from such a process could be used to validate simple CFD modelling of the spinner. Due to the lack of appropriate published data on such spinning components at low Reynolds numbers, this would allow for validation by the utilisation of a relatively inexpensive means of obtaining the aforementioned data.

Following this approach, a range of common aerofoil shapes and spinner sizes could be tested in the wind tunnel to lead to the creation of a database that could greatly improve the aerodynamic input accuracy for modelling and simulation of similar small-scale aerial vehicles.

6.2.2 Automatic Control

Obtaining real flight data for such small scale vehicles can be challenging. Although various flight controllers with data loggers are available to log all available data, they are seldom useful outside the range of the test conditions (see following subsections). The lack of an on-board skilled pilot and the control limitations imposed on RC pilots make traditional test manoeuvres difficult. This is where the simulation models could be tested in advance with real hardware (HITL) to ascertain limitations and check the autopilot code (SITL).

6.2.2.1 Arduino Based Preliminary Test

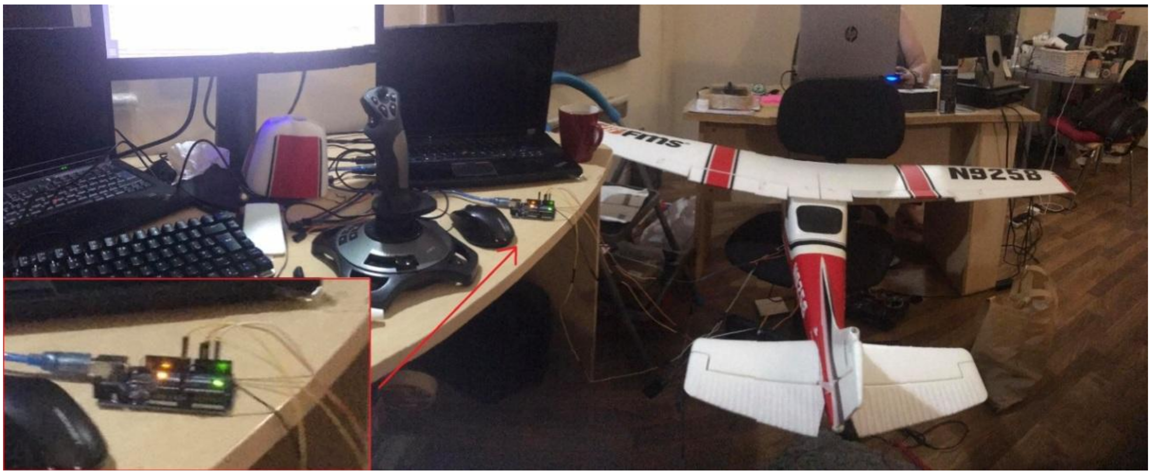


Figure 6.2 Preliminary Arduino based test setup for the FMS 182

The MATLAB support package for Arduino [1] allows for bidirectional communication between the controller and MATLAB. Part of the functionality offers the ability to send both digital and PWM signals. Since all control surfaces of small-scale aircraft, such as the FMS 182, are moved using servo motors, this helps in the development of a test bed to determine the actual physical limits of the installed servos. Two MATLAB scripts (servo_test.m & sweep_test.m) are provided in the Appendix (Chapter 7.7) as examples of preliminary servo testing. This was then integrated within the existing Simulink model and X-Plane simulation separately. The Arduino board can be initialised from the parameters file of the Simulink model as global variable. Following that, additional code was added to the m-file of the Forces & Moments block of the simulation (see Appendix), which captured the inputs from the simulator, scaled them based on the previously mentioned servo tests and passed them on to the Arduino to simulate identical control surface movements in the actual aircraft. For integration with X-Plane, basic Simulink UDP blocks were utilised to capture the inputs from X-Plane and scale them, before driving the servos in the actual aircraft as before. The custom models and functions written for them are made available in the Appendix (Chapter 7.7).

Even though this preliminary test method is incomplete requiring debugging, its usefulness in diagnosing mechanical limitation factors and actual response characteristics of the control system is very important and thus deserves further development work. Video recordings of the tests stated here have been documented and linked to in the Appendix (Chapter 7.7), which shows control surface deflection both on the physical FMS 182 airframe and also in X-Plane simultaneously in real time.

6.2.2.2 Pixhawk/PX4 Integration

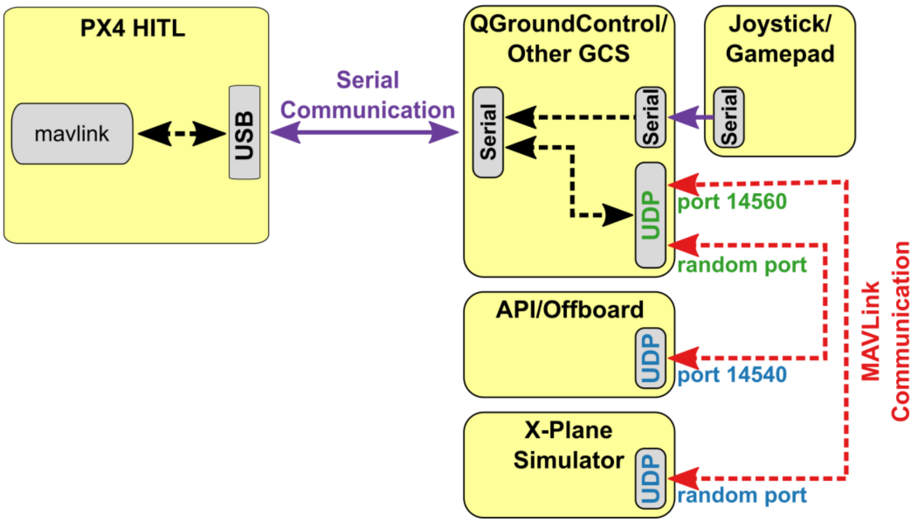


Figure 6.3 PX4 X-Plane HITL system architecture [2]

Pixhawk (controller) and PX4 (autopilot software) [2][3] are both open-source projects that are widely utilised in the drone community. Combined, they can offer HITL & SITL solutions for the simulation and modelling of the small-scale fixed-wing class of aircraft, such as the FMS 182. For the Pixhawk/PX4 platform, a Simulink support package [4] exists which allows for integration of Simulink with the architecture of the PX4. Independent of this, PX4 allows HITL/SITL testing with X-Plane via QGroundControl/MAVLINK [5][6].

Although this suite of hardware integration was briefly explored during the present research, the extent of the code development work required to successfully integrate them with the existing simulations proved too ambitious to deliver within the available time frame. It remains a very promising line of development; successfully integrating these platforms would enable rapid testing of any guidance, navigation or control algorithm both within the simulation environments and subsequent real-world test flights. This would greatly enhance the usefulness of the Simulink and X-Plane models presented here and allow for appropriate preparation of test flights.

6.2.3 State Estimation

For Automatic Flight Control, the controller will rely on a range of sensors. Every such peripheral will suffer from additive noise. This can lead to the two following major issues:

- Noisy Flight Data recorded leading to degraded quality of analysis.
- Inappropriate responses of the Flight Controller based on noisy data.

For both of these challenges, there are existent solutions in the form of online and offline estimation algorithms. The Parameter Identification method presented in this thesis is an example of an offline method where the tools and techniques are applied to existing complete flight data. But since the flight data used was obtained from simulations with no noise, there was no need to use standard filters (e.g. Kalman Filters). The same techniques applied to noisy real flight data may not remain as effective. In the case of the online method, the tools and techniques must be deployable in the flight controller and be capable of real-time filtering and estimation. This means that the computational capabilities of the flight controller could limit the type of algorithm implemented if the processing power proved insufficient.

In case of the previously mentioned Pixhawk 4 running the PX4 software, a potential solution for the online estimation is available. The Estimation & Control (ECL) library of the PX4 includes an Extended Kalman Filter implementation for noise filtering and estimation [7]. There however exists no detailed study on the number of possible filter implementations within PX4, nor their respective resource consumption and accuracy. While such problems can be addressed with better hardware (e.g. improved sensors, sophisticated mounting, etc.), that generally results in additional costs. Therefore, the development of such filters and their implementation on a platform like the PX4 could potentially result in lowering the overall cost of hardware while improving the performance of the autopilot system.

6.2.4 Flight Testing

6.2.4.1 Prerequisites

Due to the range of previously discussed limitations imposed while working with such small-scale aircraft, conventional means of flight testing methods need to be altered to suit the unique requirements of such small-scale fixed-wing platforms. In light of that, the following guidelines are proposed as prerequisites to any future flight-test campaigns for a generic target airframe:

- Obtain relevant performance estimates from the suite of analyses presented in this thesis.
- Generate simulation models and test fly them first in simulated airspace.
- Carry out HITL & SITL development described in Chapter 6.2.2.2.
- Test custom flight-test code in simulation. For example, testing of a script that provides automatic control inputs for a specific flight test.
- Ensure adequate instrumentation for data collection (GPS, IMU, Airspeed Sensors, etc.).
- Ensure necessary online/offline State Estimation and data filtering has been completed.

6.2.4.2 Tests

The simulation models of the FMS 182 were subjected to a range of tests resulting in the dataset presented and discussed in this thesis. Multiple different types of flight tests could be conducted to help improve the fidelity of these models. It is important to ensure that the test conditions of the real-world flight tests match that of the simulations. A primary set of tests can be summarised as follows.

- Flight tests from trimmed steady flight recording the dynamic responses of the aircraft to control inputs. This follows the same procedure undertaken for the X-Plane simulated flight tests. The processing of the flight data should follow the same principles described in the Parameter Identification process [8] described in Chapter 2.6/3.3.13/4.2.7. This should reveal the key stability & control characteristics of the aircraft and allow for comparison with the simulation models and facilitate their improvement.
- Flight tests across the operational envelop (take-off, climb, cruise, descent/dive, landing, etc.) to gather performance data. A comparative analysis would then identify the specific aspects of the models that require further attention.
- The experiments conducted with variable mass and shifting the centre of gravity (see Chapter 3.2.5) in XFLR5 could be replicated in real test flights. It would require extensive amount of missions flown with identical tests as shown in the above bullet points. For example, if we test for 10 mass variations and 10 centre of gravity variations, the primary set of tests would then have to be repeated 10 times for the mass variation and 10 times for the centre of gravity variation.

Details of specific control inputs, manoeuvres and analysis techniques can be found within existing literature on the subjects of Stability & Control, Aircraft Performance, System Identification/Parameter Estimation, Flight Tests, etc. [8-10].

6.2.5 **XFLR5**

The utilisation of XFLR5 resulted in a large amount of data on the modelled aerofoils of the FMS 182. While the accuracy of the results are known, within limit, the degree to which the number of panels used influences the results is unknown. Typically in FEA &

CFD methods, increasing the number of cells improves the accuracy of the results. However, every additional cell adds to the demand of computational resources. XFLR5 uses panels instead of cells but the nature of the problem remains the same. Therefore, the following steps could be taken to determine the optimum number of panels:

- Create a set of aerofoils starting with the minimum number of panels and ending with the maximum number of panels allowed.
- Perform analysis on them as described in the Methodology section.
- Import all the data into MATLAB.
- Perform the panel independence/convergence study to determine the optimum number of panels.

There will be a large amount of data from this process. The custom functions written for this research to handle XFLR5 data in MATLAB could be utilised for this process. Once the panel independence/convergence study is complete, the results should indicate the point beyond which increasing the number of panels don't lead to a significant difference in the results. While this process is standard practice in the field of CFD [11-14], it is both involved and outside the scope of this research and as such a review of the most effective ways of conducting this grid/panel refinement study is needed. Such a study would enable prospective researchers to efficiently utilise their computing power without sacrificing modelling accuracy.

6.2.6 GNU Octave

GNU Octave is a programming language environment for scientific computing [15]. It is highly compatible with MATLAB. While some aspects of the research conducted rely on Simulink, there are many that only utilises basic MATLAB code. All such code therefore could be studied and transcribed to the Free & Open-source GNU Octave, thus eliminating the requirement for even the basic MATLAB package, for groups interested in developing only the XFLR5 & X-Plane simulations.

6.3 REFERENCES

[1] Arduino Support Package, https://www.mathworks.com/hardware-support/arduino-matlab.html, 02/01/2022

[2] PX4 Autopilot Project, http://docs.px4.io/master/en/, retrieved 02/01/2022

[3] Pixhawk 4, Holybro, http://www.holybro.com/product/pixhawk-4/, 02/01/2022

[4] PX4 Support Package, MathWorks, https://www.mathworks.com/hardware-support/px4-autopilots.html, 02/01/2022

[5] QGroundControl, http://qgroundcontrol.com/, retrieved 02/01/2022

[6]MAVLink, https://mavlink.io/en/, retrieved 02/01/2022

[7] Estimation & Control Library (PX4), https://docs.px4.io/master/en/advanced_config/tuning_the_ecl_ekf.html, retrieved 02/01/2022

[8] G. J. Mullen, Aircraft Parameter Identification Using Matlab, Volume 11 of Cranfield College of Aeronautics report, Cranfield University, 2000.

[9] R. C. Nelson, Flight Stability and Automatic Control, 2nd Edition, Singapore, McGraw-Hill, 1998

[10] Raymer, Daniel P. (2006). Aircraft Design: A Conceptual Approach, Fourth edition. AIAA Education Series. ISBN 1-56347-829-3

[11] CONVERGENCE AND MESH INDEPENDENCE STUDY, https://www.computationalfluiddynamics.com.au/convergence-and-mesh-independent-study/, retrieved 02/01/2022

[12]Mesh Convergence Study, AutoDesk, https://knowledge.autodesk.com/search-result/caas/sfdcarticles/sfdcarticles/How-to-Perform-a-Mesh-Convergence-Study.html, retrieved 02/01/2022

[13]Yong Zhao, Xiaohui Su, Grid Convergence Study, in Computational Fluid-Structure Interaction, 2019

[14]Examining Spatial Grid Convergence, NASA, https://www.grc.nasa.gov/www/wind/valid/tutorial/spatconv.html, retrieved 02/01/2022

[15] GNU Octave, https://www.gnu.org/software/octave/index, retrieved 02/01/2022

7 Appendix

Due to the large number of datasets generated for this research work, it is not possible to archive them all in the Appendix. There are also various digital files that simply cannot be shared in the text format. As such bulk of this useful information is presented in an online repository. This repository can be accessed by visiting:

https://drive.google.com/drive/folders/11SfZQ53BMh9fUEXWBpL3WnqeZnOgkJLH?usp=sharing

The following subsections of the Appendix document the additional materials not shown within the main body of this thesis.

7.1 XFLR5

The following MATLAB function was used to import mass-data from XFLR5 log files onto MATLAB and output processed CSV files.

importLog.m

```
function logFile = importLog(fname)
% Parse XFLR5 v6.47 log file to extract longitudinal, lateral and control
% derivatives for each control point of a T7 analysis.
%
% Please verify the value names below if using a different version of XFLR5
%% Values needed
lonDerValues = {'Xu'; 'Xw'; 'Zu'; 'Zw'; 'Zq'; 'Mu'; 'Mw'; 'Mq'; 'Cxu';...
   'Cxa'; 'Czu'; 'CLa'; 'CLq'; 'Cmu'; 'Cma'; 'Cmq';...
   'Neutral Point position'};
latDerValues = {'Yv'; 'Yp'; 'Yr'; 'Lv'; 'Lp'; 'Lr'; 'Nv'; 'Np'; 'Nr';...
   'CYb'; 'CYp'; 'CYr'; 'Clb'; 'Clp'; 'Clr'; 'Cnb'; 'Cnp'; 'Cnr'};
conDerValues = {'Xde'; 'Yde'; 'Zde'; 'Lde'; 'Mde'; 'Nde'; 'CXde';...
   'CYde'; 'CZde'; 'CLde'; 'CMde'; 'CNde'};
%% Read file
fid = fopen(fname,'r'); % fid assigns a file id for the subsequent operations
inStr = fscanf(fid,'%c',inf); % using fscanf to look for ascii characters from the start of the file to
the end.
fid = fclose(fid); % attempting to close the file to be used subsequently
if fid; warning('File did not close properly.'); end % incase there was an error, this should tell
us!
%% Parse file
% Control points
```

```
[~, cpi] = regexp(inStr,'Calculation for control position'); % size of cpi should equal the number
of control points unless something goes wrong
controlPoints = zeros(length(cpi),1);
for i = 1:length(cpi)
   controlPoints(i) = sscanf(inStr(cpi(i)+1:cpi(i)+10),'%f'); % skips to the control point and looks
for the floating point number
end
% Longitudinal derivatives
lonDer.controlPoints = controlPoints;
for i = 1:length(lonDerValues)
   [~, vind] = regexp(inStr,sprintf(' %s',lonDerValues{i})); %creating a set of indices based on the
list of longitudinal variables defined at start
pName = lonDerValues{i}; %temp variable to accomodate special case
   if i == length(lonDerValues)
      pName = 'X_NP'; % in the log the NP has spaces in the name so we process it this way. This
will break if you change the NP bit in the end of long list above.
   end
   lonDer.(pName) = zeros(length(cpi),1); %preallocating the variable to the size of the control
points
   for ii = 1:length(vind)
      lonDer.(pName)(ii) = sscanf(inStr(vind(ii)+2:vind(ii)+15),'%f',1); % finds the associated
value and stores it in order
   end
end
% Lateral derivatives
% The same as above except that the lack of spaces in names allowed for simpler code
latDer.controlPoints = controlPoints;
for i = 1:length(latDerValues)
   [~, vind] = regexp(inStr,sprintf(' %s',latDerValues{i}));
   latDer.(latDerValues{i}) = zeros(length(cpi),1);
      for ii = 1:length(vind)
      latDer.(latDerValues{i})(ii) = sscanf(inStr(vind(ii)+2:vind(ii)+15),'%f',1);
   end
end
% Control derivatives
%same as above
conDer.controlPoints = controlPoints;
for i = 1:length(conDerValues)
   [~, vind] = regexp(inStr,sprintf(' %s',conDerValues{i}));
   conDer.(conDerValues{i}) = zeros(length(cpi),1);
      for ii = 1:length(vind)
      conDer.(conDerValues{i})(ii) = sscanf(inStr(vind(ii)+2:vind(ii)+15),'%f',1);
   end
end
%% Assign to tables
longitudinalDerivatievs = table; %allocating table objects
lateralDerivatives = table;
controlDerivatives = table;
lonFNames = fieldnames(lonDer); % pulls a list of variable names from the structure
for i = 1:numel(lonFNames)
```

```
   longitudinalDerivatievs.(lonFNames{i}) = lonDer.(lonFNames{i}); % the name from the
structure is used here for the name in the final table
end
latFNames = fieldnames(latDer);
for i = 1:numel(latFNames)
   lateralDerivatives.(latFNames{i}) = latDer.(latFNames{i});
end
conFNames = fieldnames(conDer);
for i = 1:numel(conFNames)
   controlDerivatives.(conFNames{i}) = conDer.(conFNames{i});
end
%% Assign output
logFile.LongitudinalDerivatives = longitudinalDerivatievs;
logFile.LateralDerivatives      = lateralDerivatives;
logFile.ControlDerivatives      = controlDerivatives;
%% Write CSV
[~, fnm, ~] = fileparts(fname); % [~. fnm, ~] is [folder, filename, extension]. So this only grabs
the filename of the inputfile
% This is where you write the tables containing all the extracted parameters and name them
based on the input file name.
% Thiss is where retaining field names pays off, the export function will automatically write the
field names as the top line in the csv file.
writetable(longitudinalDerivatievs,strcat(fnm,'_longitudinal_derivatives.csv'));
writetable(lateralDerivatives,strcat(fnm,'_lateral_derivatives.csv'));
writetable(controlDerivatives,strcat(fnm,'_control_derivatives.csv'));
end
```

7.2 X-Plane

The following MATLAB code (discussed in the thesis) were used for the X-Plane modelling and simulation work.

importPolar.m

```
function outData = importPolars(fname)
% Import polar data dump.
%% Read file in for parsing
fid = fopen(fname,'r');
vInfo   = fgetl(fid);   % xflr5 version for posterity
void    = fgetl(fid);   % Skip empty line
idinfo  = fgetl(fid);   % Project/foil name string
% Skip some more lines
void = fgetl(fid);  void = fgetl(fid);  void = fgetl(fid);  void = fgetl(fid);
meta0   = fgetl(fid);   % Metadata, i.e. Mach, Re, etc.
void    = fgetl(fid);   % Skip yet another empty line
meta1   = fgetl(fid);   % Variable names
varcnt  = fgetl(fid);   % Dashes under v. names to determine variable count
cInd    = regexp(varcnt, ' ');  % List spaces
varNames = cell([numel(cInd),1]);
% Select and isolate variable names
```

```
for i = 1:numel(cInd)
   if i == numel(cInd)
      varNames{i} = meta1(cInd(i)+1:end);     % Pull variable names out of meta1
   else
      varNames{i} = meta1(cInd(i)+1:cInd(i+1)-1);     % Pull variable names out of meta1
   end
   varNames{i}(varNames{i}==' ') = [];     % Remove spaces
end
in = fscanf(fid,'%f', [10 inf])';  % Load all data
% in = reshape(in,[numel(in)./numel(varNames), numel(varNames)]);
fid = fclose(fid);  % Close file and note error code
if fid; warning('Polar file did not close properly'); end        % Just in case something went wrong
%% Parse remaining metadata
% Reynolds number
[~, re0] = regexp(meta0,'Re =');
re1 = regexp(meta0,'Ncrit =');
reStrg = meta0(re0+1:re1-1);
reStrg(reStrg == ' ') = []; % Remove spaces
Re = sscanf(reStrg, '%f');  % Pull double out of string
%% Assign output
outData.Re = Re;    % Reynolds number
for i = 1:numel(varNames)
   outData.(varNames{i}) = in(:,i);    % All other variables
end
end
```

dumpAFL.m

```
function err = dumpAfl(polar,ofname)
% Export data in AFL format for copy/paste
alpha = polar.alpha;
cl = polar.CL;
cd = polar.CD;
cm = polar.Cm;
% dumpTable = [alpha cl cd cm];
fid = fopen(ofname,'w+');
fprintf(fid, '%6.1f %8.5f %8.5f %8.5f\n', [alpha, cl, cd, cm]');
err = fclose(fid);
end
```

The following code has been adapted from the free javascript referenced in the thesis.

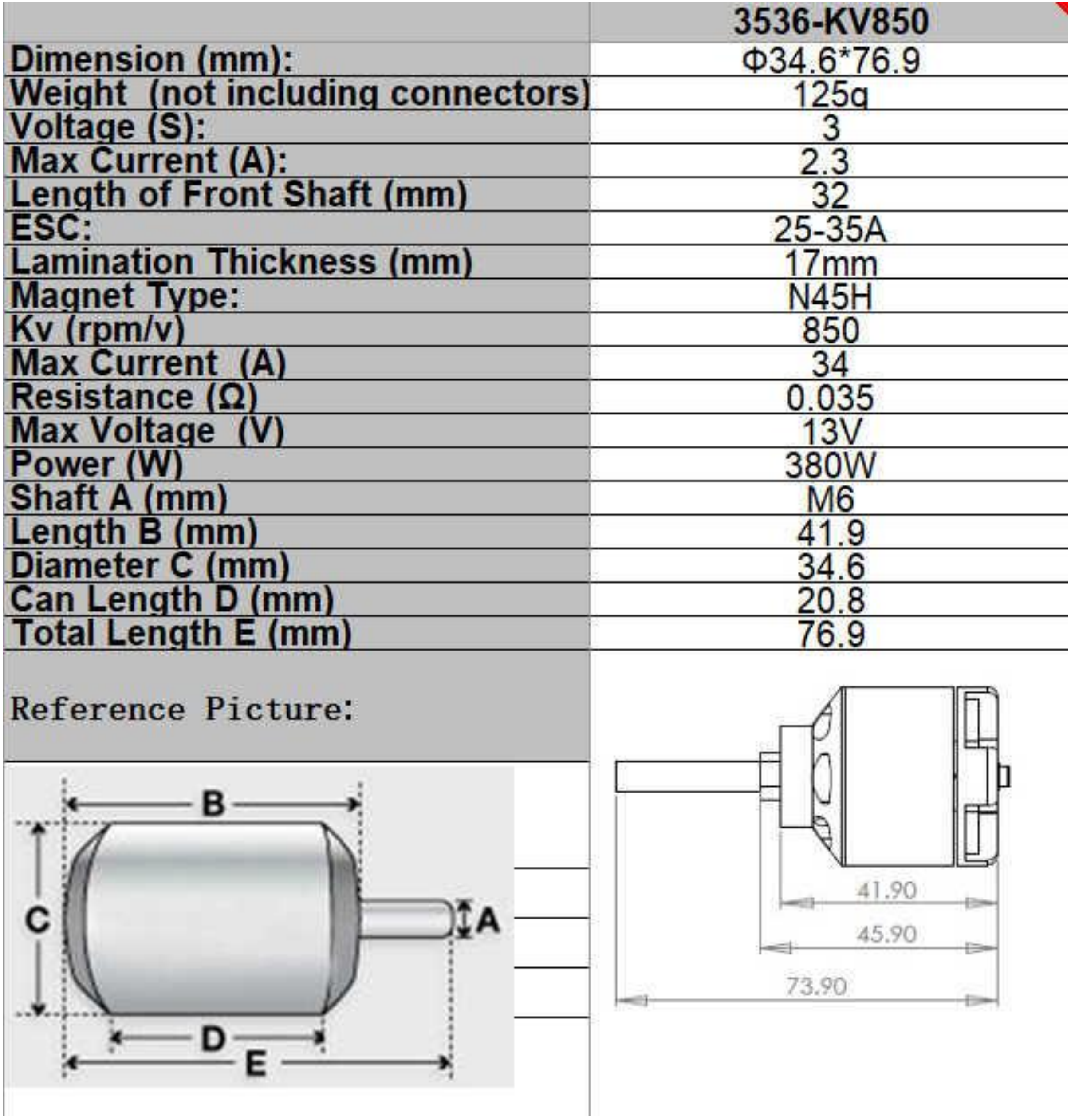
UPD.js

```
const PORT = 49000;
// const HOST = '127.0.0.1';
const HOST = '192.168.0.24';                  // 192.168.0.24
const FREQ = 100;                             // Sample rate required. This seems to track the sim's
frame rate
const fs = require('fs');             // What-for writing/reading files
let samples = 0;                      // Data sample counter
// Generate file name
const   fnamePrefix = 'udpDump';
let     fnameIndex = 0;
let    outFile = fnamePrefix.concat('_',fnameIndex,'.csv');
// Check if the file exists and increment the counter if it does
while (fs.existsSync(outFile)) { fnameIndex++; outFile =
fnamePrefix.concat('_',fnameIndex,'.csv');}
const dgram = require('dgram');
const client = dgram.createSocket('udp4');
// CSV header, keep in sync with the actual definitions below
const drefNames =        ['Time_sec',
                          'P_deg/s',
                          'Q_deg/s',
                          'R_deg/s',
                          'Pdot_deg/s^2',
                          'Qdot_deg/s^2',
                          'Rdot_deg/s^2',
                          'Vertical_speed_SI',
                          'Altitude_MSL_SI',
                          'Height_AGL_SI',
                          'alpha_deg',
                          'IAS0_knots',
                          'IAS1_knots',
                          'TAS_SI',
                          'Left_aileron_deg',
                          'Right_aileron_deg',
                          'Elevator_deg',
                          'Rudder_deg',
                          'Roll_deg',
                          'Pitch_deg',
                          'Yaw_deg',];
// Concatenate into CSV header
let nameString = drefNames[0];
for (let i = 1; i < drefNames.length; i++) {
  nameString = nameString.concat(',',drefNames[i]);
}
console.log(nameString);          // Print header to console for debugging
fs.writeFile(outFile, nameString.concat('\n'), { flag: 'w' }, err => {} );          // Write header to file
const createMessage = (dref, idx, freq) => {
  // A dataref request should be 413 bytes long
  // {
  //     label: null terminated 4 chars (5 bytes), e.g. "RREF\0"
  //     frequency: int (4 bytes)
  //     index: int (4 bytes)
  //     name. char (400 bytes)
  // }
  const message = Buffer.alloc(413);
  // Label that tells X Plane that we are asking for datarefs
  message.write('RREF\0');
  // Frequency that we want X Plane to send the data (timer per sedond)
  message.writeInt8(freq, 5);
  // Index: X Plane will respond with this index to let you know what message it is responding
to
  message.writeInt8(idx, 9);
  // This is the dataref you are asking for
  message.write(dref, 13);
  return message;
};
const messages = [
  // 'sim/name/of/dataref', index, frequency'
  // https://developer.x-plane.com/datarefs/
    // Flight time
  createMessage('sim/time/total_flight_time_sec', 1, FREQ),
  // Euler rates and derivatives, deg/s & deg/s^2
  createMessage('sim/flightmodel/position/P', 1, FREQ),                // Roll
  createMessage('sim/flightmodel/position/Q', 1, FREQ),                // Pitch
  createMessage('sim/flightmodel/position/R', 1, FREQ),                // Yaw
  createMessage('sim/flightmodel/position/P_dot', 1, FREQ),
  createMessage('sim/flightmodel/position/Q_dot', 1, FREQ),
  createMessage('sim/flightmodel/position/R_dot', 1, FREQ),
  // Vertical speed, m/s
  createMessage('sim/flightmodel/position/vh_ind', 1, FREQ),
  // Elevation above MSL, m
  createMessage('sim/flightmodel/position/elevation', 1, FREQ),
  // Elevation above groud, m
  createMessage('sim/flightmodel/position/y_agl', 1, FREQ),
  // Angle of attack, degrees
  createMessage('sim/flightmodel/position/alpha', 1, FREQ),
    // Indicated airspeed, kias
  createMessage('sim/flightmodel/position/indicated_airspeed', 1, FREQ),
  createMessage('sim/flightmodel/position/indicated_airspeed2', 1, FREQ),
  // True airspeed, m/s
  createMessage('sim/flightmodel/position/true_airspeed', 1, FREQ),
    // Ailerons deflection, deg
  createMessage('sim/flightmodel/controls/mwing10_ail1def', 1, FREQ),
  createMessage('sim/flightmodel/controls/mwing06_ail1def', 1, FREQ),
  // Elevator deflection, deg
  createMessage('sim/flightmodel/controls/hstab1_elv1def', 1, FREQ),
  // Rudder deflection, deg
  createMessage('sim/flightmodel/controls/vstab1_rud1def', 1, FREQ),
    // Roll; pitch; yaw, deg
  createMessage('sim/flightmodel/position/true_phi', 1, FREQ),
  createMessage('sim/flightmodel/position/true_theta', 1, FREQ),
  createMessage('sim/flightmodel/position/true_psi', 1, FREQ),
    /*// Position, orientation
  createMessage('sim/flightmodel/position/latitude', 1, FREQ),
  createMessage('sim/flightmodel/position/longitude', 1, FREQ),
  createMessage('sim/flightmodel/position/mag_psi', 1, FREQ),*/
  // Add as many as you like (within X Plane's recommended limitation)
];
// "Listen" event handler
client.on('listening', () => {
  const address = client.address();
  console.log(`UDP client listening on ${address.address}:${address.port}`);
});
```

```
// "Message received" event handler
client.on('message', (message, remote) => {
  // Message structure received from X Plane:
  // {
  //     label: 4 bytes,
  //     1 byte (for internal use by X Plane)
  //     index: 4 bytes
  //     value: float - 8 bytes x n
  // }
  // Read the first 4 bytes. This is the label that x-plane responds with to indicate
  // what type of data you are receiving. In our case, this should be "RREF". If it is
  // not, ignore the message.
  // The next byte (offset 4) is used by x plane, and not of interest
  // The index (at offset 5) is the index that you specified in the message. To specify
  // which request X Plane is responding to
  // The values start at offset 9. 8 bytes per value. Values will appear in the same order
  // as the requested values
  const label = message.toString('utf8', 0, 4);
  if (label !== 'RREF') {
    console.log('Unknown package. Ignoring');
  } else {
        // let idxoffset = 5;
    let msgoffset = 9;    // Keeping track of where we are in the message X-Plane sent back
    let messages = [];
        let valnum = 0;              // Value counter
        let msgstring = '';
    // RREFs values are floats. They occupy 8 bytes. One message can contain several values,
    // depending on how many you asked for. Read every value by iterating over message and
    // increasing the offset by 8.
    while (msgoffset < message.length) {
      //const index = message.readFloatLE(idxoffset);
      // Decode value
      const value = message.readFloatLE(msgoffset);
      messages.push(value);
            // Append to CSV output line, reconstructed every sample
      if (valnum == 0) { msgstring = msgstring.concat(value); }
      else { msgstring = msgstring.concat(',',value); }
      //console.log('Value' + valnum + '(' + index + ')' + ': ' + value);
      //console.log(drefNames[valnum] + ': ' + value);
      msgoffset += 8;    // Double floating point values are made up of 8 bytes
      valnum++;
    }
    // Append CSV line to output file
    fs.writeFile(outFile, msgstring.concat('\n'), { flag: 'a' }, err => {} );
    // Increment sample counter and report in terminal window
    samples++;
    console.log('Sample: ' + samples);
    // Do something with the values (e.g. emit them over socket.io to a client, or whatever)
  }
});
```

```
// This is what actually requests the DREF data from X-Plane, everything else leading up to this
point is setting things up in memory
for (let i = 0; i < messages.length; i++) {
  client.send(messages[i], 0, messages[i].length, PORT, HOST, (err, bytes) => {
    if (err) {
      console.log('Error', err)
    } else {
      console.log(`UDP message sent to ${HOST}:${PORT}`);
    }
  });
}
```

Data sheet for FMS 3536 KV850 Brushless motor (source: https://www.aliexpress.com/item/638882178.html):



	3536-KV850
Dimension (mm):	Φ34.6*76.9
Weight (not including connectors)	125g
Voltage (S):	3
Max Current (A):	2.3
Length of Front Shaft (mm)	32
ESC:	25-35A
Lamination Thickness (mm)	17mm
Magnet Type:	N45H
Kv (rpm/v)	850
Max Current (A)	34
Resistance (Ω)	0.035
Max Voltage (V)	13V
Power (W)	380W
Shaft A (mm)	M6
Length B (mm)	41.9
Diameter C (mm)	34.6
Can Length D (mm)	20.8
Total Length E (mm)	76.9
Reference Picture:	

B
C
A
D
E
41.90
45.90
73.90

Figure 7.1 FMS Brushless motor datasheet

Custom attachment for Motor testing:

This custom attachment was made from sheet metal drilled to fit the small motor of the FMS 182 such that it could be mounted on the test rig discussed in the thesis.

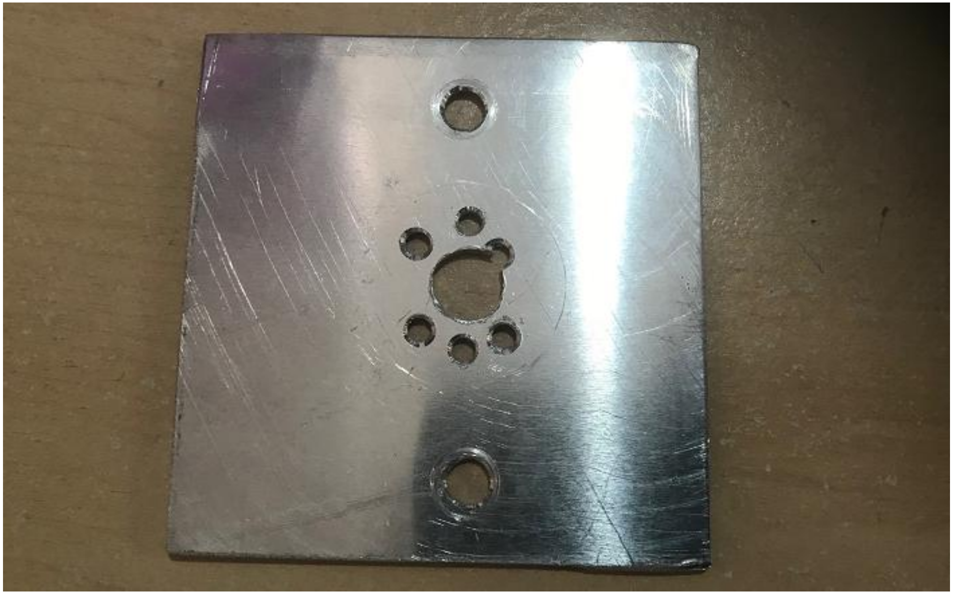


Figure 7.2 Custom attachment

Parameter identification code:

loadUDPdata.m

```
function dataSet = loadUdpData(lddir)
% Scan current or requested directory and load all csv files from it
if nargin<1
   lddir = pwd;
end
filesInDir = dir(lddir);
f_ind = [];
for i = 1:numel(filesInDir)
   if min(ismember('.csv',filesInDir(i).name))
      f_ind = [f_ind;i];
   end
end
if isempty(f_ind)
   error('No csv files present');
end
fnames = cell(numel(f_ind),1);
varNames = fnames;
for i = 1:numel(f_ind)
   fnames{i} = filesInDir(f_ind(i)).name;
   [~, varNames{i}, ~] = fileparts(fnames{i});
end
for i = 1:numel(fnames)
   dataSet.(varNames{i}) = udpDumpImport(fnames{i});
end
```

```
end
```

reprocessData.m

```
function newDataSet = reprocessDataSet(oldDataSet)
% This function receives an entire set (made up of multiple CSV files) as
% returned by loadUdpData and reprocesses it so it is compatible with other
% existing code.
%
% Features:
%   - Remove duplicate entries
%   - Flag up multiple data sets in the same file [MAYBE]
%   - Reset simulation timer to t0 = 0
%   - Resample data to a constant sample rate
%% Control variables
resamplingFrequency = 50;   % Hz
dataNames = fieldnames(oldDataSet);
%% Remove duplicates
for i = 1:numel(dataNames)
   newDataSet.(dataNames{i}) = unique(oldDataSet.(dataNames{i}),'rows');
end
%% Look for multiple data sets?
%% Reset time to t0 = 0
for i = 1:numel(dataNames)
   newDataSet.(dataNames{i}).Time_sec = newDataSet.(dataNames{i}).Time_sec -
newDataSet.(dataNames{i}).Time_sec(1);
end
%% Resample data
for i = 1:numel(dataNames)
   tableFieldNames = oldDataSet.(dataNames{i}).Properties.VariableNames;
   resampledTime = cell(size(tableFieldNames));
   for j = 1:numel(tableFieldNames)
      % Skip resampling time
      if strcmpi(tableFieldNames{j},'Time_sec')
         continue
      end
      % Resample and assign to output. Keep note of resampled time for
      % later
      [newDataSet.(dataNames{i}).(tableFieldNames{j}), resampledTime{j}]...
         = resample(oldDataSet.(dataNames{i}).(tableFieldNames{j}),...
         oldDataSet.(dataNames{i}).Time_sec, resamplingFrequency);
         end
      % Verify resampled time
   for k = 1:numel(resampledTime)-1
      if ~isequal(resampledTime{i},resampledTime{i+1})
         error('Resampling function done a dumb!');
      end
   end
```

```
end
end
```

runScript.m

```
%% Controls
checkPlot      = false;
runCalc        = true;
cutoffSample   = 502;
generatePlots  = false; % Enable plots in ls_long and ls_sp
testToRun      = 'pitchStabHoomanHigh';
varToPlot      = 'Roll_deg';
%% Load data
dataSet = loadUdpData;
%% Tidy up data
dataSetReprocessed = reprocessDataSet(dataSet);
%% Plot check data
if checkPlot
  testFig = figure;
  testAx = axes;
  hold(testAx,'on');
  plot(testAx,dataSet.(testToRun).Time_sec-dataSet.(testToRun).Time_sec(1),...
    dataSet.(testToRun).(varToPlot),'-k');
  plot(testAx,dataSetReprocessed.(testToRun).Time_sec,...
    dataSetReprocessed.(testToRun).(varToPlot),'--r');
end
%% Compute longitudinal static stability and phugoid
if runCalc
  testResult = computeLSS(dataSetReprocessed.(testToRun),cutoffSample,generatePlots);
  %% Transfer function
[nQDeltaE1,dQDeltaE1]=ss2tf(testResult.sp.A,testResult.sp.B,testResult.sp.C(2,:),testResult.sp.D(2,:),1)
  %% TF
  z=[dataSetReprocessed.(testToRun).Q_deg,dataSetReprocessed.(testToRun).Elevator_deg];
  nn=[3 2 0];
  th=oe(z,nn,100,0.01,1.6,4096,1);
  th=sett(th,0.02);
  present(th)
  [dnum,dden] = th2tf(th);
  [qnum2,qden2] = d2cm(dnum,dden,0.02,'tustin');
  qnum2 = qnum2(2:3);    % Get rid of that 2nd order term, it's inconvenient
  qnum2
  qden2
  end
```

computeLSS.m

```
function stabilityDerivatives = computeLSS(rawDataTable, cutoffSample,generatePlots)
% Compute longitudinal static stability derivatives using Cranfield method
% (see referenced paper). This is the driving function which should make everything
% else work with the raw data captured from X-Plane via UDP and reprocessed
% accordingly.
%
% rawDataTable is the data table generated by the UDP import function and
% held within the dataSet structs. This should be de-nested for the purpose
% of this function.
%
% cutoffSample is the last sample just before the actual test starts. It's
% used by the Cranfield algorithms to establish baseline values and remove
% zero offsets such that the following code can compute accurate estimates.
%
% The Cranfield code has been modified to work with current matlab (2017b)
% as well as to present neater code, but functionally it should remain
% identical.
pro_sp1_output = pro_sp1(rawDataTable,cutoffSample);
ls_sp_output = ls_sp(pro_sp1_output,generatePlots);
pro_lof_output = pro_lof(rawDataTable,cutoffSample);
ls_long_output = ls_long(pro_lof_output,generatePlots);
stabilityDerivatives.sp=ls_sp_output;
stabilityDerivatives.long=ls_long_output;
fprintf(1,'Stability output SP1\n');
damp(ls_sp_output.A);
fprintf(1,'Stability output long\n');
damp(ls_long_output.A);
end
```

The above code was used with the code from G.J. Mullen as explained in the thesis. The original Matlab files are available in the appendix of the source. They need to be upgraded to more recent MATLAB code manually based on whatever MATLAB version the above code are to be used with. All the code that was converted is available in the web link provided with the Appendix.

7.3 Simulink

The empty template files utilised for this thesis can be obtained from the Beard & MClain (2012) reference cited in the thesis. The following are the modifications done for modularity and development mentioned in the thesis:

MATLAB code for the example Pyramid:

```
%Pyramid code
clear all
% Physical location of the vertices on the graph
V = [1 1 0; 1 -1 0; -1 -1 0; -1 1 0; 0 0 -3];
% Defining the 5 surfaces/faces based on the vertices
```

```
F = [1 2 3; 1 4 3;1 2 5; 2 3 5; 3 4 5; 4 1 5];
% colors
red = [1 0 0];
green = [0 1 0];
blue = [0 0 1];
yellow = [1 1 0];
colors = [green; green; green;yellow;red;blue];
%for graph
title('Pyramid')
xlabel('x') %label for the x-axis
ylabel('y') %label for the y-axis
zlabel('-z') %label for the z-axis
view(50,20)  % initial the vieew angle for figure
axis([-3 3,-3 3,-4 4]); % axis limiters for x y & z
grid minor %
%This is where it all is put together into one 3D object
patch('Vertices',V,'Faces',F,'FaceVertexCData', colors,'FaceColor','flat')
```

MATLAB function (Geometry.m) to import V & F data for visualisation within the simulation:

```
% GEOMETRY INFORMATION : Manipulate this section for your own design
function [V,F,colours] = Geometry
% This is where the location of the vertices are specified in the form of
% triplets.
load V.mat %loading the V matrix from the scaleddemorq9.m file
V = V'; %transposing it to match expected form
% If you however have your own geometry and the vertices information, you
% may load them in via:
% V = [triplet 1; triplet 2;....]'; % the triplets are the coordinates for
% the points
% define faces as a list of vertices numbered above
load F.mat %loading the F matrix from the scaleddemorq9.m file
F = F;
% If you have your own custom geometry, you can create the F matrix here in
% the format:
% F =[triplet;triplet;...so on and on]; % the number of triplets there
% depends on the number of faces your design makes from the vertices
% Color triplets from  https://uk.mathworks.com/help/matlab/ref/colorspec.html
yellow = [1 1 0];
magenta = [1 0 1];
cyan = [0 1 1];
red = [1 0 0];
green = [0 1 0];
blue = [0 0 1];
white = [1 1 1];
black = [0 0 0];
```

```
% With these triplets, we can create a matrix (size of F) containing the
% colour information for every face. I am going with just one colour so I do:
bs = ones(size(F)); % This is just to help the size of the colour matrix match the
expected matrix size
colours = bs.*green;
%If you need specific color definition for every surface, you may follow
%the format:
%    colours = [...
%    green;...    % color corresponding to the first face
%    .
%    . so on and on
%    red;...     % color corresponding to the last face
%    ];
% This matrix needs to match the size of the F matrix.
% If you want to play around with gradients and more complex colouring, look up
vertexcdata https://uk.mathworks.com/help/matlab/ref/patch.html
end
```

MATLAB code for rendering the 3D aircraft(Render.m):

```
%This is where carry out the plotting/drawing/rendering via the patch
%function. Use of handle in graphics is presented in Appendix C (Animation in
%Simulink) of Small Unmanned Aircraft Theory & Practice (Beard & McLain
%2012). The sample code has been adapted to fit this particular demo.
function handle = Render(V,F,colours,n,e,d,phi,theta,psi,handle)
% this uses the rotate.m file to rotate the geometry
% As shown in this example (http://planning.cs.uiuc.edu/node99.html),
% we first rotate and then translate
V = rotate(V, phi, theta, psi);
% this uses the translate.m file to translate the geometry
V = translate(V, n, e, d);
% Axis transformation
% The second row of R needs modification for this specific STL(eg. mq9.stl)
% as we need to rotate about the z-axis. It is set to -1 0 0 instead of 1 0 0 for the
% necessary rotation to match the MATLAB coordinate system.
R = [0 -1 0 ; -1 0 0; 0 0 1];
V = R*V; % The V matrix transformed
% During the initialisation, the first drawing is made when the handle
% is empty (empty array passed to handle/see input argument for render in the
% if statement for t==0 set to []).
if isempty(handle)
handle = patch('Vertices', V', 'Faces', F,'FaceVertexCData',colours,'FaceColor','flat');
% The rendering is done repeatedly for every step in time by simply changing
% the property of the
else
set(handle,'Vertices',V','Faces',F);
```

```
grid on
drawnow
end
end
```

Function (rotate.m) used for handling rotation of the imported aircraft body from V & F matrices:

```
%Use this function to rotate every vertex of the imported STL
%through the 3 axis
function vpoints=rotate(vpoints,phi,theta,psi)
  % We are using a right hand rotation. Use appropriate rotation matrices
  % here. See details: https://mathworld.wolfram.com/RotationMatrix.html
  RollMat = [1 0 0; 0 cos(phi) sin(phi); 0 -sin(phi) cos(phi)];
  PitchMat = [cos(theta) 0 -sin(theta); 0 1 0; sin(theta) 0 cos(theta)];
  YawMat = [cos(psi) sin(psi) 0; -sin(psi) cos(psi) 0; 0 0 1];
  %Make your rotation matrix
  R = RollMat*PitchMat*YawMat;
  % Make it right-handed
  R = R';
  % Simply multiply vpoints with the above Rotation Matrix (R)
  vpoints = R*vpoints;
  end
```

Separate MATLAB function (translate.m) for handling translation of renderings:

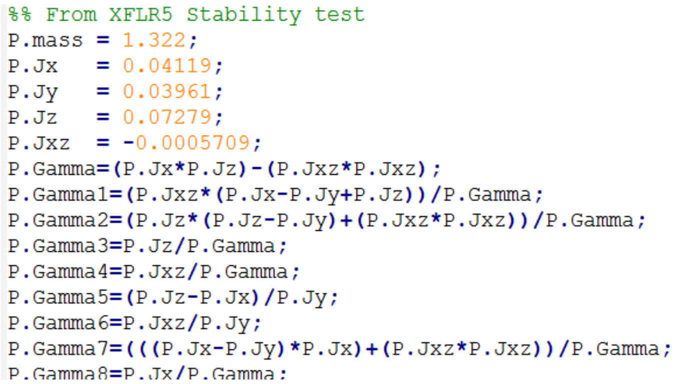
```
% The inputs n,e & d provide the surge/sway/heave translational inputs to
% inform where in the 3D space the points of the vertices (vpoints) moved to. The
% following function handles that in similar fashion to Beard & McLain.
function vpoints = translate(vpoints,n,e,d)
% The n,e&d inputs are the amounts by which the vpoints need to move. That
% move is to be made from previous-vpoints to current-vpoints by updating
% it with the n,e&d inputs.
% For a given x y z input, we need to create a matrix that when added to
% the vpoints matrix will represent the new position.
input = [n;e;d]; % These are the n e d values you input through the slx
% The dimensions must match so we use repmat
% (https://uk.mathworks.com/help/matlab/ref/repmat.html#d123e1164936) to
% create and match sizes. We can initialise a matrix based on "input" in a
% similar fashion as shown in the Mathworks examples. In the example a 3 by
% 2 matrix is created from a scaler number (10) by repmat(10,3,2). The
% vpoints matrix is 3x28830 size and so should the updatevpoints .
dim2 = size(vpoints,2); % This will give us the second dimension of vpoints
updatevpoints = repmat(input,1,dim2);
vpoints = vpoints + updatevpoints; % Just add them to get the current pos.
end
```

MATLAB Function for visualisation with kinematic inputs (Visualisation.m):

Modifications made to work with the modular functions:

```
function Visualisation(u,V,F,colours)
n       = u(1);      % North position
e       = u(2);      % East position
d       = u(3);      % Down Position
phi      = u(4);      % roll angle
theta    = u(5);      % pitch angle
psi      = u(6);      % yaw angle
t        = u(7);      % time
% This only applies to the initialisation when the simulation time is
% zero and the rendering and Vertices, Faces and facecolors are
% initialised.
if t==0
figure,
[Vertices,Faces,facecolors] = Geometry;
render_handle = Render(Vertices,Faces,facecolors,n,e,d,phi,theta,psi,[]);
% Beyond the initialisation period, we just call the following function
% to keep updating the visuals for every simulation step
else
   Render(Vertices,Faces,facecolors,n,e,d,phi,theta,psi,render_handle);
   End
```

Dynamics block (dynamics.m) and Forces&Moments block can be generated from the default MATLAB templates for s-function blocks with the MATLAB format of the derivation (see Literature) equation outputs being assigned as system output (see Simulink documentation) and the structure provided in the template. Sufficient code modification and enhancement details are already provided within the thesis. The nParam.m file mentioned in the thesis is entirely constructed out of text file containing all the mentioned parameters and their values in the format "P.Parameter = value".



```
%% From XFLR5 Stability test
P.mass = 1.322;
P.Jx   = 0.04119;
P.Jy   = 0.03961;
P.Jz   = 0.07279;
P.Jxz  = -0.0005709;
P.Gamma=(P.Jx*P.Jz)-(P.Jxz*P.Jxz);
P.Gamma1=(P.Jxz*(P.Jx-P.Jy+P.Jz))/P.Gamma;
P.Gamma2=(P.Jz*(P.Jz-P.Jy)+(P.Jxz*P.Jxz))/P.Gamma;
P.Gamma3=P.Jz/P.Gamma;
P.Gamma4=P.Jxz/P.Gamma;
P.Gamma5=(P.Jz-P.Jx)/P.Jy;
P.Gamma6=P.Jxz/P.Jy;
P.Gamma7=(((P.Jx-P.Jy)*P.Jx)+(P.Jxz*P.Jxz))/P.Gamma;
P.Gamma8=P.Jx/P.Gamma;
```

The only calculation made are the Gamma values used by the equations used for the simulation which are calculated based on the FMS 182 parameters as shown above.

Scripts such as nGeneratorLMN.m was used to output the specific formulation of the Moments equations for the block based on FMS 182 parameters (same for Forces) :

```
clear
nParam;
syms theta phi Va p q r delta_e delta_a delta_r delta_t alpha beta;
A1=[0.5*P.rho*(Va*Va)*P.S_wing]
A2=[...
P.b*(P.C_ell_0+(P.C_ell_beta*beta)+(P.C_ell_p*(P.b/(2*Va))*p)+(P.C_ell_r*(P.b/(2*Va))*r)+(P.
C_ell_delta_a*delta_a)+(P.C_ell_delta_r*delta_r));...

P.c*(P.C_m_0+(P.C_m_alpha*alpha)+(P.C_m_q*(P.c/(2*Va))*q)+(P.C_m_delta_e*delta_e));...
P.b*(P.C_n_0+(P.C_n_beta*beta)+(P.C_n_p*(P.b/(2*Va))*p)+(P.C_n_r*(P.b/(2*Va))*r)+(P.C_n_
delta_a*delta_a)+(P.C_n_delta_r*delta_r));...
   ]
 B=[...
   -P.k_T_P*(P.k_Omega*delta_t)^2;...%*(P.k_Omega*delta_t));...
   0;...
   0;...
   ]
FinalOld= (A1*A2)+B
Fin=A1*A2
B1=B(1)
Fin(1)=Fin(1)-B(1);
FinalNew=Fin
%Result = FinalOld/FinalNew
% Result: This is the l m n matrix. don't run it like a code
% (69751*Va^2*((7239*delta_a)/31250 - (21717*beta)/62500 + (152019*delta_r)/500000 -
(681240573*p)/(625000000*Va) + (366821847*r)/(625000000*Va)))/200000
%                         -(69751*Va^2*((180443*alpha)/2500000 + (9497*delta_e)/100000 +
(811737081*q)/(12500000000*Va) + 11101993/2500000000))/200000
%   (69751*Va^2*((7239*beta)/10000 + (21717*delta_a)/125000 - (7239*delta_r)/78125 +
(576434331*p)/(6250000000*Va) - (366821847*r)/(250000000*Va)))/200000
%
%% FMS 182 lmn
-(169995869*Va^2*((2042123578229226986 7*beta)/4611686018427387904000 -
(3275879529536456763*delta_a)/7205759403792793600 + (606171*delta_r)/250000000 +
(8800849064558605800155 49*p)/(18014398509481984000000000*Va) -
(5898109508594069179868 1*r)/(9007199254740992000000000*Va)))/1000000000
-(169995869*Va^2*((134289077*alpha)/500000000 +
(2124043554118379715 9*delta_e)/5629499534213120000 0 +
(678543851713*q)/(2500000000000*Va) -
8287323209772602706 1/7205759403792793600000))/1000000000
 (169995869*Va^2*((30012513*beta)/200000000 + (42291*delta_a)/8000000 -
(8034794849825592105 9*delta_r)/7205759403792793600 00 -
(1430879189385966561652599*p)/(576460752303423488000000000*Va) -
(2021236134939*r)/(20000000000000*Va) +
1164996005407734282 63/1180591620717411303424 0000))/1000000000
```

MATLAB code for the sigmoid function demonstration:

```
clear all
clf
```

```
P.M = 9.471;%50; % This is the transition rate. This determines how quickly (Cl/Alpha) the airfoil's Cl drops per unit of alpha. It's always a positive constant.
P.alpha0 = 0.2657; %0.4712; % Angle of attack reference condition (initial)
P.C_L_0 = 0.1064;%0.28; % The value of the lift coefficient when alpha = 0.
P.C_L_alpha = 6.642; %3.45; % Stability derivative: Change in lift coefficient for a unit change in alpha
alpha = [-pi/2:1e-4:pi/2]; % This is just us declaring the range of Alpha for which we wish to do the following maths and plot later
%This is where we do the maths. Notice that we have broken down the
%equations into managable pieces. Please refer back to the original
%equations to see the full picture.
%Calculations to determine the values for the sigmoid function used for
%blending
Sigmaofalpha1 = 1+ exp(-P.M.*(alpha-P.alpha0)) + exp(P.M.*(alpha+P.alpha0));
Sigmaofalpha2a = 1+ exp(-P.M.*(alpha-P.alpha0));
Sigmaofalpha2b = 1+ exp(P.M.*(alpha+P.alpha0));
Sigmaofalpha2c = Sigmaofalpha2a.*Sigmaofalpha2b;
Sigmaofalpha = Sigmaofalpha1./Sigmaofalpha2c;
%Calculating the values for lift coefficient variation with alpha for the
%flat plate
Clofflatplate = 2.*sign(alpha).*(sin(alpha).^2).*cos(alpha);
%Calculating the values for lift coefficient variation with alpha for the
%generic linear model
Cloflinear = P.C_L_0 + P.C_L_alpha.*alpha;
%This is where we blend the two with the sigmoid function
Clofalpha = (1- Sigmaofalpha).*(Cloflinear) + Sigmaofalpha.*(Clofflatplate);
%This is where we plot the curves:
% The linear model curve represented with '-.'
plot(alpha.*(180./pi),Cloflinear, '-.')
hold on
% The flat-plate model curve represented with '--'
plot(alpha.*(180./pi), Clofflatplate,'--')
hold on
% The combined and more realistic model curve represented with '-.'
plot(alpha.*(180./pi), Clofalpha,'-')
hold on
title('Cl against Alpha');
xlabel('Alpha (degrees)');
ylabel('Cl');
axis([-45 45 -2 2]);
```

CL_alpha.m function:

```
function CL = CL_alpha(alpha,M,alpha0,CL0,CLa)
sigmaAlpha  = @(alpha,M,alpha0)...
   (1+exp(-M.*(alpha-alpha0))+(exp(M.*(alpha+alpha0))))...
   ./...
   ((1+exp(-M.*(alpha-alpha0))).*(1+exp(M.*(alpha+alpha0))));
CL = (1-sigmaAlpha(alpha,M,alpha0))...
   .*...
   (CL0+CLa.*alpha)...
   + sigmaAlpha(alpha,M,alpha0)...
   .*...
   (2.*sign(alpha).*(sin(alpha).^2).*cos(alpha));
End
```

fitSigmaAlpha.m

```
function [fitresult, gof] = fitSigmaAlpha(alpha, CL)
%CREATEFIT(ALPHA,CL)
%  Create a fit.
%
%  Data for 'sigmaAlpha' fit:
%      X Input : alpha
%      Y Output: CL
%  Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of fit info.
%
%  See also FIT, CFIT, SFIT.
%  Auto-generated by MATLAB on 22-Aug-2021 17:45:31

%% Housekeeping
if max(alpha) > (2*pi)  % If alpha exceeds 2pi, it's clearly not in radians
   alpha = deg2rad(alpha); % Therefore convert it to radians before this mess gets any worse
end
%% Fit: 'sigmaAlpha'.
[xData, yData] = prepareCurveData( alpha, CL );
% Set up fittype and options.
ft = fittype( 'CL_alpha(alpha,M,alpha0,CL0,CLa);', 'independent', 'alpha', 'dependent', 'CL' );
% ft = fittype( 'CL_alpha(alpha,M,alpha0,0.0422,5.2128);', 'independent', 'alpha', 'dependent',
'CL' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.Algorithm = 'Levenberg-Marquardt';
opts.Display = 'Off';
opts.MaxFunEvals = 20000;
opts.MaxIter = 5000;
opts.Robust = 'Bisquare';
opts.StartPoint = [0.0731 5 50 0.5];
% opts.StartPoint = [50 0.5];
% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
% Plot fit with data.
figure( 'Name', 'sigmaAlpha' );
h = plot( fitresult, xData, yData );
legend( h, 'CL vs. alpha', 'sigmaAlpha', 'Location', 'NorthEast' );
% Label axes
xlabel alpha
ylabel CL
grid on
```

```
end
%% Nested functions
function CL = CL_alpha(alpha,M,alpha0,CL0,CLa)
% Calculate CL, including sigma
sigmaAlpha  = @(alpha,M,alpha0)...
   (1+exp(-M.*(alpha-alpha0))+(exp(M.*(alpha+alpha0))))...
   ./...
   ((1+exp(-M.*(alpha-alpha0))).*(1+exp(M.*(alpha+alpha0))));
CL = (1-sigmaAlpha(alpha,M,alpha0))...
   .*...
   (CL0+CLa.*alpha)...
   + sigmaAlpha(alpha,M,alpha0)...
   .*...
   (2.*sign(alpha).*(sin(alpha).^2).*cos(alpha));
End
```

generatecurve.m

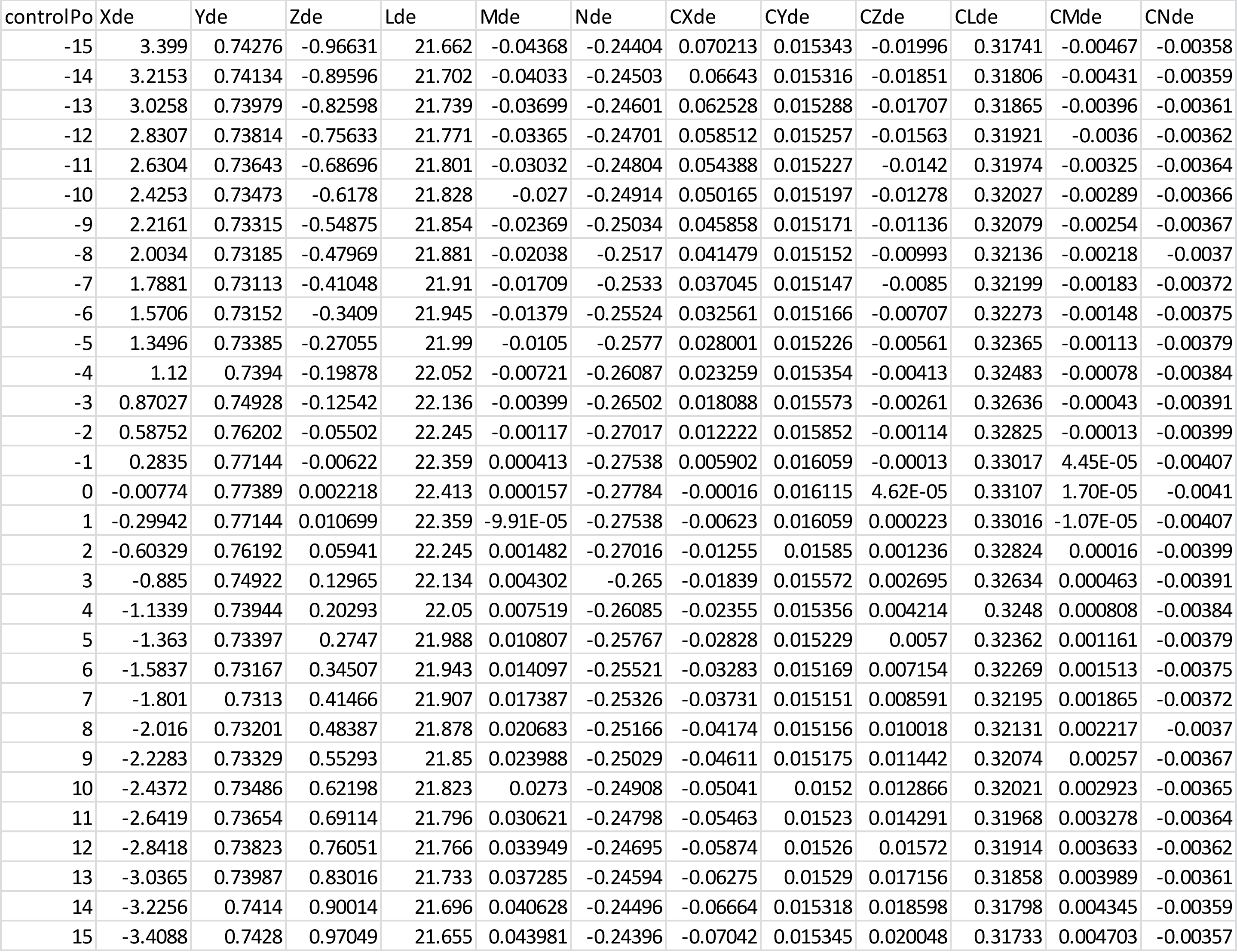
```
% First load the csv as column vectors
% Second set the AR to correct valuie
alpha = deg2rad(alpha1);
AR = 7.4;
CL = CL.*AR/(AR+2);
[fitOut, gof] = fitSigmaAlpha(alpha,CL)
```

7.4 Stability & Control (From XFLR5)

The following pages contain the research data generated from large number of trials for documentation and reference.

The order of the dataset:

1. Control derivatives from aileron deflection (base model/SC0)
2. Control derivatives from rudder deflections (base model/SC0)
3. Dataset used for determining Elevator deflections up related derivatives (base model/SC0)
4. Dataset used for determining Elevator deflections down related derivatives (base model/SC0)
5. Lateral stability derivatives for aileron deflection (base model/SC0)
6. Longitudinal stability derivatives for aileron deflection (base model/SC0)
7. Lateral stability derivates for rudder deflection (base model/SC0)
8. Longitudinal stability derivatives for rudder deflection (base model/SC0)

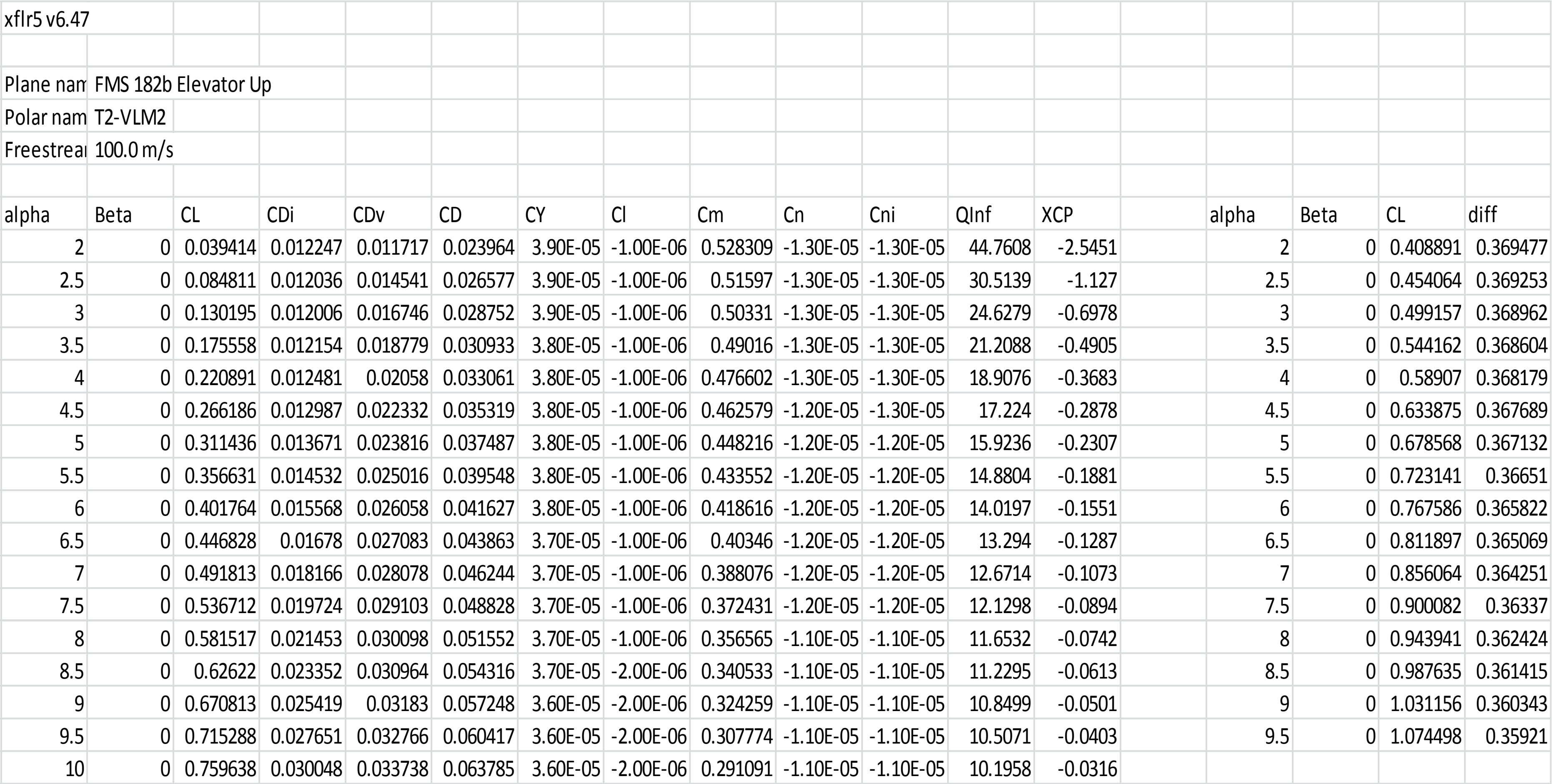


controlPo	Xde	Yde	Zde	Lde	Mde	Nde	CXde	CYde	CZde	CLde	CMde	CNde
-15	3.399	0.74276	-0.96631	21.662	-0.04368	-0.24404	0.070213	0.015343	-0.01996	0.31741	-0.00467	-0.00358
-14	3.2153	0.74134	-0.89596	21.702	-0.04033	-0.24503	0.06643	0.015316	-0.01851	0.31806	-0.00431	-0.00359
-13	3.0258	0.73979	-0.82598	21.739	-0.03699	-0.24601	0.062528	0.015288	-0.01707	0.31865	-0.00396	-0.00361
-12	2.8307	0.73814	-0.75633	21.771	-0.03365	-0.24701	0.058512	0.015257	-0.01563	0.31921	-0.0036	-0.00362
-11	2.6304	0.73643	-0.68696	21.801	-0.03032	-0.24804	0.054388	0.015227	-0.0142	0.31974	-0.00325	-0.00364
-10	2.4253	0.73473	-0.6178	21.828	-0.027	-0.24914	0.050165	0.015197	-0.01278	0.32027	-0.00289	-0.00366
-9	2.2161	0.73315	-0.54875	21.854	-0.02369	-0.25034	0.045858	0.015171	-0.01136	0.32079	-0.00254	-0.00367
-8	2.0034	0.73185	-0.47969	21.881	-0.02038	-0.2517	0.041479	0.015152	-0.00993	0.32136	-0.00218	-0.0037
-7	1.7881	0.73113	-0.41048	21.91	-0.01709	-0.2533	0.037045	0.015147	-0.0085	0.32199	-0.00183	-0.00372
-6	1.5706	0.73152	-0.3409	21.945	-0.01379	-0.25524	0.032561	0.015166	-0.00707	0.32273	-0.00148	-0.00375
-5	1.3496	0.73385	-0.27055	21.99	-0.0105	-0.2577	0.028001	0.015226	-0.00561	0.32365	-0.00113	-0.00379
-4	1.12	0.7394	-0.19878	22.052	-0.00721	-0.26087	0.023259	0.015354	-0.00413	0.32483	-0.00078	-0.00384
-3	0.87027	0.74928	-0.12542	22.136	-0.00399	-0.26502	0.018088	0.015573	-0.00261	0.32636	-0.00043	-0.00391
-2	0.58752	0.76202	-0.05502	22.245	-0.00117	-0.27017	0.012222	0.015852	-0.00114	0.32825	-0.00013	-0.00399
-1	0.2835	0.77144	-0.00622	22.359	0.000413	-0.27538	0.005902	0.016059	-0.00013	0.33017	4.45E-05	-0.00407
0	-0.00774	0.77389	0.002218	22.413	0.000157	-0.27784	-0.00016	0.016115	4.62E-05	0.33107	1.70E-05	-0.0041
1	-0.29942	0.77144	0.010699	22.359	-9.91E-05	-0.27538	-0.00623	0.016059	0.000223	0.33016	-1.07E-05	-0.00407
2	-0.60329	0.76192	0.05941	22.245	0.001482	-0.27016	-0.01255	0.01585	0.001236	0.32824	0.00016	-0.00399
3	-0.885	0.74922	0.12965	22.134	0.004302	-0.265	-0.01839	0.015572	0.002695	0.32634	0.000463	-0.00391
4	-1.1339	0.73944	0.20293	22.05	0.007519	-0.26085	-0.02355	0.015356	0.004214	0.3248	0.000808	-0.00384
5	-1.363	0.73397	0.2747	21.988	0.010807	-0.25767	-0.02828	0.015229	0.0057	0.32362	0.001161	-0.00379
6	-1.5837	0.73167	0.34507	21.943	0.014097	-0.25521	-0.03283	0.015169	0.007154	0.32269	0.001513	-0.00375
7	-1.801	0.7313	0.41466	21.907	0.017387	-0.25326	-0.03731	0.015151	0.008591	0.32195	0.001865	-0.00372
8	-2.016	0.73201	0.48387	21.878	0.020683	-0.25166	-0.04174	0.015156	0.010018	0.32131	0.002217	-0.0037
9	-2.2283	0.73329	0.55293	21.85	0.023988	-0.25029	-0.04611	0.015175	0.011442	0.32074	0.00257	-0.00367
10	-2.4372	0.73486	0.62198	21.823	0.0273	-0.24908	-0.05041	0.0152	0.012866	0.32021	0.002923	-0.00365
11	-2.6419	0.73654	0.69114	21.796	0.030621	-0.24798	-0.05463	0.01523	0.014291	0.31968	0.003278	-0.00364
12	-2.8418	0.73823	0.76051	21.766	0.033949	-0.24695	-0.05874	0.01526	0.01572	0.31914	0.003633	-0.00362
13	-3.0365	0.73987	0.83016	21.733	0.037285	-0.24594	-0.06275	0.01529	0.017156	0.31858	0.003989	-0.00361
14	-3.2256	0.7414	0.90014	21.696	0.040628	-0.24496	-0.06664	0.015318	0.018598	0.31798	0.004345	-0.00359
15	-3.4088	0.7428	0.97049	21.655	0.043981	-0.24396	-0.07042	0.015345	0.020048	0.31733	0.004703	-0.00357

Table 7-1 Control derivatives from stability test for SC0 (aileron deflection)

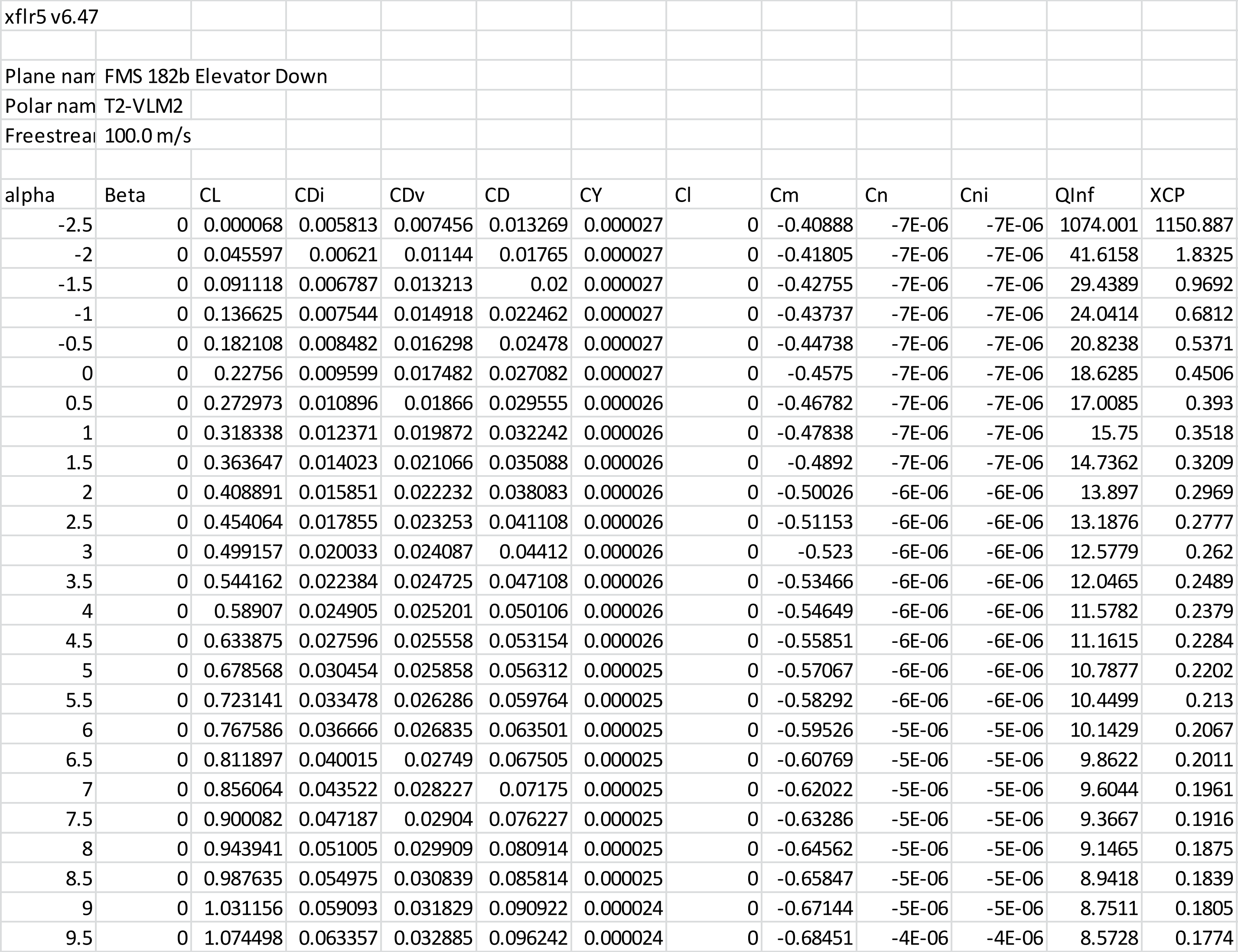
controlPo	Xde	Yde	Zde	Lde	Mde	Nde	CXde	CYde	CZde	CLde	CMde	CNde
-15	0.91245	-10.353	-0.55668	-0.06069	0.4211	6.8012	0.015765	-0.17887	-0.00962	-0.00074	0.037663	0.083356
-14	0.75082	-8.9131	-0.96989	-0.01254	0.065562	5.8419	0.014771	-0.17535	-0.01908	-0.00018	0.006677	0.081524
-13	0.65136	-8.062	-1.2549	0.016261	-0.17836	5.2743	0.013969	-0.1729	-0.02691	0.000247	-0.0198	0.080236
-12	0.58684	-7.5935	-1.44	0.032675	-0.34195	4.9595	0.013252	-0.17148	-0.03252	0.000523	-0.03997	0.079443
-11	0.54039	-7.3785	-1.5423	0.04073	-0.44456	4.811	0.012514	-0.17087	-0.03572	0.000669	-0.05329	0.079028
-10	0.50197	-7.3281	-1.5754	0.043031	-0.5003	4.7697	0.011699	-0.17079	-0.03672	0.000711	-0.06036	0.078855
-9	0.46588	-7.3792	-1.5509	0.041389	-0.52008	4.7944	0.010795	-0.17098	-0.03594	0.00068	-0.06238	0.078802
-8	0.42899	-7.4871	-1.4792	0.037123	-0.51232	4.8557	0.009812	-0.17124	-0.03383	0.000602	-0.06066	0.078779
-7	0.38956	-7.6204	-1.3685	0.031229	-0.48348	4.9336	0.008764	-0.17143	-0.03079	0.000498	-0.0563	0.078731
-6	0.34666	-7.7571	-1.2261	0.024479	-0.43845	5.0142	0.007663	-0.17148	-0.02711	0.000384	-0.05017	0.078628
-5	0.29984	-7.8823	-1.058	0.017494	-0.38103	5.0879	0.006518	-0.17136	-0.023	0.00027	-0.04288	0.07846
-4	0.24897	-7.9864	-0.86948	0.010805	-0.31428	5.1492	0.005333	-0.17107	-0.01862	0.000164	-0.03485	0.078238
-3	0.19395	-8.0638	-0.66533	0.004891	-0.24079	5.195	0.004104	-0.17065	-0.01408	7.34E-05	-0.02638	0.077985
-2	0.13421	-8.1131	-0.44971	0.000211	-0.16276	5.2251	0.002815	-0.17017	-0.00943	3.14E-06	-0.01767	0.077743
-1	0.068508	-8.1373	-0.22587	-0.00281	-0.08206	5.2415	0.001429	-0.16976	-0.00471	-4.16E-05	-0.00886	0.077565
0	-0.00265	-8.1437	0.002861	-0.00386	-0.00016	5.2465	-5.51E-05	-0.16958	5.96E-05	-5.70E-05	-1.75E-05	0.077499
1	-0.07355	-8.1371	0.23147	-0.00281	0.08173	5.2413	-0.00153	-0.16976	0.004829	-4.15E-05	0.008826	0.077563
2	-0.13882	-8.1123	0.45509	0.000223	0.16243	5.2247	-0.00291	-0.17016	0.009546	3.31E-06	0.017637	0.077739
3	-0.19826	-8.0624	0.67053	0.004907	0.24045	5.1943	-0.0042	-0.17063	0.014191	7.37E-05	0.026343	0.07798
4	-0.25306	-7.9845	0.87451	0.010824	0.31394	5.1483	-0.00542	-0.17104	0.018734	0.000164	0.034813	0.078231
5	-0.30374	-7.88	1.0629	0.017515	0.38067	5.0869	-0.0066	-0.17133	0.023109	0.00027	0.042843	0.078452
6	-0.35036	-7.7545	1.2308	0.0245	0.43808	5.013	-0.00775	-0.17144	0.027212	0.000384	0.050137	0.078618
7	-0.39305	-7.6175	1.373	0.031248	0.4831	4.9324	-0.00884	-0.17139	0.030892	0.000499	0.056266	0.078719
8	-0.43227	-7.4841	1.4835	0.037138	0.51194	4.8544	-0.00989	-0.17119	0.033933	0.000603	0.060618	0.078766
9	-0.46894	-7.3762	1.5551	0.041397	0.5197	4.793	-0.01087	-0.17093	0.036036	0.00068	0.062342	0.078787
10	-0.5048	-7.3251	1.5794	0.043031	0.49995	4.7685	-0.01177	-0.17073	0.036813	0.000711	0.060321	0.078839
11	-0.543	-7.3757	1.5464	0.040717	0.44424	4.8098	-0.01258	-0.1708	0.03581	0.000669	0.053254	0.07901
12	-0.58922	-7.5911	1.4442	0.032642	0.34167	4.9586	-0.01331	-0.17141	0.03261	0.000523	0.039938	0.079424
13	-0.65352	-8.0602	1.2593	0.016197	0.17813	5.2739	-0.01401	-0.17283	0.027002	0.000246	0.019772	0.080217
14	-0.75278	-8.9125	0.97468	-0.01266	-0.0658	5.8424	-0.01481	-0.17528	0.019169	-0.00018	-0.0067	0.081503
15	-0.91432	-10.354	0.56193	-0.06091	-0.42149	6.8035	-0.01579	-0.1788	0.009703	-0.00075	-0.03768	0.083334

Table 7-2 Control derivatives from stability test for SC0 (rudder deflection)



xflr5 v6.47																	
Plane nam	FMS 182b Elevator Up																
Polar nam	T2-VLM2																
Freestrea	100.0 m/s																
alpha	Beta	CL	CDi	CDv	CD	CY	Cl	Cm	Cn	Cni	QInf	XCP		alpha	Beta	CL	diff
2	0	0.039414	0.012247	0.011717	0.023964	3.90E-05	-1.00E-06	0.528309	-1.30E-05	-1.30E-05	44.7608	-2.5451		2	0	0.408891	0.369477
2.5	0	0.084811	0.012036	0.014541	0.026577	3.90E-05	-1.00E-06	0.51597	-1.30E-05	-1.30E-05	30.5139	-1.127		2.5	0	0.454064	0.369253
3	0	0.130195	0.012006	0.016746	0.028752	3.90E-05	-1.00E-06	0.50331	-1.30E-05	-1.30E-05	24.6279	-0.6978		3	0	0.499157	0.368962
3.5	0	0.175558	0.012154	0.018779	0.030933	3.80E-05	-1.00E-06	0.49016	-1.30E-05	-1.30E-05	21.2088	-0.4905		3.5	0	0.544162	0.368604
4	0	0.220891	0.012481	0.02058	0.033061	3.80E-05	-1.00E-06	0.476602	-1.30E-05	-1.30E-05	18.9076	-0.3683		4	0	0.58907	0.368179
4.5	0	0.266186	0.012987	0.022332	0.035319	3.80E-05	-1.00E-06	0.462579	-1.20E-05	-1.30E-05	17.224	-0.2878		4.5	0	0.633875	0.367689
5	0	0.311436	0.013671	0.023816	0.037487	3.80E-05	-1.00E-06	0.448216	-1.20E-05	-1.20E-05	15.9236	-0.2307		5	0	0.678568	0.367132
5.5	0	0.356631	0.014532	0.025016	0.039548	3.80E-05	-1.00E-06	0.433552	-1.20E-05	-1.20E-05	14.8804	-0.1881		5.5	0	0.723141	0.36651
6	0	0.401764	0.015568	0.026058	0.041627	3.80E-05	-1.00E-06	0.418616	-1.20E-05	-1.20E-05	14.0197	-0.1551		6	0	0.767586	0.365822
6.5	0	0.446828	0.01678	0.027083	0.043863	3.70E-05	-1.00E-06	0.40346	-1.20E-05	-1.20E-05	13.294	-0.1287		6.5	0	0.811897	0.365069
7	0	0.491813	0.018166	0.028078	0.046244	3.70E-05	-1.00E-06	0.388076	-1.20E-05	-1.20E-05	12.6714	-0.1073		7	0	0.856064	0.364251
7.5	0	0.536712	0.019724	0.029103	0.048828	3.70E-05	-1.00E-06	0.372431	-1.20E-05	-1.20E-05	12.1298	-0.0894		7.5	0	0.900082	0.36337
8	0	0.581517	0.021453	0.030098	0.051552	3.70E-05	-1.00E-06	0.356565	-1.10E-05	-1.10E-05	11.6532	-0.0742		8	0	0.943941	0.362424
8.5	0	0.62622	0.023352	0.030964	0.054316	3.70E-05	-2.00E-06	0.340533	-1.10E-05	-1.10E-05	11.2295	-0.0613		8.5	0	0.987635	0.361415
9	0	0.670813	0.025419	0.03183	0.057248	3.60E-05	-2.00E-06	0.324259	-1.10E-05	-1.10E-05	10.8499	-0.0501		9	0	1.031156	0.360343
9.5	0	0.715288	0.027651	0.032766	0.060417	3.60E-05	-2.00E-06	0.307774	-1.10E-05	-1.10E-05	10.5071	-0.0403		9.5	0	1.074498	0.35921
10	0	0.759638	0.030048	0.033738	0.063785	3.60E-05	-2.00E-06	0.291091	-1.10E-05	-1.10E-05	10.1958	-0.0316					

Table 7-3 Control derivatives from stability test for SC0 (Elevator up model)

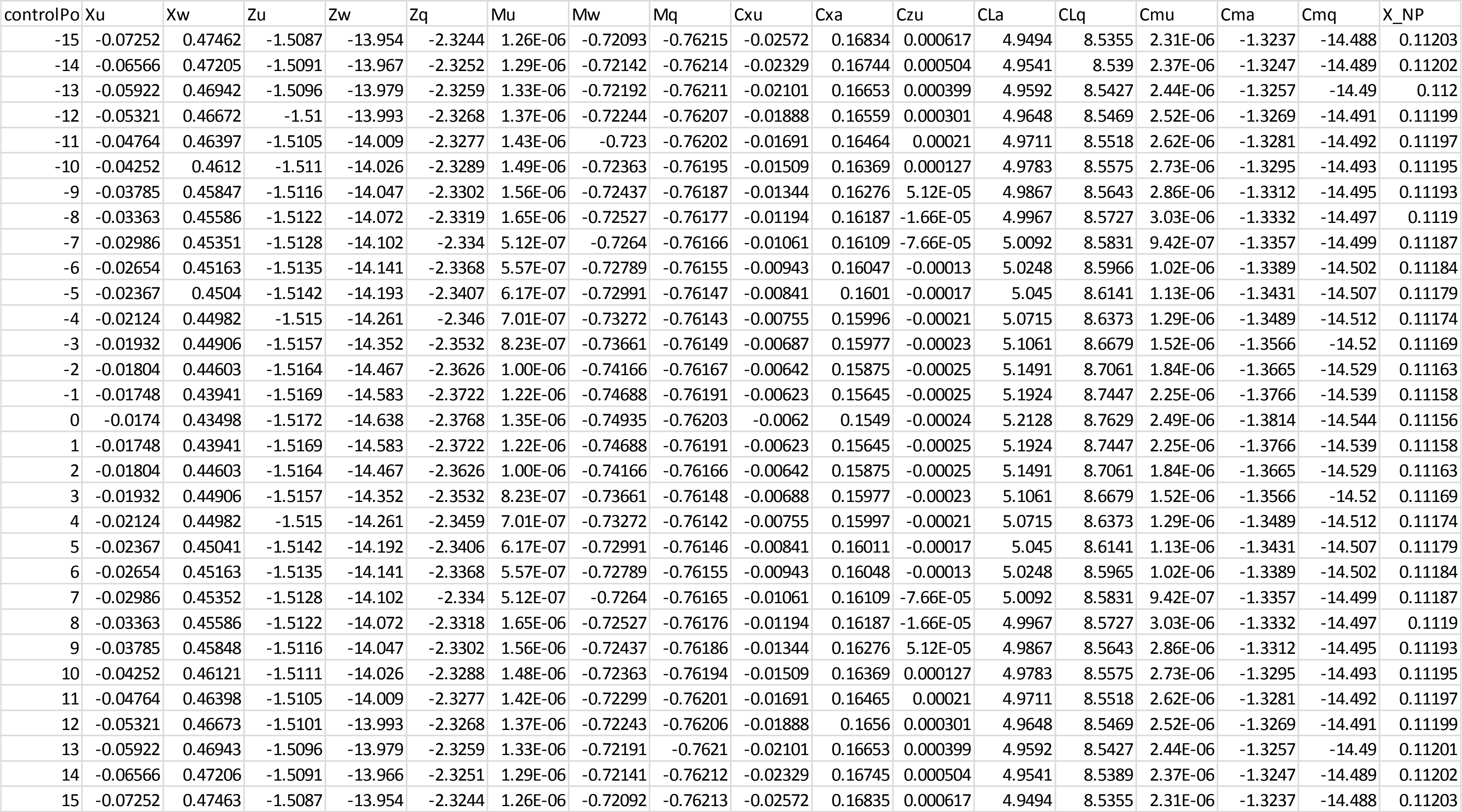


xflr5 v6.47												
Plane nam	FMS 182b Elevator Down											
Polar nam	T2-VLM2											
Freestreai	100.0 m/s											
alpha	Beta	CL	CDi	CDv	CD	CY	Cl	Cm	Cn	Cni	QInf	XCP
-2.5	0	0.000068	0.005813	0.007456	0.013269	0.000027	0	-0.40888	-7E-06	-7E-06	1074.001	1150.887
-2	0	0.045597	0.00621	0.01144	0.01765	0.000027	0	-0.41805	-7E-06	-7E-06	41.6158	1.8325
-1.5	0	0.091118	0.006787	0.013213	0.02	0.000027	0	-0.42755	-7E-06	-7E-06	29.4389	0.9692
-1	0	0.136625	0.007544	0.014918	0.022462	0.000027	0	-0.43737	-7E-06	-7E-06	24.0414	0.6812
-0.5	0	0.182108	0.008482	0.016298	0.02478	0.000027	0	-0.44738	-7E-06	-7E-06	20.8238	0.5371
0	0	0.22756	0.009599	0.017482	0.027082	0.000027	0	-0.4575	-7E-06	-7E-06	18.6285	0.4506
0.5	0	0.272973	0.010896	0.01866	0.029555	0.000026	0	-0.46782	-7E-06	-7E-06	17.0085	0.393
1	0	0.318338	0.012371	0.019872	0.032242	0.000026	0	-0.47838	-7E-06	-7E-06	15.75	0.3518
1.5	0	0.363647	0.014023	0.021066	0.035088	0.000026	0	-0.4892	-7E-06	-7E-06	14.7362	0.3209
2	0	0.408891	0.015851	0.022232	0.038083	0.000026	0	-0.50026	-6E-06	-6E-06	13.897	0.2969
2.5	0	0.454064	0.017855	0.023253	0.041108	0.000026	0	-0.51153	-6E-06	-6E-06	13.1876	0.2777
3	0	0.499157	0.020033	0.024087	0.04412	0.000026	0	-0.523	-6E-06	-6E-06	12.5779	0.262
3.5	0	0.544162	0.022384	0.024725	0.047108	0.000026	0	-0.53466	-6E-06	-6E-06	12.0465	0.2489
4	0	0.58907	0.024905	0.025201	0.050106	0.000026	0	-0.54649	-6E-06	-6E-06	11.5782	0.2379
4.5	0	0.633875	0.027596	0.025558	0.053154	0.000026	0	-0.55851	-6E-06	-6E-06	11.1615	0.2284
5	0	0.678568	0.030454	0.025858	0.056312	0.000025	0	-0.57067	-6E-06	-6E-06	10.7877	0.2202
5.5	0	0.723141	0.033478	0.026286	0.059764	0.000025	0	-0.58292	-6E-06	-6E-06	10.4499	0.213
6	0	0.767586	0.036666	0.026835	0.063501	0.000025	0	-0.59526	-5E-06	-5E-06	10.1429	0.2067
6.5	0	0.811897	0.040015	0.02749	0.067505	0.000025	0	-0.60769	-5E-06	-5E-06	9.8622	0.2011
7	0	0.856064	0.043522	0.028227	0.07175	0.000025	0	-0.62022	-5E-06	-5E-06	9.6044	0.1961
7.5	0	0.900082	0.047187	0.02904	0.076227	0.000025	0	-0.63286	-5E-06	-5E-06	9.3667	0.1916
8	0	0.943941	0.051005	0.029909	0.080914	0.000025	0	-0.64562	-5E-06	-5E-06	9.1465	0.1875
8.5	0	0.987635	0.054975	0.030839	0.085814	0.000025	0	-0.65847	-5E-06	-5E-06	8.9418	0.1839
9	0	1.031156	0.059093	0.031829	0.090922	0.000024	0	-0.67144	-5E-06	-5E-06	8.7511	0.1805
9.5	0	1.074498	0.063357	0.032885	0.096242	0.000024	0	-0.68451	-4E-06	-4E-06	8.5728	0.1774

Table 7-4 Control derivatives from stability test for SC0 (Elevator down model)

controlPo	Yv	Yp	Yr	Lv	Lp	Lr	Nv	Np	Nr	CYb	CYp	CYr	Clb	Clp	Clr	Cnb	Cnp	Cnr
-15	-0.688	-0.11304	0.47034	0.00152	-1.3653	0.18503	0.42204	-0.06816	-0.29142	-0.24402	-0.05688	0.23667	0.000382	-0.48731	0.066042	0.10618	-0.02433	-0.10402
-14	-0.68782	-0.11313	0.4705	0.001433	-1.3656	0.18493	0.42206	-0.06803	-0.2914	-0.24398	-0.05693	0.23677	0.000361	-0.48748	0.066014	0.1062	-0.02428	-0.10402
-13	-0.68764	-0.11323	0.47064	0.001344	-1.366	0.18484	0.42206	-0.06789	-0.29138	-0.24394	-0.05699	0.23686	0.000338	-0.48766	0.065987	0.10621	-0.02424	-0.10402
-12	-0.68745	-0.11334	0.47075	0.00125	-1.3664	0.18474	0.42205	-0.06775	-0.29135	-0.24391	-0.05705	0.23695	0.000315	-0.48786	0.06596	0.10622	-0.02419	-0.10403
-11	-0.68725	-0.11348	0.47085	0.001152	-1.3669	0.18464	0.42204	-0.06761	-0.29131	-0.24387	-0.05713	0.23704	0.00029	-0.48811	0.065936	0.10623	-0.02414	-0.10403
-10	-0.68704	-0.11364	0.47092	0.001047	-1.3675	0.18454	0.422	-0.06745	-0.29127	-0.24384	-0.05722	0.23712	0.000264	-0.48842	0.065913	0.10624	-0.02409	-0.10403
-9	-0.68682	-0.11383	0.47097	0.000934	-1.3683	0.18445	0.42196	-0.06729	-0.29122	-0.24382	-0.05733	0.2372	0.000235	-0.48881	0.065893	0.10626	-0.02404	-0.10404
-8	-0.68659	-0.11409	0.47099	0.000808	-1.3693	0.18435	0.42189	-0.06711	-0.29117	-0.2438	-0.05747	0.23727	0.000204	-0.48932	0.065876	0.10627	-0.02398	-0.10405
-7	-0.68633	-0.11441	0.47099	0.000666	-1.3708	0.18426	0.42182	-0.0669	-0.2911	-0.24379	-0.05765	0.23734	0.000168	-0.49001	0.065865	0.10628	-0.02392	-0.10405
-6	-0.68607	-0.11484	0.47095	0.000499	-1.3729	0.18417	0.42173	-0.06667	-0.29103	-0.24378	-0.05789	0.2374	0.000126	-0.49092	0.065856	0.1063	-0.02384	-0.10407
-5	-0.68579	-0.1154	0.47089	0.000297	-1.3759	0.18409	0.42163	-0.06638	-0.29096	-0.24378	-0.0582	0.23747	7.48E-05	-0.49217	0.065852	0.10631	-0.02375	-0.10408
-4	-0.68552	-0.11614	0.47081	4.23E-05	-1.38	0.184	0.42154	-0.06603	-0.29089	-0.24379	-0.05859	0.23753	1.07E-05	-0.49388	0.065849	0.10634	-0.02363	-0.1041
-3	-0.68528	-0.11703	0.47071	-0.00028	-1.3859	0.18391	0.42146	-0.06557	-0.29083	-0.24381	-0.05907	0.23759	-7.08E-05	-0.49619	0.065845	0.10636	-0.02348	-0.10413
-2	-0.68507	-0.11799	0.47062	-0.00067	-1.3934	0.18381	0.42142	-0.06502	-0.2908	-0.24384	-0.05958	0.23764	-0.00017	-0.49909	0.065837	0.1064	-0.02329	-0.10416
-1	-0.68491	-0.11876	0.47052	-0.00106	-1.4012	0.18371	0.42141	-0.06446	-0.29079	-0.24387	-0.05999	0.23767	-0.00027	-0.50206	0.065828	0.10643	-0.0231	-0.10419
0	-0.68484	-0.11905	0.47047	-0.00124	-1.4049	0.18367	0.4214	-0.0642	-0.29079	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10645	-0.02301	-0.10421
1	-0.68491	-0.11876	0.47052	-0.00106	-1.4012	0.18372	0.42141	-0.06446	-0.29079	-0.24387	-0.05999	0.23767	-0.00027	-0.50206	0.065828	0.10643	-0.0231	-0.10419
2	-0.68507	-0.11799	0.47061	-0.00067	-1.3934	0.18381	0.42142	-0.06502	-0.2908	-0.24383	-0.05958	0.23764	-0.00017	-0.49909	0.065838	0.1064	-0.02329	-0.10416
3	-0.68527	-0.11703	0.47071	-0.00028	-1.3859	0.18391	0.42146	-0.06557	-0.29083	-0.24381	-0.05907	0.23759	-7.07E-05	-0.49619	0.065846	0.10636	-0.02348	-0.10413
4	-0.68552	-0.11613	0.47081	4.28E-05	-1.38	0.184	0.42153	-0.06603	-0.29088	-0.24379	-0.05859	0.23753	1.08E-05	-0.49388	0.06585	0.10634	-0.02363	-0.1041
5	-0.68579	-0.1154	0.47089	0.000297	-1.3759	0.18409	0.42163	-0.06639	-0.29095	-0.24378	-0.0582	0.23747	7.49E-05	-0.49217	0.065853	0.10631	-0.02375	-0.10408
6	-0.68606	-0.11484	0.47095	0.0005	-1.3729	0.18418	0.42172	-0.06667	-0.29103	-0.24378	-0.05789	0.2374	0.000126	-0.49092	0.065858	0.1063	-0.02384	-0.10407
7	-0.68633	-0.11441	0.47098	0.000667	-1.3708	0.18427	0.42181	-0.06691	-0.2911	-0.24379	-0.05765	0.23734	0.000168	-0.49001	0.065867	0.10628	-0.02392	-0.10405
8	-0.68658	-0.11409	0.47099	0.000809	-1.3693	0.18436	0.42189	-0.06711	-0.29116	-0.2438	-0.05747	0.23727	0.000204	-0.48932	0.065879	0.10627	-0.02398	-0.10405
9	-0.68681	-0.11383	0.47097	0.000935	-1.3682	0.18445	0.42195	-0.06729	-0.29122	-0.24382	-0.05733	0.2372	0.000235	-0.48881	0.065896	0.10625	-0.02404	-0.10404
10	-0.68703	-0.11363	0.47092	0.001048	-1.3674	0.18455	0.422	-0.06746	-0.29127	-0.24384	-0.05722	0.23712	0.000264	-0.48842	0.065916	0.10624	-0.02409	-0.10403
11	-0.68724	-0.11347	0.47084	0.001153	-1.3668	0.18465	0.42203	-0.06761	-0.29131	-0.24387	-0.05713	0.23704	0.00029	-0.48811	0.065939	0.10623	-0.02415	-0.10403
12	-0.68744	-0.11334	0.47075	0.001252	-1.3664	0.18475	0.42205	-0.06776	-0.29134	-0.2439	-0.05705	0.23695	0.000315	-0.48786	0.065964	0.10622	-0.02419	-0.10403
13	-0.68763	-0.11323	0.47063	0.001345	-1.366	0.18484	0.42205	-0.0679	-0.29137	-0.24394	-0.05699	0.23686	0.000339	-0.48766	0.065991	0.10621	-0.02424	-0.10402
14	-0.68781	-0.11313	0.47049	0.001435	-1.3656	0.18494	0.42205	-0.06803	-0.29139	-0.24398	-0.05693	0.23677	0.000361	-0.48748	0.066018	0.10619	-0.02429	-0.10402
15	-0.68798	-0.11304	0.47033	0.001522	-1.3652	0.18504	0.42203	-0.06816	-0.29141	-0.24402	-0.05688	0.23667	0.000383	-0.48731	0.066047	0.10618	-0.02433	-0.10402

Table 7-5 Elevator deflection related stability/control dataset



controlPo	Xu	Xw	Zu	Zw	Zq	Mu	Mw	Mq	Cxu	Cxa	Czu	CLa	CLq	Cmu	Cma	Cmq	X_NP
-15	-0.07252	0.47462	-1.5087	-13.954	-2.3244	1.26E-06	-0.72093	-0.76215	-0.02572	0.16834	0.000617	4.9494	8.5355	2.31E-06	-1.3237	-14.488	0.11203
-14	-0.06566	0.47205	-1.5091	-13.967	-2.3252	1.29E-06	-0.72142	-0.76214	-0.02329	0.16744	0.000504	4.9541	8.539	2.37E-06	-1.3247	-14.489	0.11202
-13	-0.05922	0.46942	-1.5096	-13.979	-2.3259	1.33E-06	-0.72192	-0.76211	-0.02101	0.16653	0.000399	4.9592	8.5427	2.44E-06	-1.3257	-14.49	0.112
-12	-0.05321	0.46672	-1.51	-13.993	-2.3268	1.37E-06	-0.72244	-0.76207	-0.01888	0.16559	0.000301	4.9648	8.5469	2.52E-06	-1.3269	-14.491	0.11199
-11	-0.04764	0.46397	-1.5105	-14.009	-2.3277	1.43E-06	-0.723	-0.76202	-0.01691	0.16464	0.00021	4.9711	8.5518	2.62E-06	-1.3281	-14.492	0.11197
-10	-0.04252	0.4612	-1.511	-14.026	-2.3289	1.49E-06	-0.72363	-0.76195	-0.01509	0.16369	0.000127	4.9783	8.5575	2.73E-06	-1.3295	-14.493	0.11195
-9	-0.03785	0.45847	-1.5116	-14.047	-2.3302	1.56E-06	-0.72437	-0.76187	-0.01344	0.16276	5.12E-05	4.9867	8.5643	2.86E-06	-1.3312	-14.495	0.11193
-8	-0.03363	0.45586	-1.5122	-14.072	-2.3319	1.65E-06	-0.72527	-0.76177	-0.01194	0.16187	-1.66E-05	4.9967	8.5727	3.03E-06	-1.3332	-14.497	0.1119
-7	-0.02986	0.45351	-1.5128	-14.102	-2.334	5.12E-07	-0.7264	-0.76166	-0.01061	0.16109	-7.66E-05	5.0092	8.5831	9.42E-07	-1.3357	-14.499	0.11187
-6	-0.02654	0.45163	-1.5135	-14.141	-2.3368	5.57E-07	-0.72789	-0.76155	-0.00943	0.16047	-0.00013	5.0248	8.5966	1.02E-06	-1.3389	-14.502	0.11184
-5	-0.02367	0.4504	-1.5142	-14.193	-2.3407	6.17E-07	-0.72991	-0.76147	-0.00841	0.1601	-0.00017	5.045	8.6141	1.13E-06	-1.3431	-14.507	0.11179
-4	-0.02124	0.44982	-1.515	-14.261	-2.346	7.01E-07	-0.73272	-0.76143	-0.00755	0.15996	-0.00021	5.0715	8.6373	1.29E-06	-1.3489	-14.512	0.11174
-3	-0.01932	0.44906	-1.5157	-14.352	-2.3532	8.23E-07	-0.73661	-0.76149	-0.00687	0.15977	-0.00023	5.1061	8.6679	1.52E-06	-1.3566	-14.52	0.11169
-2	-0.01804	0.44603	-1.5164	-14.467	-2.3626	1.00E-06	-0.74166	-0.76167	-0.00642	0.15875	-0.00025	5.1491	8.7061	1.84E-06	-1.3665	-14.529	0.11163
-1	-0.01748	0.43941	-1.5169	-14.583	-2.3722	1.22E-06	-0.74688	-0.76191	-0.00623	0.15645	-0.00025	5.1924	8.7447	2.25E-06	-1.3766	-14.539	0.11158
0	-0.0174	0.43498	-1.5172	-14.638	-2.3768	1.35E-06	-0.74935	-0.76203	-0.0062	0.1549	-0.00024	5.2128	8.7629	2.49E-06	-1.3814	-14.544	0.11156
1	-0.01748	0.43941	-1.5169	-14.583	-2.3722	1.22E-06	-0.74688	-0.76191	-0.00623	0.15645	-0.00025	5.1924	8.7447	2.25E-06	-1.3766	-14.539	0.11158
2	-0.01804	0.44603	-1.5164	-14.467	-2.3626	1.00E-06	-0.74166	-0.76166	-0.00642	0.15875	-0.00025	5.1491	8.7061	1.84E-06	-1.3665	-14.529	0.11163
3	-0.01932	0.44906	-1.5157	-14.352	-2.3532	8.23E-07	-0.73661	-0.76148	-0.00688	0.15977	-0.00023	5.1061	8.6679	1.52E-06	-1.3566	-14.52	0.11169
4	-0.02124	0.44982	-1.515	-14.261	-2.3459	7.01E-07	-0.73272	-0.76142	-0.00755	0.15997	-0.00021	5.0715	8.6373	1.29E-06	-1.3489	-14.512	0.11174
5	-0.02367	0.45041	-1.5142	-14.192	-2.3406	6.17E-07	-0.72991	-0.76146	-0.00841	0.16011	-0.00017	5.045	8.6141	1.13E-06	-1.3431	-14.507	0.11179
6	-0.02654	0.45163	-1.5135	-14.141	-2.3368	5.57E-07	-0.72789	-0.76155	-0.00943	0.16048	-0.00013	5.0248	8.5965	1.02E-06	-1.3389	-14.502	0.11184
7	-0.02986	0.45352	-1.5128	-14.102	-2.334	5.12E-07	-0.7264	-0.76165	-0.01061	0.16109	-7.66E-05	5.0092	8.5831	9.42E-07	-1.3357	-14.499	0.11187
8	-0.03363	0.45586	-1.5122	-14.072	-2.3318	1.65E-06	-0.72527	-0.76176	-0.01194	0.16187	-1.66E-05	4.9967	8.5727	3.03E-06	-1.3332	-14.497	0.1119
9	-0.03785	0.45848	-1.5116	-14.047	-2.3302	1.56E-06	-0.72437	-0.76186	-0.01344	0.16276	5.12E-05	4.9867	8.5643	2.86E-06	-1.3312	-14.495	0.11193
10	-0.04252	0.46121	-1.5111	-14.026	-2.3288	1.48E-06	-0.72363	-0.76194	-0.01509	0.16369	0.000127	4.9783	8.5575	2.73E-06	-1.3295	-14.493	0.11195
11	-0.04764	0.46398	-1.5105	-14.009	-2.3277	1.42E-06	-0.72299	-0.76201	-0.01691	0.16465	0.00021	4.9711	8.5518	2.62E-06	-1.3281	-14.492	0.11197
12	-0.05321	0.46673	-1.5101	-13.993	-2.3268	1.37E-06	-0.72243	-0.76206	-0.01888	0.1656	0.000301	4.9648	8.5469	2.52E-06	-1.3269	-14.491	0.11199
13	-0.05922	0.46943	-1.5096	-13.979	-2.3259	1.33E-06	-0.72191	-0.7621	-0.02101	0.16653	0.000399	4.9592	8.5427	2.44E-06	-1.3257	-14.49	0.11201
14	-0.06566	0.47206	-1.5091	-13.966	-2.3251	1.29E-06	-0.72141	-0.76212	-0.02329	0.16745	0.000504	4.9541	8.5389	2.37E-06	-1.3247	-14.489	0.11202
15	-0.07252	0.47463	-1.5087	-13.954	-2.3244	1.26E-06	-0.72092	-0.76213	-0.02572	0.16835	0.000617	4.9494	8.5355	2.31E-06	-1.3237	-14.488	0.11203

Table 7-6 Elevator deflection related stability/control dataset

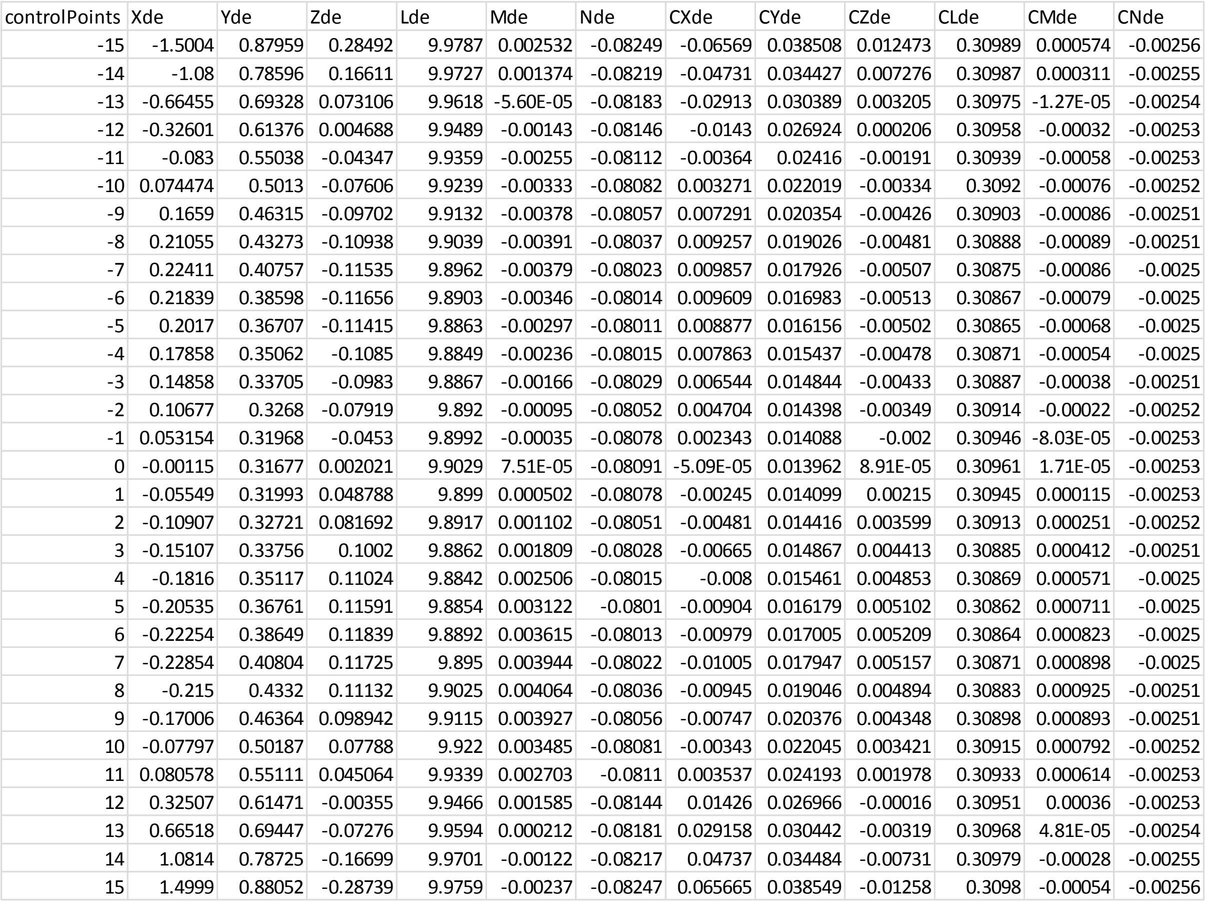
controlPo	Yv	Yp	Yr	Lv	Lp	Lr	Nv	Np	Nr	CYb	CYp	CYr	Clb	Clp	Clr	Cnb	Cnp	Cnr
-15	-0.79357	-0.11869	0.54739	-0.00683	-1.543	0.17344	0.49427	-0.0568	-0.34204	-0.25742	-0.05462	0.25191	-0.00157	-0.50369	0.056618	0.11373	-0.01854	-0.11166
-14	-0.73078	-0.11222	0.50506	-0.0024	-1.4454	0.18128	0.45408	-0.0671	-0.31469	-0.25295	-0.05511	0.24801	-0.00059	-0.50346	0.063146	0.11149	-0.02337	-0.10962
-13	-0.6914	-0.10859	0.47846	0.00056	-1.3838	0.18687	0.42883	-0.07381	-0.29746	-0.24987	-0.05568	0.24531	0.000144	-0.50329	0.067964	0.10993	-0.02684	-0.10818
-12	-0.66896	-0.10696	0.46318	0.002375	-1.3483	0.1903	0.41433	-0.07756	-0.28749	-0.24808	-0.05627	0.24368	0.000625	-0.50317	0.071019	0.10899	-0.02894	-0.10729
-11	-0.65848	-0.10675	0.45584	0.003301	-1.3313	0.1919	0.40737	-0.079	-0.28262	-0.24728	-0.05687	0.24286	0.000879	-0.50311	0.072524	0.10852	-0.02985	-0.10681
-10	-0.65592	-0.10747	0.45377	0.003572	-1.327	0.19216	0.40537	-0.07878	-0.28109	-0.24711	-0.05744	0.24253	0.000955	-0.5031	0.072853	0.10833	-0.02987	-0.10657
-9	-0.65816	-0.1088	0.45488	0.00339	-1.3309	0.19153	0.40638	-0.07748	-0.28159	-0.24724	-0.05798	0.24242	0.000903	-0.50312	0.072406	0.10828	-0.02929	-0.10645
-8	-0.66291	-0.11044	0.45765	0.002917	-1.3397	0.19042	0.40894	-0.07558	-0.28314	-0.24741	-0.05847	0.24232	0.000772	-0.50316	0.07152	0.10826	-0.02839	-0.10634
-7	-0.66856	-0.1122	0.46104	0.002278	-1.3509	0.18911	0.41208	-0.07341	-0.28509	-0.24746	-0.05892	0.2421	0.000598	-0.50321	0.070442	0.10819	-0.02734	-0.10619
-6	-0.67402	-0.11392	0.46432	0.001565	-1.363	0.18778	0.41514	-0.07122	-0.287	-0.24731	-0.0593	0.2417	0.000407	-0.50327	0.069338	0.10805	-0.0263	-0.10597
-5	-0.67861	-0.11549	0.46707	0.000847	-1.3746	0.18656	0.41771	-0.06918	-0.2886	-0.24692	-0.05962	0.2411	0.000219	-0.50332	0.068314	0.10781	-0.02533	-0.10568
-4	-0.68195	-0.11681	0.46903	0.000176	-1.3849	0.18553	0.41958	-0.06742	-0.28975	-0.2463	-0.05986	0.24033	4.51E-05	-0.50337	0.067433	0.1075	-0.02451	-0.10532
-3	-0.68395	-0.11784	0.47015	-0.0004	-1.3934	0.18471	0.42072	-0.06602	-0.29043	-0.24554	-0.06002	0.23945	-0.0001	-0.50341	0.066731	0.10714	-0.02385	-0.10493
-2	-0.68479	-0.11854	0.47055	-0.00086	-1.3997	0.18413	0.42125	-0.06501	-0.29073	-0.24475	-0.06011	0.23859	-0.00022	-0.50344	0.066226	0.1068	-0.02338	-0.10457
-1	-0.6849	-0.11893	0.47054	-0.00114	-1.4036	0.18379	0.42139	-0.0644	-0.29079	-0.24412	-0.06014	0.23794	-0.00029	-0.50346	0.065924	0.10654	-0.0231	-0.10431
0	-0.68484	-0.11905	0.47047	-0.00124	-1.4049	0.18367	0.4214	-0.0642	-0.29079	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10645	-0.02301	-0.10421
1	-0.68489	-0.11893	0.47053	-0.00114	-1.4036	0.18379	0.42139	-0.06441	-0.29079	-0.24412	-0.06014	0.23794	-0.00029	-0.50346	0.065925	0.10654	-0.0231	-0.10431
2	-0.68477	-0.11854	0.47054	-0.00085	-1.3997	0.18413	0.42124	-0.06501	-0.29072	-0.24475	-0.06011	0.23859	-0.00022	-0.50344	0.066229	0.1068	-0.02338	-0.10457
3	-0.68393	-0.11784	0.47013	-0.0004	-1.3934	0.18471	0.42071	-0.06602	-0.29042	-0.24554	-0.06002	0.23945	-0.0001	-0.50341	0.066735	0.10714	-0.02385	-0.10493
4	-0.68192	-0.11681	0.46901	0.000179	-1.3848	0.18553	0.41956	-0.06743	-0.28974	-0.2463	-0.05986	0.24033	4.60E-05	-0.50337	0.067438	0.1075	-0.02451	-0.10532
5	-0.67857	-0.11549	0.46704	0.000851	-1.3745	0.18657	0.41769	-0.06919	-0.28859	-0.24692	-0.05962	0.2411	0.00022	-0.50332	0.06832	0.10781	-0.02534	-0.10568
6	-0.67398	-0.11392	0.4643	0.00157	-1.3629	0.18779	0.41511	-0.07123	-0.28699	-0.24731	-0.0593	0.2417	0.000409	-0.50327	0.069345	0.10805	-0.0263	-0.10597
7	-0.66851	-0.11219	0.46101	0.002283	-1.3509	0.18912	0.41206	-0.07342	-0.28508	-0.24746	-0.05892	0.2421	0.000599	-0.50321	0.070449	0.10819	-0.02735	-0.10619
8	-0.66287	-0.11043	0.45762	0.002922	-1.3396	0.19043	0.40892	-0.07559	-0.28313	-0.2474	-0.05848	0.24231	0.000774	-0.50316	0.071527	0.10826	-0.02839	-0.10634
9	-0.65812	-0.10879	0.45485	0.003394	-1.3308	0.19154	0.40636	-0.07749	-0.28157	-0.24723	-0.05798	0.24242	0.000904	-0.50312	0.072412	0.10828	-0.0293	-0.10645
10	-0.65589	-0.10747	0.45375	0.003576	-1.3269	0.19216	0.40536	-0.07878	-0.28108	-0.24711	-0.05744	0.24253	0.000956	-0.5031	0.072858	0.10833	-0.02987	-0.10657
11	-0.65846	-0.10675	0.45584	0.003304	-1.3313	0.19191	0.40737	-0.079	-0.28261	-0.24728	-0.05687	0.24286	0.00088	-0.50311	0.072526	0.10852	-0.02986	-0.10681
12	-0.66898	-0.10697	0.46319	0.002376	-1.3483	0.1903	0.41434	-0.07755	-0.2875	-0.24808	-0.05627	0.24368	0.000625	-0.50317	0.071017	0.10899	-0.02894	-0.10729
13	-0.69146	-0.1086	0.4785	0.000558	-1.384	0.18687	0.42887	-0.0738	-0.29749	-0.24987	-0.05567	0.24531	0.000143	-0.50329	0.067955	0.10993	-0.02684	-0.10818
14	-0.73091	-0.11224	0.50515	-0.00241	-1.4456	0.18126	0.45417	-0.06707	-0.31475	-0.25295	-0.05511	0.24801	-0.00059	-0.50346	0.063128	0.11149	-0.02336	-0.10962
15	-0.79382	-0.11872	0.54756	-0.00684	-1.5435	0.17341	0.49444	-0.05676	-0.34215	-0.25742	-0.05462	0.25191	-0.00157	-0.50369	0.05659	0.11374	-0.01852	-0.11166

Table 7-7 Elevator deflection related stability/control dataset

controlPo	Xu	Xw	Zu	Zw	Zq	Mu	Mw	Mq	Cxu	Cxa	Czu	CLa	CLq	Cmu	Cma	Cmq	X_NP
-15	-0.03013	0.40083	-1.3811	-16.084	-2.6213	1.02E-06	-0.8034	-0.8366	-0.00977	0.13002	6.96E-05	5.2173	8.8031	1.71E-06	-1.349	-14.544	0.11031
-14	-0.02937	0.42627	-1.474	-15.065	-2.4498	6.45E-07	-0.76567	-0.78385	-0.01017	0.14755	-1.34E-05	5.2145	8.7792	1.16E-06	-1.3719	-14.541	0.11119
-13	-0.02913	0.44416	-1.5393	-14.422	-2.3417	9.99E-07	-0.74241	-0.75064	-0.01053	0.16052	-0.0001	5.2121	8.7615	1.87E-06	-1.3889	-14.539	0.11184
-12	-0.02879	0.45517	-1.5797	-14.051	-2.2791	6.58E-07	-0.72915	-0.73146	-0.01068	0.1688	-0.00018	5.2105	8.7501	1.26E-06	-1.3997	-14.538	0.11226
-11	-0.0281	0.4605	-1.5998	-13.873	-2.249	5.41E-07	-0.72283	-0.7223	-0.01055	0.17293	-0.00023	5.2097	8.7442	1.05E-06	-1.4052	-14.537	0.11247
-10	-0.02704	0.46163	-1.605	-13.828	-2.2413	5.16E-07	-0.72117	-0.72001	-0.01019	0.17392	-0.00026	5.2095	8.7423	1.01E-06	-1.4065	-14.538	0.11252
-9	-0.02574	0.46	-1.6004	-13.868	-2.2481	5.43E-07	-0.72252	-0.72215	-0.00967	0.1728	-0.00028	5.2096	8.7431	1.06E-06	-1.405	-14.539	0.11246
-8	-0.02432	0.45677	-1.59	-13.96	-2.2634	6.04E-07	-0.72565	-0.72691	-0.00908	0.17047	-0.00028	5.21	8.7455	1.17E-06	-1.4019	-14.54	0.11234
-7	-0.02289	0.4528	-1.577	-14.077	-2.283	6.92E-07	-0.7297	-0.73299	-0.00847	0.1676	-0.00027	5.2106	8.7486	1.33E-06	-1.3981	-14.541	0.1122
-6	-0.02155	0.44867	-1.5632	-14.202	-2.3039	8.01E-07	-0.73405	-0.73948	-0.00791	0.16463	-0.00027	5.2111	8.7519	1.52E-06	-1.3942	-14.541	0.11205
-5	-0.02035	0.44479	-1.5502	-14.323	-2.3241	9.23E-07	-0.73826	-0.74573	-0.0074	0.16184	-0.00026	5.2116	8.755	1.74E-06	-1.3905	-14.542	0.11191
-4	-0.01932	0.44138	-1.5387	-14.431	-2.3421	1.05E-06	-0.74204	-0.7513	-0.00698	0.15942	-0.00026	5.212	8.7578	1.96E-06	-1.3874	-14.543	0.11178
-3	-0.0185	0.43863	-1.5295	-14.519	-2.3569	1.17E-06	-0.74515	-0.75588	-0.00664	0.15747	-0.00025	5.2123	8.76	2.17E-06	-1.3848	-14.543	0.11169
-2	-0.0179	0.43661	-1.5227	-14.585	-2.3678	1.26E-06	-0.74746	-0.75926	-0.0064	0.15605	-0.00025	5.2126	8.7616	2.34E-06	-1.3829	-14.543	0.11161
-1	-0.01753	0.43539	-1.5185	-14.625	-2.3745	1.33E-06	-0.74888	-0.76133	-0.00625	0.15519	-0.00025	5.2127	8.7625	2.45E-06	-1.3818	-14.544	0.11157
0	-0.0174	0.43498	-1.5172	-14.638	-2.3768	1.35E-06	-0.74935	-0.76203	-0.0062	0.1549	-0.00024	5.2128	8.7629	2.49E-06	-1.3814	-14.544	0.11156
1	-0.01753	0.4354	-1.5186	-14.624	-2.3745	1.33E-06	-0.74887	-0.76133	-0.00625	0.15519	-0.00025	5.2127	8.7625	2.45E-06	-1.3818	-14.544	0.11157
2	-0.0179	0.43663	-1.5227	-14.584	-2.3678	1.26E-06	-0.74745	-0.75925	-0.0064	0.15606	-0.00025	5.2126	8.7616	2.34E-06	-1.3829	-14.543	0.11161
3	-0.01851	0.43865	-1.5295	-14.519	-2.3568	1.17E-06	-0.74514	-0.75585	-0.00664	0.15748	-0.00025	5.2123	8.76	2.17E-06	-1.3848	-14.543	0.11169
4	-0.01933	0.44141	-1.5388	-14.43	-2.342	1.05E-06	-0.74202	-0.75127	-0.00698	0.15943	-0.00026	5.212	8.7577	1.96E-06	-1.3874	-14.543	0.11179
5	-0.02036	0.44481	-1.5503	-14.322	-2.324	9.23E-07	-0.73824	-0.74569	-0.00741	0.16186	-0.00026	5.2116	8.755	1.74E-06	-1.3906	-14.542	0.11191
6	-0.02157	0.4487	-1.5633	-14.202	-2.3038	8.00E-07	-0.73402	-0.73944	-0.00791	0.16464	-0.00027	5.2111	8.7519	1.52E-06	-1.3943	-14.541	0.11205
7	-0.02291	0.45283	-1.577	-14.076	-2.2828	6.91E-07	-0.72967	-0.73295	-0.00848	0.16762	-0.00027	5.2106	8.7486	1.32E-06	-1.3982	-14.541	0.1122
8	-0.02434	0.4568	-1.5901	-13.959	-2.2632	6.03E-07	-0.72563	-0.72687	-0.00908	0.17049	-0.00028	5.21	8.7455	1.17E-06	-1.402	-14.54	0.11235
9	-0.02576	0.46003	-1.6005	-13.868	-2.248	5.42E-07	-0.7225	-0.72212	-0.00968	0.17282	-0.00028	5.2096	8.7431	1.05E-06	-1.405	-14.539	0.11246
10	-0.02706	0.46165	-1.6051	-13.827	-2.2413	5.16E-07	-0.72116	-0.71999	-0.0102	0.17393	-0.00026	5.2094	8.7423	1.01E-06	-1.4065	-14.538	0.11252
11	-0.02812	0.46051	-1.5998	-13.873	-2.249	5.41E-07	-0.72283	-0.72229	-0.01056	0.17294	-0.00023	5.2097	8.7442	1.05E-06	-1.4052	-14.537	0.11247
12	-0.02881	0.45517	-1.5796	-14.051	-2.2791	6.58E-07	-0.72917	-0.73148	-0.01068	0.16879	-0.00018	5.2105	8.7502	1.26E-06	-1.3997	-14.538	0.11226
13	-0.02914	0.44413	-1.5391	-14.424	-2.3419	1.00E-06	-0.74245	-0.7507	-0.01053	0.16049	-0.0001	5.2122	8.7616	1.87E-06	-1.3889	-14.539	0.11184
14	-0.02939	0.42621	-1.4738	-15.067	-2.4503	6.47E-07	-0.76577	-0.78399	-0.01017	0.1475	-1.29E-05	5.2145	8.7792	1.16E-06	-1.3719	-14.541	0.11119
15	-0.03016	0.40073	-1.3807	-16.089	-2.6221	1.03E-06	-0.80358	-0.83685	-0.00978	0.12995	7.01E-05	5.2173	8.8032	1.72E-06	-1.349	-14.544	0.11031

Table 7-8 Elevator deflection related stability/control dataset

Flapped Aileron deflection result:



controlPoints	Xde	Yde	Zde	Lde	Mde	Nde	CXde	CYde	CZde	CLde	CMde	CNde
-15	-1.5004	0.87959	0.28492	9.9787	0.002532	-0.08249	-0.06569	0.038508	0.012473	0.30989	0.000574	-0.00256
-14	-1.08	0.78596	0.16611	9.9727	0.001374	-0.08219	-0.04731	0.034427	0.007276	0.30987	0.000311	-0.00255
-13	-0.66455	0.69328	0.073106	9.9618	-5.60E-05	-0.08183	-0.02913	0.030389	0.003205	0.30975	-1.27E-05	-0.00254
-12	-0.32601	0.61376	0.004688	9.9489	-0.00143	-0.08146	-0.0143	0.026924	0.000206	0.30958	-0.00032	-0.00253
-11	-0.083	0.55038	-0.04347	9.9359	-0.00255	-0.08112	-0.00364	0.02416	-0.00191	0.30939	-0.00058	-0.00253
-10	0.074474	0.5013	-0.07606	9.9239	-0.00333	-0.08082	0.003271	0.022019	-0.00334	0.3092	-0.00076	-0.00252
-9	0.1659	0.46315	-0.09702	9.9132	-0.00378	-0.08057	0.007291	0.020354	-0.00426	0.30903	-0.00086	-0.00251
-8	0.21055	0.43273	-0.10938	9.9039	-0.00391	-0.08037	0.009257	0.019026	-0.00481	0.30888	-0.00089	-0.00251
-7	0.22411	0.40757	-0.11535	9.8962	-0.00379	-0.08023	0.009857	0.017926	-0.00507	0.30875	-0.00086	-0.0025
-6	0.21839	0.38598	-0.11656	9.8903	-0.00346	-0.08014	0.009609	0.016983	-0.00513	0.30867	-0.00079	-0.0025
-5	0.2017	0.36707	-0.11415	9.8863	-0.00297	-0.08011	0.008877	0.016156	-0.00502	0.30865	-0.00068	-0.0025
-4	0.17858	0.35062	-0.1085	9.8849	-0.00236	-0.08015	0.007863	0.015437	-0.00478	0.30871	-0.00054	-0.0025
-3	0.14858	0.33705	-0.0983	9.8867	-0.00166	-0.08029	0.006544	0.014844	-0.00433	0.30887	-0.00038	-0.00251
-2	0.10677	0.3268	-0.07919	9.892	-0.00095	-0.08052	0.004704	0.014398	-0.00349	0.30914	-0.00022	-0.00252
-1	0.053154	0.31968	-0.0453	9.8992	-0.00035	-0.08078	0.002343	0.014088	-0.002	0.30946	-8.03E-05	-0.00253
0	-0.00115	0.31677	0.002021	9.9029	7.51E-05	-0.08091	-5.09E-05	0.013962	8.91E-05	0.30961	1.71E-05	-0.00253
1	-0.05549	0.31993	0.048788	9.899	0.000502	-0.08078	-0.00245	0.014099	0.00215	0.30945	0.000115	-0.00253
2	-0.10907	0.32721	0.081692	9.8917	0.001102	-0.08051	-0.00481	0.014416	0.003599	0.30913	0.000251	-0.00252
3	-0.15107	0.33756	0.1002	9.8862	0.001809	-0.08028	-0.00665	0.014867	0.004413	0.30885	0.000412	-0.00251
4	-0.1816	0.35117	0.11024	9.8842	0.002506	-0.08015	-0.008	0.015461	0.004853	0.30869	0.000571	-0.0025
5	-0.20535	0.36761	0.11591	9.8854	0.003122	-0.0801	-0.00904	0.016179	0.005102	0.30862	0.000711	-0.0025
6	-0.22254	0.38649	0.11839	9.8892	0.003615	-0.08013	-0.00979	0.017005	0.005209	0.30864	0.000823	-0.0025
7	-0.22854	0.40804	0.11725	9.895	0.003944	-0.08022	-0.01005	0.017947	0.005157	0.30871	0.000898	-0.0025
8	-0.215	0.4332	0.11132	9.9025	0.004064	-0.08036	-0.00945	0.019046	0.004894	0.30883	0.000925	-0.00251
9	-0.17006	0.46364	0.098942	9.9115	0.003927	-0.08056	-0.00747	0.020376	0.004348	0.30898	0.000893	-0.00251
10	-0.07797	0.50187	0.07788	9.922	0.003485	-0.08081	-0.00343	0.022045	0.003421	0.30915	0.000792	-0.00252
11	0.080578	0.55111	0.045064	9.9339	0.002703	-0.0811	0.003537	0.024193	0.001978	0.30933	0.000614	-0.00253
12	0.32507	0.61471	-0.00355	9.9466	0.001585	-0.08144	0.01426	0.026966	-0.00016	0.30951	0.00036	-0.00253
13	0.66518	0.69447	-0.07276	9.9594	0.000212	-0.08181	0.029158	0.030442	-0.00319	0.30968	4.81E-05	-0.00254
14	1.0814	0.78725	-0.16699	9.9701	-0.00122	-0.08217	0.04737	0.034484	-0.00731	0.30979	-0.00028	-0.00255
15	1.4999	0.88052	-0.28739	9.9759	-0.00237	-0.08247	0.065665	0.038549	-0.01258	0.3098	-0.00054	-0.00256

Table 7-9 Aileron deflection result for SC0 flapped model

Stability derivatives based on mass variation of base model

mass	Yv	Yp	Yr	Lv	Lp	Lr	Nv	Np	Nr	CYb	CYp	CYr	Clb	Clp	Clr	Cnb	Cnp	Cnr
1.32	-0.68297	-0.11873	0.46919	-0.00124	-1.401	0.18317	0.42027	-0.06952	-0.28302	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10645	-0.02498	-0.10171
1.37	-0.69577	-0.12095	0.47798	-0.00126	-1.4273	0.18661	0.42815	-0.07106	-0.28835	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10646	-0.02507	-0.10171
1.42	-0.70834	-0.12314	0.48661	-0.00129	-1.4531	0.18998	0.43589	-0.07258	-0.29372	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10646	-0.02515	-0.10177
1.47	-0.72068	-0.12528	0.49509	-0.00131	-1.4784	0.19329	0.4435	-0.07407	-0.29889	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10646	-0.02523	-0.10179
1.52	-0.73282	-0.12739	0.50343	-0.00133	-1.5033	0.19654	0.45097	-0.07553	-0.3039	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10646	-0.0253	-0.10178
1.57	-0.74476	-0.12947	0.51163	-0.00135	-1.5278	0.19975	0.45833	-0.07697	-0.30896	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10646	-0.02536	-0.10182
1.62	-0.75652	-0.13151	0.51971	-0.00137	-1.5519	0.2029	0.46557	-0.07838	-0.31398	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10646	-0.02543	-0.10186
1.67	-0.76809	-0.13353	0.52766	-0.00139	-1.5756	0.206	0.47269	-0.07977	-0.3187	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10646	-0.02549	-0.10184
1.72	-0.77949	-0.13551	0.53549	-0.00142	-1.599	0.20906	0.47972	-0.08114	-0.32356	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10647	-0.02555	-0.10187
1.77	-0.79073	-0.13746	0.54321	-0.00144	-1.6221	0.21207	0.48663	-0.08248	-0.32829	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10647	-0.0256	-0.1019
1.82	-0.80181	-0.13939	0.55082	-0.00146	-1.6448	0.21505	0.49346	-0.08382	-0.33302	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10647	-0.02566	-0.10193
1.87	-0.81274	-0.14129	0.55833	-0.00148	-1.6672	0.21798	0.50019	-0.08512	-0.3375	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10647	-0.0257	-0.10192
1.92	-0.82352	-0.14316	0.56574	-0.0015	-1.6893	0.22087	0.50683	-0.0864	-0.34209	-0.24388	-0.06015	0.23769	-0.00031	-0.50346	0.065824	0.10647	-0.02575	-0.10195

Table 7-10 Lateral Stability Derivatives

mass	Xu	Xw	Zu	Zw	Zq	Mu	Mw	Mq	Cxu	Cxa	Czu	CLa	CLq	Cmu	Cma	Cmq	X_NP
1.32	-0.07778	0.37101	-1.513	-14.598	-2.3703	-0.00274	-0.75214	-0.76007	-0.02777	0.13248	0.002714	5.2128	8.7629	-0.00507	-1.3903	-14.546	0.11189
1.37	-0.07868	0.37581	-1.5414	-14.872	-2.4147	-0.00278	-0.76635	-0.77431	-0.02758	0.13173	0.002714	5.2128	8.7629	-0.00505	-1.3905	-14.546	0.1119
1.42	-0.07894	0.38051	-1.5692	-15.14	-2.4583	-0.00277	-0.78032	-0.7883	-0.02718	0.13101	0.002714	5.2128	8.7629	-0.00493	-1.3908	-14.546	0.1119
1.47	-0.082	0.38523	-1.5966	-15.404	-2.5011	-0.00294	-0.79403	-0.80203	-0.02775	0.13036	0.002714	5.2128	8.7629	-0.00515	-1.391	-14.546	0.11191
1.52	-0.08289	0.39001	-1.6235	-15.663	-2.5433	-0.003	-0.8075	-0.81554	-0.02759	0.12979	0.002714	5.2128	8.7629	-0.00516	-1.3911	-14.546	0.11192
1.57	-0.08365	0.39477	-1.6499	-15.919	-2.5847	-0.003	-0.82076	-0.82883	-0.02739	0.12927	0.002714	5.2128	8.7629	-0.00509	-1.3913	-14.546	0.11192
1.62	-0.08392	0.3994	-1.676	-16.17	-2.6255	-0.00301	-0.83381	-0.84191	-0.02705	0.12876	0.002714	5.2128	8.7629	-0.00501	-1.3915	-14.546	0.11193
1.67	-0.08494	0.40394	-1.7016	-16.417	-2.6657	-0.00307	-0.84666	-0.85479	-0.02697	0.12826	0.002714	5.2128	8.7629	-0.00504	-1.3916	-14.546	0.11194
1.72	-0.08535	0.40847	-1.7269	-16.661	-2.7052	-0.00305	-0.85932	-0.86749	-0.0267	0.1278	0.002714	5.2128	8.7629	-0.00495	-1.3918	-14.546	0.11194
1.77	-0.08591	0.41293	-1.7517	-16.901	-2.7442	-0.00308	-0.87181	-0.88	-0.0265	0.12736	0.002714	5.2128	8.7629	-0.00492	-1.3919	-14.546	0.11195
1.82	-0.08612	0.41731	-1.7763	-17.138	-2.7827	-0.00308	-0.88412	-0.89234	-0.0262	0.12693	0.002714	5.2128	8.7629	-0.00485	-1.3921	-14.546	0.11195
1.87	-0.08731	0.42164	-1.8005	-17.372	-2.8206	-0.00312	-0.89626	-0.9045	-0.0262	0.12652	0.002714	5.2128	8.7629	-0.00485	-1.3922	-14.546	0.11196
1.92	-0.08752	0.4259	-1.8244	-17.602	-2.858	-0.00312	-0.90824	-0.91651	-0.02592	0.12613	0.002714	5.2128	8.7629	-0.00479	-1.3924	-14.547	0.11196

Table 7-11 Longitudinal Stability Derivatives

Stability Derivatives based on mass variation of flapped model

mass	Yv	Yp	Yr	Lv	Lp	Lr	Nv	Np	Nr	CYb	CYp	CYr	Clb	Clp	Clr	Cnb	Cnp	Cnr
1.32	-0.46015	-0.07269	0.32662	0.007282	-0.90728	0.20111	0.28111	-0.10143	-0.19182	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10352	-0.05299	-0.10021
1.37	-0.46877	-0.07405	0.33274	0.007419	-0.92428	0.20488	0.28639	-0.10359	-0.19546	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10352	-0.05312	-0.10024
1.42	-0.47724	-0.07539	0.33875	0.007553	-0.94098	0.20858	0.29158	-0.10572	-0.19912	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10353	-0.05325	-0.1003
1.47	-0.48556	-0.0767	0.34466	0.007684	-0.95738	0.21221	0.29668	-0.10782	-0.20263	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10353	-0.05338	-0.10032
1.52	-0.49374	-0.07799	0.35046	0.007814	-0.97351	0.21579	0.3017	-0.10989	-0.20617	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10354	-0.0535	-0.10038
1.57	-0.50178	-0.07927	0.35617	0.007941	-0.98937	0.21931	0.30663	-0.11194	-0.20966	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10355	-0.05363	-0.10044
1.62	-0.5097	-0.08052	0.3618	0.008067	-1.005	0.22277	0.31149	-0.11396	-0.21295	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10355	-0.05375	-0.10043
1.67	-0.5175	-0.08175	0.36733	0.00819	-1.0204	0.22617	0.31627	-0.11592	-0.21632	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10356	-0.05385	-0.10049
1.72	-0.52518	-0.08296	0.37278	0.008312	-1.0355	0.22953	0.32098	-0.11785	-0.21965	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10356	-0.05395	-0.10054
1.77	-0.53275	-0.08416	0.37816	0.008431	-1.0504	0.23284	0.32562	-0.11977	-0.22293	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10357	-0.05404	-0.10059
1.82	-0.54022	-0.08534	0.38346	0.008549	-1.0651	0.2361	0.33019	-0.12164	-0.226	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10357	-0.05413	-0.10057
1.87	-0.54758	-0.0865	0.38868	0.008666	-1.0797	0.23932	0.33471	-0.12348	-0.22917	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10357	-0.05421	-0.10061
1.92	-0.55485	-0.08765	0.39384	0.008781	-1.094	0.2425	0.33916	-0.12529	-0.23232	-0.23888	-0.05353	0.24055	0.002682	-0.47398	0.10506	0.10358	-0.05429	-0.10065

Table 7-12 Lateral Stability Derivatives

mass	Xu	Xw	Zu	Zw	Zq	Mu	Mw	Mq	Cxu	Cxa	Czu	CLa	CLq	Cmu	Cma	Cmq	X_NP
1.32	-0.17252	0.46916	-2.1997	-9.0877	-1.4687	-0.00333	-0.57844	-0.52331	-0.08956	0.24355	0.005716	4.7177	7.8937	-0.00895	-1.5544	-14.56	0.12401
1.37	-0.1758	0.47744	-2.2409	-9.2579	-1.4962	-0.00335	-0.58927	-0.53309	-0.08958	0.24329	0.005716	4.7177	7.8937	-0.00884	-1.5544	-14.559	0.12401
1.42	-0.17814	0.4856	-2.2813	-9.4251	-1.5232	-0.00338	-0.5999	-0.5427	-0.08916	0.24306	0.005716	4.7177	7.8937	-0.00877	-1.5544	-14.558	0.12401
1.47	-0.18083	0.49361	-2.3211	-9.5894	-1.5498	-0.00345	-0.61035	-0.55214	-0.08896	0.24284	0.005716	4.7177	7.8937	-0.00878	-1.5544	-14.558	0.12401
1.52	-0.18306	0.50147	-2.3602	-9.751	-1.5759	-0.00348	-0.62063	-0.56143	-0.08857	0.24262	0.005716	4.7177	7.8937	-0.00871	-1.5544	-14.557	0.12401
1.57	-0.18523	0.50919	-2.3987	-9.9099	-1.6016	-0.00351	-0.63073	-0.57056	-0.08818	0.2424	0.005716	4.7177	7.8937	-0.00865	-1.5543	-14.557	0.12401
1.62	-0.18838	0.51685	-2.4365	-10.066	-1.6268	-0.00353	-0.64068	-0.57954	-0.08829	0.24223	0.005716	4.7177	7.8937	-0.00856	-1.5543	-14.557	0.12401
1.67	-0.19055	0.52454	-2.4738	-10.22	-1.6517	-0.00355	-0.65047	-0.58839	-0.08796	0.24213	0.005716	4.7177	7.8937	-0.00849	-1.5543	-14.556	0.12401
1.72	-0.19261	0.53211	-2.5105	-10.372	-1.6763	-0.00358	-0.66011	-0.59711	-0.08761	0.24203	0.005716	4.7177	7.8937	-0.00843	-1.5543	-14.556	0.12401
1.77	-0.19475	0.53957	-2.5467	-10.521	-1.7004	-0.00361	-0.66961	-0.6057	-0.08732	0.24193	0.005716	4.7177	7.8937	-0.00838	-1.5542	-14.555	0.12401
1.82	-0.19762	0.54702	-2.5824	-10.669	-1.7242	-0.00365	-0.67897	-0.61417	-0.08739	0.24189	0.005716	4.7177	7.8937	-0.00836	-1.5542	-14.555	0.124
1.87	-0.20008	0.55446	-2.6176	-10.814	-1.7477	-0.00364	-0.6882	-0.62253	-0.08728	0.24188	0.005716	4.7177	7.8937	-0.00821	-1.5541	-14.555	0.124
1.92	-0.20204	0.56182	-2.6523	-10.958	-1.7709	-0.00366	-0.69731	-0.63078	-0.08698	0.24188	0.005716	4.7177	7.8937	-0.00815	-1.5541	-14.554	0.124

Table 7-13 Longitudinal Stability Derivatives

Stability Derivatives based on C.G. variation for the base model

CoG_x	Yv	Yp	Yr	Lv	Lp	Lr	Nv	Np	Nr	CYb	CYp	CYr	Clb	Clp	Clr	Cnb	Cnp	Cnr
0.009	-0.98414	-0.16105	0.72022	-0.0221	-2.0039	0.15453	0.66241	-0.01527	-0.48001	-0.24614	-0.05715	0.25555	-0.00392	-0.50438	0.038893	0.11752	-0.00384	-0.12081
0.014	-0.95853	-0.15733	0.69698	-0.02049	-1.9525	0.15579	0.6401	-0.02015	-0.46067	-0.24603	-0.05729	0.2538	-0.00373	-0.50434	0.04024	0.11654	-0.00521	-0.11899
0.019	-0.93229	-0.15354	0.67356	-0.01881	-1.8999	0.15725	0.61759	-0.02463	-0.44131	-0.2459	-0.05746	0.25205	-0.00352	-0.50429	0.04174	0.11555	-0.00654	-0.11714
0.024	-0.90535	-0.14967	0.64992	-0.01707	-1.8458	0.15895	0.59487	-0.02947	-0.42215	-0.24577	-0.05764	0.2503	-0.00329	-0.50424	0.043423	0.11455	-0.00805	-0.11532
0.029	-0.87767	-0.14571	0.62605	-0.01524	-1.7903	0.16093	0.57191	-0.03427	-0.40292	-0.24561	-0.05785	0.24855	-0.00303	-0.50417	0.045319	0.11353	-0.00965	-0.11347
0.034	-0.84919	-0.14167	0.60191	-0.01333	-1.7332	0.16321	0.54867	-0.03936	-0.384	-0.24543	-0.05809	0.2468	-0.00273	-0.5041	0.04747	0.11249	-0.01145	-0.11169
0.039	-0.81984	-0.13753	0.57749	-0.01132	-1.6744	0.16586	0.52514	-0.04487	-0.36499	-0.24523	-0.05836	0.24506	-0.0024	-0.50402	0.049926	0.11142	-0.01351	-0.10987
0.044	-0.78955	-0.13329	0.55274	-0.00919	-1.6138	0.16894	0.50127	-0.05018	-0.34593	-0.24499	-0.05868	0.24332	-0.00202	-0.50393	0.052753	0.11033	-0.01567	-0.10802
0.049	-0.75825	-0.12896	0.52764	-0.00694	-1.5512	0.17253	0.47701	-0.05569	-0.32681	-0.24471	-0.05904	0.24159	-0.00159	-0.50381	0.056036	0.1092	-0.01809	-0.10614
0.054	-0.72585	-0.12452	0.50215	-0.00454	-1.4865	0.17674	0.45232	-0.06148	-0.30758	-0.24439	-0.05948	0.23986	-0.00108	-0.50368	0.059885	0.10803	-0.02083	-0.10422
0.059	-0.69225	-0.11997	0.47625	-0.00197	-1.4195	0.18169	0.42715	-0.06775	-0.28831	-0.244	-0.05999	0.23815	-0.00049	-0.50351	0.064446	0.1068	-0.02403	-0.10227
0.064	-0.65739	-0.11533	0.44991	0.000801	-1.3501	0.18755	0.40145	-0.07466	-0.26881	-0.24353	-0.06061	0.23645	0.000211	-0.50331	0.069918	0.10549	-0.02783	-0.10021
0.069	-0.62117	-0.1106	0.42312	0.003799	-1.2781	0.19454	0.37523	-0.0824	-0.24969	-0.24295	-0.06137	0.23478	0.001054	-0.50305	0.07657	0.1041	-0.03243	-0.09828
0.074	-0.58353	-0.10579	0.39589	0.007062	-1.2035	0.20294	0.34835	-0.09109	-0.2301	-0.24222	-0.0623	0.23313	0.002079	-0.50272	0.084771	0.10257	-0.03805	-0.09612
0.079	-0.54448	-0.10095	0.3683	0.01063	-1.1263	0.2131	0.32081	-0.10058	-0.21032	-0.24129	-0.06347	0.23155	0.003341	-0.50227	0.095034	0.10085	-0.04486	-0.09379
0.084	-0.50413	-0.09614	0.3405	0.014538	-1.0468	0.22548	0.29261	-0.11164	-0.19035	-0.24007	-0.06495	0.23004	0.004911	-0.50167	0.10806	0.098843	-0.0535	-0.09122
0.089	-0.46275	-0.09146	0.31277	0.018807	-0.96575	0.24061	0.26406	-0.12822	-0.17061	-0.23846	-0.06686	0.22866	0.006875	-0.50082	0.12478	0.096523	-0.06649	-0.08848
0.094	-0.4209	-0.08703	0.28562	0.023412	-0.88431	0.25903	0.2355	-0.14978	-0.15118	-0.2363	-0.06932	0.22749	0.009324	-0.49962	0.14635	0.093787	-0.08462	-0.08541

Table 7-14 Lateral Stability Derivatives

CoG_x	Xu	Xw	Zu	Zw	Zq	Mu	Mw	Mq	Cxu	Cxa	Czu	CLa	CLq	Cmu	Cma	Cmq	X_NP
0.009	-0.07446	0.22818	-1.0603	-20.881	-4.4901	-0.00332	-2.0757	-1.3672	-0.01862	0.05707	0.001192	5.2226	11.627	-0.0043	-2.6874	-18.327	0.10842
0.014	-0.07444	0.23329	-1.088	-20.346	-4.2718	-0.00331	-1.9246	-1.3009	-0.01911	0.059879	0.001281	5.2222	11.352	-0.00439	-2.5572	-17.895	0.10861
0.019	-0.07474	0.24155	-1.118	-19.798	-4.056	-0.00326	-1.7777	-1.2364	-0.01972	0.063713	0.001378	5.2219	11.076	-0.00445	-2.4273	-17.477	0.10881
0.024	-0.07404	0.24997	-1.1505	-19.235	-3.8429	-0.00321	-1.6354	-1.1736	-0.0201	0.067856	0.001485	5.2214	10.8	-0.00451	-2.2981	-17.074	0.10904
0.029	-0.07469	0.26016	-1.186	-18.656	-3.6322	-0.00319	-1.4976	-1.1124	-0.0209	0.072804	0.001603	5.2208	10.523	-0.00462	-2.1695	-16.683	0.10929
0.034	-0.07234	0.26481	-1.2248	-18.062	-3.4242	-0.00293	-1.3652	-1.0528	-0.02091	0.076534	0.001733	5.2202	10.246	-0.00439	-2.0424	-16.308	0.1096
0.039	-0.0732	0.27612	-1.2675	-17.449	-3.2187	-0.00293	-1.2374	-0.99468	-0.0219	0.082592	0.001879	5.2194	9.9676	-0.00453	-1.9159	-15.945	0.10993
0.044	-0.07397	0.2933	-1.3148	-16.818	-3.0158	-0.0029	-1.1145	-0.93784	-0.02295	0.091007	0.002042	5.2184	9.6881	-0.00465	-1.7901	-15.596	0.11028
0.049	-0.07533	0.31314	-1.3675	-16.165	-2.8154	-0.00289	-0.99703	-0.88224	-0.02431	0.10106	0.002223	5.2171	9.4073	-0.00483	-1.6657	-15.26	0.11069
0.054	-0.07708	0.33567	-1.4266	-15.49	-2.6176	-0.00289	-0.88532	-0.82778	-0.02595	0.11302	0.002427	5.2155	9.1247	-0.00503	-1.5431	-14.937	0.11117
0.059	-0.0767	0.36309	-1.4934	-14.791	-2.4225	-0.00272	-0.77959	-0.77434	-0.02704	0.12798	0.002651	5.2134	8.8401	-0.00497	-1.4224	-14.628	0.11172
0.064	-0.08008	0.39265	-1.5697	-14.066	-2.2301	-0.00277	-0.68044	-0.72181	-0.02966	0.14545	0.002891	5.2107	8.5529	-0.00531	-1.3048	-14.331	0.11239
0.069	-0.08316	0.42229	-1.6575	-13.313	-2.0405	-0.00272	-0.58802	-0.66986	-0.03252	0.16516	0.00315	5.207	8.2624	-0.00552	-1.1905	-14.041	0.11318
0.074	-0.08705	0.45735	-1.7594	-12.532	-1.854	-0.00271	-0.50336	-0.61877	-0.03613	0.18984	0.003415	5.2019	7.9677	-0.00583	-1.0816	-13.765	0.11418
0.079	-0.09061	0.50048	-1.8789	-11.722	-1.6712	-0.00261	-0.42666	-0.56836	-0.04015	0.22179	0.003653	5.1947	7.6674	-0.00598	-0.97875	-13.499	0.11541
0.084	-0.09849	0.55359	-2.02	-10.886	-1.4929	-0.00267	-0.35945	-0.51912	-0.0469	0.26363	0.003763	5.1841	7.3602	-0.00659	-0.8861	-13.249	0.11703
0.089	-0.10838	0.56569	-2.1873	-10.03	-1.3203	-0.00261	-0.30525	-0.47088	-0.05585	0.29151	0.003663	5.1684	7.0441	-0.00695	-0.81429	-13.005	0.11945
0.094	-0.12322	0.57419	-2.3849	-9.1634	-1.1557	-0.00258	-0.26255	-0.42406	-0.06918	0.32236	0.003288	5.1445	6.7174	-0.00751	-0.76303	-12.759	0.12267

Table 7-15 Longitudinal Stability Derivatives

Stability derivatives based on C.G. variations for base model (flapped)

CoG_x	Yv	Yp	Yr	Lv	Lp	Lr	Nv	Np	Nr	CYb	CYp	CYr	Clb	Clp	Clr	Cnb	Cnp	Cnr
0.009	-0.6398	-0.0927	0.4788	-0.01788	-1.2424	0.13629	0.43343	-0.0382	-0.31749	-0.24367	-0.05009	0.25871	-0.00483	-0.4762	0.052238	0.1171	-0.01464	-0.12169
0.014	-0.62406	-0.09078	0.46429	-0.01579	-1.2128	0.14051	0.41916	-0.04302	-0.30517	-0.24342	-0.05024	0.25693	-0.00437	-0.47609	0.055156	0.11598	-0.01689	-0.11979
0.019	-0.60799	-0.08885	0.44972	-0.01364	-1.1827	0.14505	0.40481	-0.04816	-0.29253	-0.24314	-0.05041	0.25515	-0.00387	-0.47596	0.058376	0.11483	-0.01938	-0.11773
0.024	-0.59157	-0.08691	0.43508	-0.01142	-1.1519	0.14995	0.39036	-0.0532	-0.28043	-0.24283	-0.05061	0.25337	-0.00333	-0.47583	0.061942	0.11366	-0.02198	-0.11584
0.029	-0.57479	-0.08496	0.42038	-0.00914	-1.1204	0.15524	0.37583	-0.05834	-0.26797	-0.24249	-0.05085	0.2516	-0.00273	-0.47567	0.065905	0.11247	-0.02477	-0.11376
0.034	-0.55763	-0.083	0.4056	-0.00677	-1.0883	0.16097	0.36113	-0.06417	-0.25586	-0.24209	-0.05112	0.24982	-0.00209	-0.4755	0.070328	0.11121	-0.02804	-0.11179
0.039	-0.54006	-0.08103	0.39075	-0.00432	-1.0555	0.1672	0.34638	-0.06911	-0.24373	-0.24165	-0.05144	0.24805	-0.00137	-0.4753	0.075287	0.10994	-0.03112	-0.10975
0.044	-0.52208	-0.07907	0.37584	-0.00178	-1.022	0.17399	0.33142	-0.07561	-0.23139	-0.24114	-0.05181	0.24628	-0.00058	-0.47507	0.080873	0.10859	-0.03514	-0.10756
0.049	-0.50366	-0.0771	0.36085	0.000861	-0.9878	0.18141	0.31632	-0.08217	-0.21931	-0.24056	-0.05225	0.24452	0.000292	-0.4748	0.087197	0.10717	-0.0395	-0.10542
0.054	-0.48479	-0.07515	0.3458	0.003613	-0.95281	0.18956	0.30102	-0.08944	-0.20721	-0.23989	-0.05276	0.24276	0.001268	-0.47448	0.094395	0.10566	-0.04454	-0.10318
0.059	-0.46546	-0.07321	0.33071	0.006484	-0.91709	0.19852	0.28547	-0.09782	-0.195	-0.23911	-0.05335	0.24102	0.002363	-0.4741	0.10263	0.10402	-0.05057	-0.10081
0.064	-0.44568	-0.0713	0.3156	0.009479	-0.88063	0.20843	0.26966	-0.10824	-0.18303	-0.23819	-0.05406	0.2393	0.003594	-0.47364	0.1121	0.10223	-0.05822	-0.09844
0.069	-0.42543	-0.06942	0.30051	0.012603	-0.84349	0.21941	0.25387	-0.12032	-0.17122	-0.23711	-0.05489	0.2376	0.004982	-0.47308	0.12306	0.10036	-0.06748	-0.09603
0.074	-0.40476	-0.06761	0.28547	0.015857	-0.80573	0.23161	0.23787	-0.13543	-0.15974	-0.23581	-0.05588	0.23594	0.006553	-0.47239	0.13579	0.098303	-0.0794	-0.09365
0.079	-0.38368	-0.06587	0.27056	0.019232	-0.76746	0.24519	0.22193	-0.14997	-0.14861	-0.23424	-0.05705	0.23434	0.008329	-0.47152	0.15064	0.09611	-0.09214	-0.0913
0.084	-0.36204	-0.06418	0.25571	0.022694	-0.72835	0.26013	0.20577	-0.16494	-0.13698	-0.23235	-0.05844	0.23282	0.010331	-0.47041	0.168	0.093676	-0.10653	-0.08847
0.089	-0.33787	-0.06219	0.23957	0.026035	-0.68448	0.27482	0.18811	-0.17887	-0.1244	-0.23005	-0.06008	0.23141	0.012575	-0.46901	0.18831	0.090855	-0.12256	-0.08524
0.094	-0.31461	-0.06051	0.22461	0.029388	-0.64273	0.29162	0.17092	-0.19498	-0.11239	-0.22724	-0.06201	0.23016	0.015057	-0.46719	0.21198	0.087574	-0.14173	-0.0817

Table 7-16 lateral Stability Derivatives

CoG_x	Xu	Xw	Zu	Zw	Zq	Mu	Mw	Mq	Cxu	Cxa	Czu	CLa	CLq	Cmu	Cma	Cmq	X_NP
0.009	-0.15511	0.39452	-1.6201	-12.52	-2.6851	-0.00512	-1.345	-0.88757	-0.05907	0.15026	0.000662	4.7683	10.588	-0.01009	-2.6518	-18.117	0.11644
0.014	-0.15523	0.40612	-1.6585	-12.219	-2.5582	-0.00497	-1.2563	-0.84729	-0.06055	0.15841	0.000992	4.7663	10.331	-0.01004	-2.5367	-17.713	0.11682
0.019	-0.15632	0.41686	-1.6996	-11.912	-2.4331	-0.00492	-1.1702	-0.80813	-0.06252	0.16671	0.00135	4.764	10.074	-0.01018	-2.4226	-17.321	0.11725
0.024	-0.15269	0.43625	-1.7438	-11.599	-2.3096	-0.00447	-1.0864	-0.77006	-0.06268	0.17908	0.001741	4.7613	9.8156	-0.0095	-2.3086	-16.941	0.11767
0.029	-0.15366	0.46175	-1.7914	-11.279	-2.188	-0.00436	-1.005	-0.73294	-0.06483	0.1948	0.002167	4.7581	9.5563	-0.00952	-2.1947	-16.571	0.11811
0.034	-0.15407	0.472	-1.8427	-10.951	-2.0681	-0.00412	-0.92771	-0.69705	-0.06689	0.20492	0.002631	4.7544	9.2957	-0.00926	-2.0849	-16.219	0.11872
0.039	-0.15454	0.50006	-1.8984	-10.616	-1.9501	-0.00397	-0.85206	-0.66192	-0.06915	0.22375	0.003133	4.75	9.0338	-0.0092	-1.9736	-15.873	0.11927
0.044	-0.15747	0.50968	-1.959	-10.272	-1.834	-0.00382	-0.78098	-0.62787	-0.07274	0.23542	0.003674	4.7447	8.7702	-0.00913	-1.8673	-15.543	0.12004
0.049	-0.15976	0.52502	-2.0251	-9.9202	-1.7199	-0.00364	-0.71294	-0.59472	-0.07631	0.25077	0.004256	4.7382	8.5048	-0.00901	-1.7628	-15.224	0.12088
0.054	-0.16363	0.53624	-2.0974	-9.5594	-1.6079	-0.00341	-0.64879	-0.56249	-0.08097	0.26535	0.004874	4.7304	8.2373	-0.00872	-1.6619	-14.917	0.12188
0.059	-0.17111	0.52362	-2.1768	-9.1895	-1.498	-0.00334	-0.59008	-0.53127	-0.0879	0.26898	0.00553	4.7207	7.9673	-0.00888	-1.5692	-14.627	0.12322
0.064	-0.17848	0.46485	-2.2643	-8.8101	-1.3906	-0.00322	-0.53828	-0.50102	-0.09539	0.24844	0.006225	4.7086	7.6944	-0.00891	-1.4892	-14.351	0.12511
0.069	-0.18881	0.44154	-2.3609	-8.4213	-1.2857	-0.00314	-0.48836	-0.47131	-0.10523	0.24608	0.006954	4.6934	7.4183	-0.00907	-1.4089	-14.078	0.127
0.074	-0.19947	0.42052	-2.4676	-8.0231	-1.1835	-0.00284	-0.44348	-0.44269	-0.11621	0.24499	0.00774	4.6741	7.1386	-0.00857	-1.3374	-13.822	0.12929
0.079	-0.2147	0.42176	-2.5856	-7.6158	-1.0845	-0.00274	-0.40054	-0.41408	-0.13108	0.25749	0.008667	4.6495	6.8549	-0.00867	-1.2658	-13.549	0.1316
0.084	-0.23521	0.41721	-2.7139	-7.1952	-0.98829	-0.00282	-0.36188	-0.38557	-0.15096	0.26776	0.012291	4.6178	6.5667	-0.00937	-1.2023	-13.262	0.13431
0.089	-0.25957	0.43008	-2.835	-6.7218	-0.89004	-0.00294	-0.32619	-0.35613	-0.17674	0.29283	0.043934	4.5768	6.2741	-0.01036	-1.1497	-12.996	0.13754
0.094	-0.2917	0.42535	-2.9723	-6.2629	-0.79931	-0.00307	-0.29954	-0.32943	-0.2107	0.30724	0.074857	4.5238	5.9774	-0.01147	-1.12	-12.753	0.14184

Table 7-17 Longitudinal Stability Derivatives

Stability parameter estimations for SC0 (unformatted):

Inertial properties in body axis-

```
___Inertia - Body Axis - CoG Origin____
   Ibxx=    0.04119 kg.m²
   Ibyy=    0.03961 kg.m²
   Ibzz=    0.07279 kg.m²
   Ibxz=  -0.0005709 kg.m²
```

Inertial properties in stability axis:

```
___Inertia - Stability Axis - CoG Origin____
     Isxx=     0.0413
     Isyy=    0.03961
     Iszz=    0.07268
     Isxz=   -0.001947
```

Stability Derivatives:

```
Longitudinal derivatives
     Xu=   -0.077779        Cxu=   -0.027774
     Xw=    0.37101        Cxa=    0.13248
     Zu=     -1.513        Czu=   0.0027144
     Zw=    -14.598        CLa=     5.2128
     Zq=    -2.3703        CLq=     8.7629
     Mu=  -0.0027448        Cmu=  -0.0050738
     Mw=    -0.75214        Cma=    -1.3903
     Mq=    -0.76007        Cmq=    -14.546
     Neutral Point position=   0.11189 m

  Lateral derivatives
     Yv=    -0.68297        CYb=    -0.24388
     Yp=    -0.11873        CYp=   -0.060148
     Yr=     0.46919        CYr=     0.23769
     Lv=  -0.0012401        Clb= -0.00031412
     Lp=      -1.401        Clp=    -0.50346
     Lr=     0.18317        Clr=    0.065824
     Nv=     0.42027        Cnb=     0.10645
     Np=   -0.069516        Cnp=   -0.024981
     Nr=    -0.28302        Cnr=    -0.10171
```

STATE MATRICES:

```
 _____State matrices__________
   Longitudinal state matrix
      -0.0588434         0.280688              0            -9.81
       -1.14467          -11.044         15.2614              0
      -0.0692877         -18.9863        -19.1867              0
          0              0             1              0
   Lateral state matrix
       -0.516698        -0.0898234         -16.6996            9.81
       -0.302982         -33.9193         4.62439             0
        5.79077        -0.0478915        -4.01811             0
          0              1             0              0
```

Eigenvalues and vectors:

```
  ___Longitudinal modes____

  Eigenvalue:   -15.12+  -16.53i  |    -15.12+   16.53i  |  -0.02282+ -0.6397i  |  -0.02282+  0.6397i
_____________________________________________________________________________________________
__
  Eigenvector:      1+      0i  |        1+      0i  |        1+      0i  |        1+      0i
             79.54+  -14.09i  |    79.54+   14.09i  |  -0.04592+-0.0003087i  |  -0.04592+0.0003087i
            -36.44+  -82.39i  |    -36.44+   82.39i  |   0.04183+ 0.001702i  |   0.04183+-0.001702i
             3.811+   1.282i  |     3.811+  -1.282i  |  -0.004986+  0.0652i  |  -0.004986+ -0.0652i

 ___Lateral modes____

  Eigenvalue:   -33.93+      0i  |    -2.302+  -9.705i  |    -2.302+   9.705i  |   0.07429+      0i
_____________________________________________________________________________________________
_____
  Eigenvector:      1+      0i  |        1+      0i  |        1+      0i  |        1+      0i
             90.31+      0i  |  -0.01877+  0.07886i  |  -0.01877+ -0.07886i  |    0.1833+      0i
          -0.04901+      0i  |    0.1027+  0.5786i  |    0.1027+  -0.5786i  |     1.413+      0i
            -2.662+      0i  |  -0.007259+-0.003655i  |  -0.007259+ 0.003655i  |     2.467+      0i
```

```
Phillips formulae:
   Phugoid eigenvalue:     -0.00110+  0.63222i
       frequency:  0.101 Hz
       damping:    0.002
   Dutch-Roll eigenvalue:   -2.30455+  9.68687i
       frequency:  1.585 Hz
       damping:    0.238
```

Stability Parameter estimates for SC0_flap:

```
  ___Inertia - Body Axis - CoG Origin____
   Ibxx=    0.04119 kg.m²
   Ibyy=    0.03961 kg.m²
   Ibzz=    0.07279 kg.m²
   Ibxz=  -0.0005709 kg.m²

___Inertia - Stability Axis - CoG Origin____
     Isxx=    0.04138
     Isyy=    0.03961
     Iszz=     0.0726
     Isxz=   -0.002514


Longitudinal derivatives
     Xu=    -0.17252        Cxu=   -0.089561
     Xw=     0.46916        Cxa=     0.24355
     Zu=     -2.1997        Czu=   0.0057159
     Zw=     -9.0877        CLa=      4.7177
     Zq=     -1.4687        CLq=      7.8937
     Mu=  -0.0033307        Cmu=  -0.0089506
     Mw=    -0.57844        Cma=     -1.5544
     Mq=    -0.52331        Cmq=      -14.56
     Neutral Point position=   0.12401 m


Lateral derivatives
     Yv=    -0.46015        CYb=    -0.23888
     Yp=   -0.072688        CYp=   -0.053534
     Yr=     0.32662        CYr=     0.24055
     Lv=   0.0072823        Clb=   0.0026817
     Lp=    -0.90728        Clp=    -0.47398
     Lr=     0.20111        Clr=     0.10506
     Nv=     0.28111        Cnb=     0.10352
     Np=    -0.10143        Cnp=    -0.05299
     Nr=    -0.19182        Cnr=    -0.10021


    _____State matrices__________

      Longitudinal state matrix
          -0.130521       0.354937          0          -9.81
           -1.66414       -6.87521      10.62              0
         -0.0840776       -14.6016     -13.21              0
                  0              0          1              0
      Lateral state matrix
          -0.348125     -0.0549919   -11.4841           9.81
         -0.0593588       -21.8854    5.03084              0
            3.87424      -0.639409   -2.81641              0
                  0              1          0              0
```

```
___Longitudinal modes____
    Eigenvalue:    -10.07+   -12.05i   |     -10.07+    12.05i   |    -0.04134+  -0.9707i   |    -
0.04134+   0.9707i
____________________________________________________________________________
_____
    Eigenvector:        1+       0i   |         1+       0i   |         1+       0i   |         1+       0i
              35.45+    3.129i   |      35.45+   -3.129i   |   -0.09336+-0.0008086i   |   -
0.09336+0.0008086i
             -6.947+   -41.16i   |     -6.947+    41.16i   |    0.09655+ 0.008014i   |    0.09655+-
0.008014i
              2.296+    1.341i   |      2.296+   -1.341i   |   -0.01247+  0.09893i   |   -0.01247+ -
0.09893i



    ___Lateral modes____
    Eigenvalue:    -21.76+       0i   |      -1.74+   -6.626i   |      -1.74+    6.626i   |     0.1856+       0i
____________________________________________________________________________
_____
    Eigenvector:        1+       0i   |         1+       0i   |         1+       0i   |         1+       0i
              26.59+       0i   |   -0.02107+   0.1353i   |   -0.02107+  -0.1353i   |      0.278+       0i
              0.693+       0i   |     0.1056+   0.5695i   |     0.1056+  -0.5695i   |      1.231+       0i
             -1.222+       0i   |   -0.01832+ -0.00799i   |   -0.01832+  0.00799i   |      1.497+
0i



  Phillips formulae:
     Phugoid eigenvalue:      -0.02044+  0.95341i
          frequency:  0.152 Hz
          damping:    0.021
     Dutch-Roll eigenvalue:   -1.74622+  6.56260i
          frequency:  1.081 Hz
          damping:    0.266
```

7.5 Methodology Extra

The following content are additional details (relating to the Methodology section) not included within the main body of the thesis but documented here for reference.

7.5.1 Measurements

7.5.2 Horizontal Stabiliser

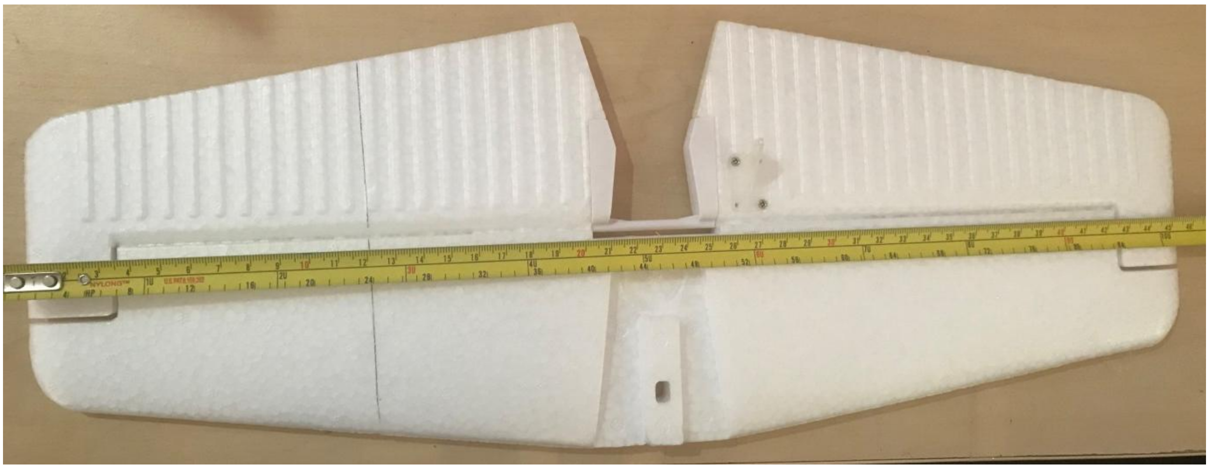


Figure 7.3 Measurement process for the horizontal stabilizer

The span of the Horizontal Stabilizer was measured with a measuring tape and determined to be 454 mm. To measure the aerofoil, because it's symmetrical, a reference chord and reference thickness was needed. As the aerofoil remains the same throughout the span of the stabiliser, the reference location can be arbitrary as long as it’s the same when both (chord and thickness) measurements are taken. As the planform changes, so does the thickness gradually decline. Which is why it is important to have a consistent reference point for this measurement. A reference chord (at approximately half span of a single elevator unit) was measured to be 140 mm. The reference maximum thickness at that same location was measured to be 14 mm (10% of the reference chord length). It is symmetric and therefore there is no camber. By standard NACA 4 digit aerofoil convention [5], this identifies it as a NACA 0010 aerofoil.

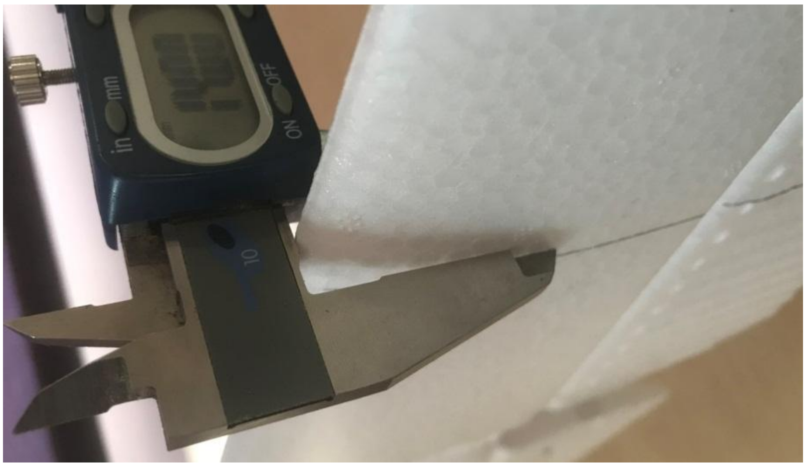


Figure 7.4 Thickness measurement for the horizontal stabiliser

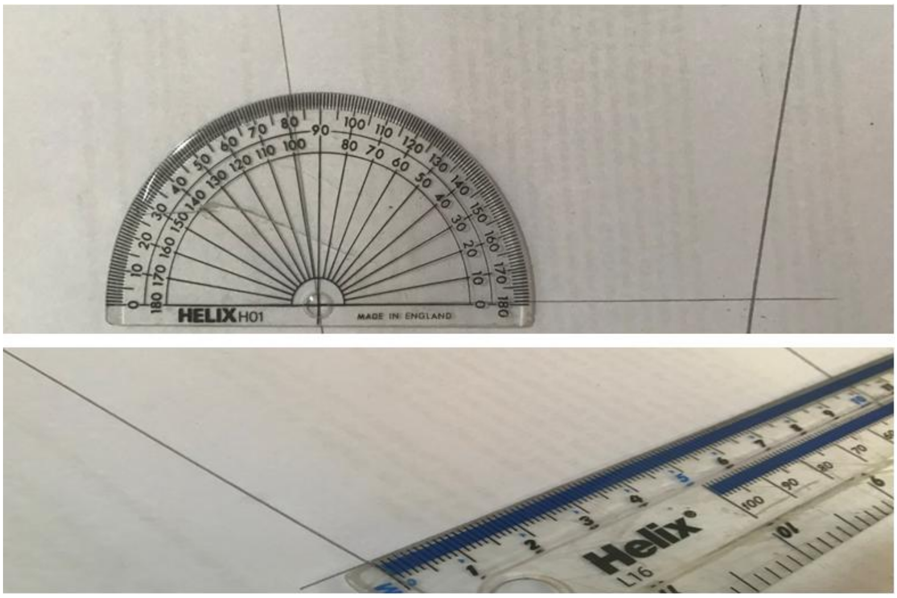


Figure 7.5 Use of tracing for measurement

The sweep angle was measured by tracing the horizontal stabiliser’s outside planform dimensions on an A4 sheet of paper and measuring the angle using a protractor. It was measured to be 8 degrees of sweep at the leading edge. The trace was also useful for obtaining the tip chord. The tip chord was measured to be 103mm. With this information, the offset at the tip was calculated the following way:

$$\tan(sweep\ angle) = \frac{offset}{(half - span)} \qquad \text{[Eq. 7.1]}$$

$$\tan(8) = \frac{offset}{(0.5 \times 454)}$$

$$offset = 31.903\ mm$$

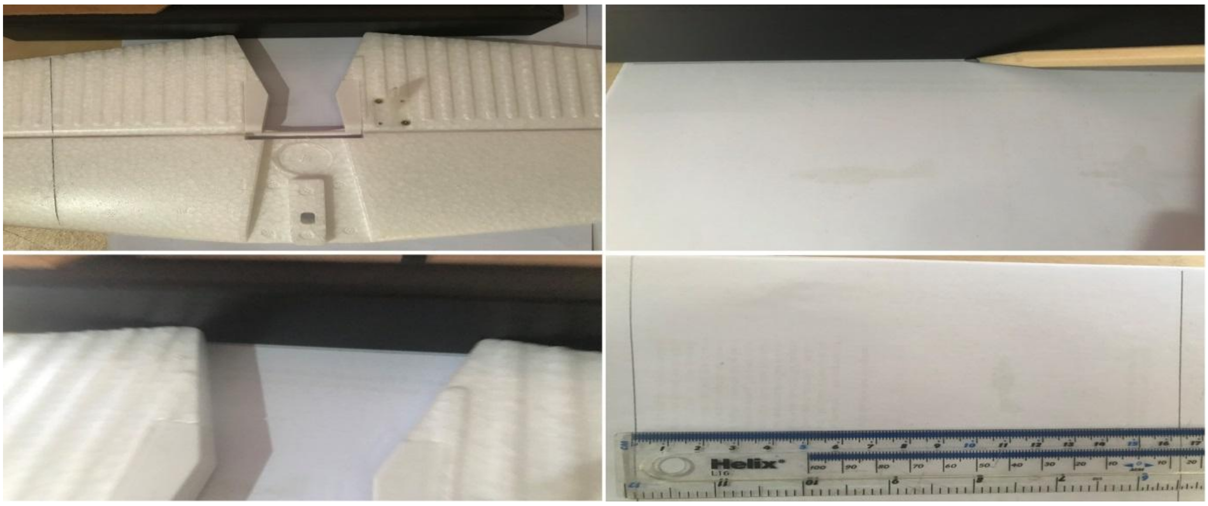


Figure 7.6 Steps in measurement via tracing for the horizontal stabiliser

As shown in the above pictures, the root chord was measured from the trace drawing to be as 169 mm. A digital scale was used to measure the weight and the horizontal stabiliser was weighed at 29 grams. The maximum deflection of the control surface was determined to be 15 degrees using a protractor.

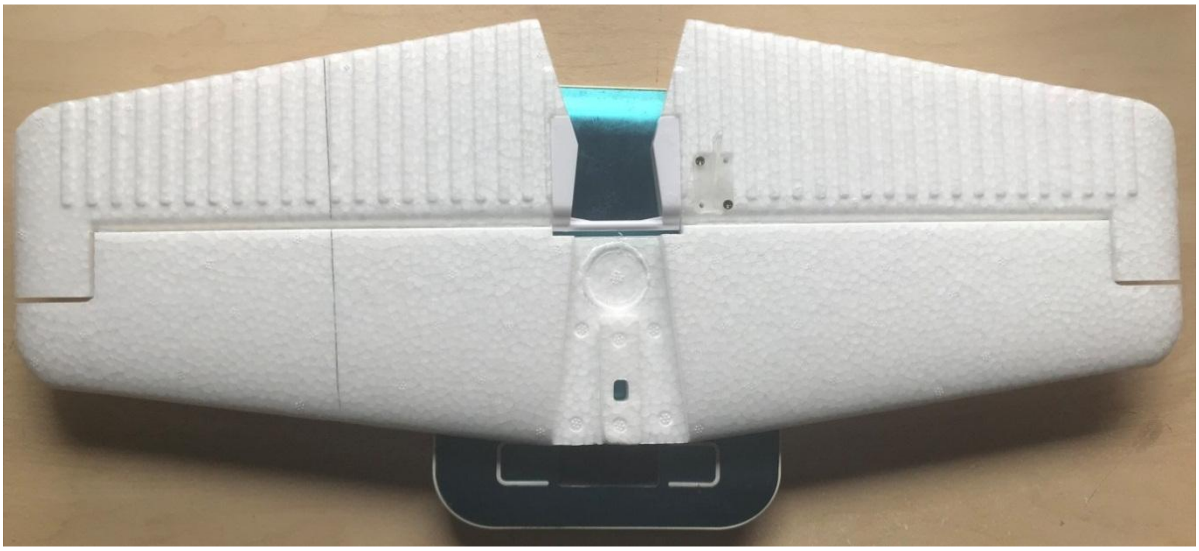


Figure 7.7 Weighing the horizontal stabiliser on a digital scale

The center of rotation of the elevator was measured with a scale and determined to be 84 mm from the leading edge of the horizontal stabiliser and spans the entire length of the horizontal stabiliser.

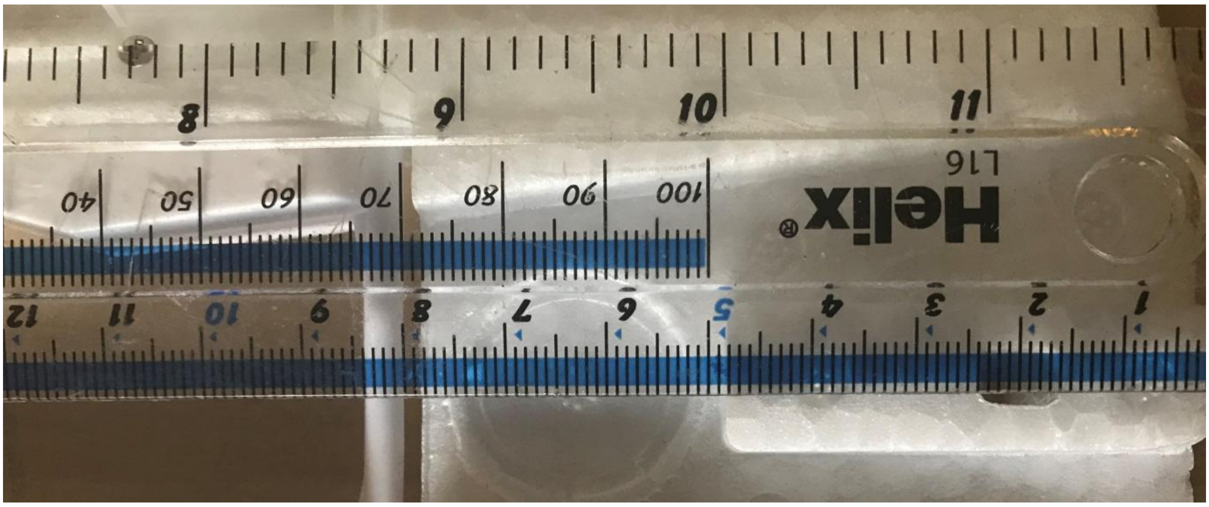


Figure 7.8 Measurement process for elevator dimension

Figure 7.9 Measurement of elevator deflection using a protractor

The maximum elevator deflection was measured using a protractor and recorded as 15 degrees in both directions.

The trailing tip of the Vertical Stabiliser 1 is 85 mm above the reference point. While the Horizontal Stabiliser chord is 205mm below the trailing tip of the vertical stab, which gives it a total vertical offset of a 120 mm below the reference point. Since the Horizontal Stabiliser is angled slightly downward relative to the wing (very small and below accuracy of the measuring tools), it was set to 1.5 degrees negative.

7.5.2.1 Vertical Stabilisers

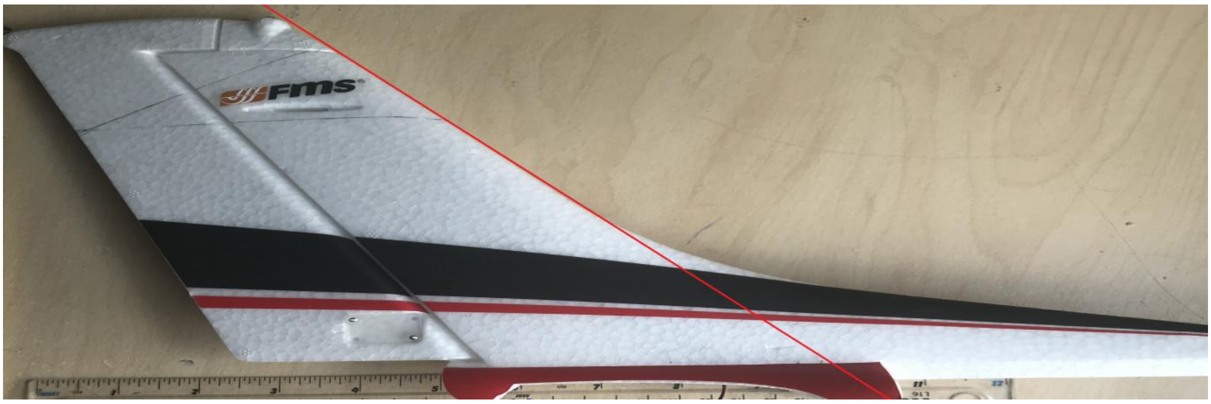


Figure 7.10 Sectioning of the vertical stabilisers (the red line marks the boundary)

For ease of modelling in both X-Plane and XFLR5, the vertical tail component has been sectioned off along the red line (as seen in the above image). The primary section (with the rudder) is referred to as Vertical Stabiliser 1 and the remaining section is referred to as Vertical Stabiliser 2.

7.5.2.1.1 Vertical Stabiliser 1

The planform of the vertical stabiliser was measured the same way through the use of a straight square edge for proper alignment for the line drawings.

Figure 7.11 Measurement process for the vertical tail section

Rules, measuring tapes and a protractor was utilised to measure the necessary dimensions. The root chord was measured to be 174mm with a ruler and the tip chord was measured to be 86 mm.

Figure 7.12 Tracing and measuring the geometry

For the sweep angle, a protractor was used and it was measured as 37.5 degrees. The span of the section was measured to be 195 mm. It should be noted that since there is no mirrored section of this aerofoil, this span is the same as half-span in the following calculation. The offset was measured as follows:

$$\tan(sweep\ angle) = \frac{offset}{(half - span)} \qquad \text{[Eq. 7.2]}$$

$$\tan(37.5) = \frac{offset}{(195)}$$

$$offset = 149.629\ mm$$

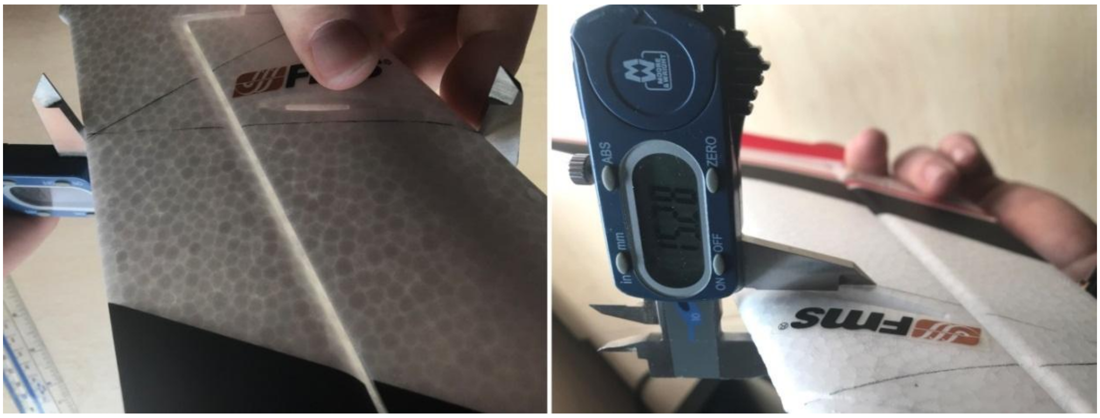


Figure 7.13 Use of callipers to take measurements of the vertical stabiliser

Multiple attempts had to be made for this foil both with altered reference chords and repeated measurements. Both the material being measured and the methods of obtaining the measurements create inconsistent results. However, the majority of them are more consistent with the following attempt. A reference chord (at approximately 25% span) was measured to be 127.39 mm with a digital calliper. The reference maximum thickness at that same location was measured with a digital calliper to be 15.28 mm (12 % of the reference chord). It is symmetric and has no camber. By standard NACA 4 digit aerofoil convention [5], this identifies it as a NACA 0012 aerofoil. The rudder was measured to be positioned at 70 mm at the wing root and 43 mm at the wing tip. In the z-axis, because the tip is 85 mm higher than the reference point, and has a span of 195 mm, that puts it at 110 mm below the ref point. The trailing tip of the Vertical Stabiliser 1 was measured to be 822 mm behind the reference point. Subtracting the tip chord of 86 mm and the previously calculated offset of 149.629, gives us a leading edge starting position of 586.371 mm. The rudder deflection was measured with a protractor and recorded as 15 degrees in both directions.

Figure 7.14 Measuring rudder root and tip lengths

Figure 7.15 Measuring rudder deflection with a protractor

7.5.2.1.2 Vertical Stabiliser 2

Figure 7.16 Measuring vertical stabiliser 2 dimensions

The root chord of the Vertical Stabiliser 2 was measured with a measuring tape to be 360 mm with a maximum thickness of 21.82 mm obtained using a digital calliper. This is a symmetrical aerofoil. It has no camber. Based on the thickness (6% of chord length) and chord length and based on standard 4 digit NACA aerofoil designations [5], it would be a NACA0006. However, the maximum thickness is in the middle of the aerofoil and not biased towards the front like a regular NACA foil, which makes it closer in geometry to camber-less compressor blades of jet engines. There are multiple modelling solutions to this with their own challenges. This is implemented very differently depending on the modelling platform. For example, in X-Plane modelling of the geometry, it can be modelled as an extension of the fuselage or as a separate section of the Vertical Stabiliser. Nevertheless, the geometric information on it is documented for aid in the modelling process.

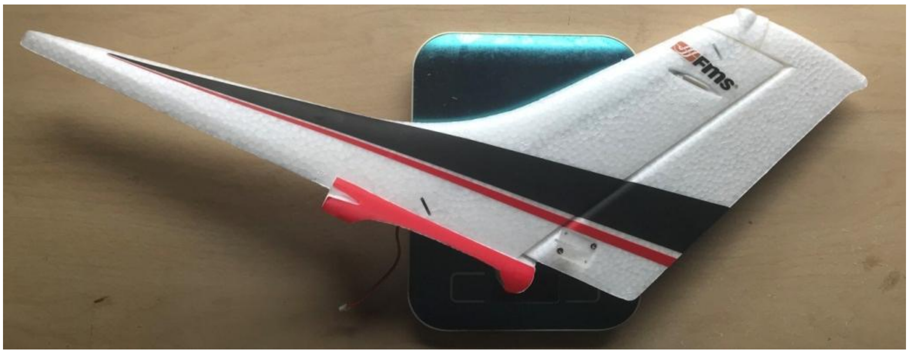


Figure 7.17 Weighing the vertical stabiliser section on a digital scale

Following a similar approach, the blending between the fuselage and the vertical stabiliser was measured. The tip chord (top surface) of this extension was measured to be 242 mm. The relative sweep between the two sections was 51 degrees. Adding to that the sweep of the vertical stabiliser (37.5 degrees), we get a total sweep for this section of 88.5 degrees. The entire section (Vertical Stabiliser 1 + Vertical Stabiliser 2) was weighed on a digital scale and it's weight was determined to be 25 grams.

7.5.2.2 Main Wing:

7.5.2.3 Fuselage:

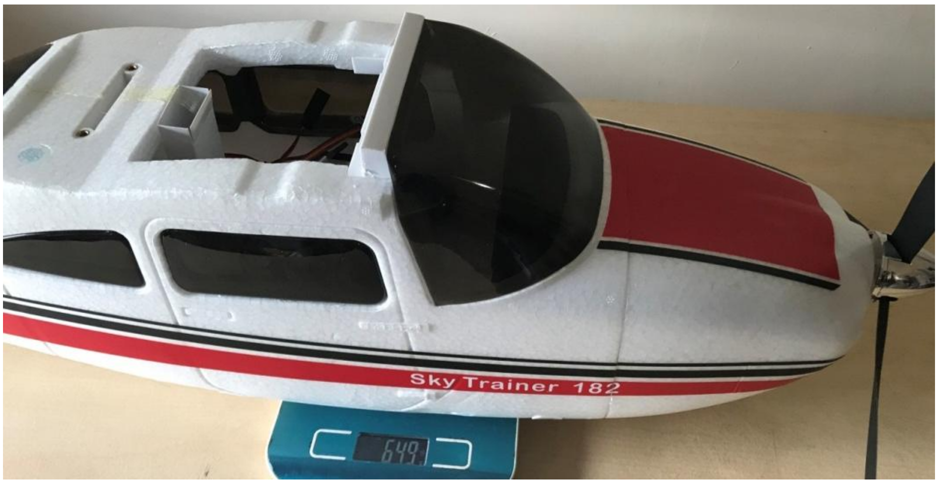


Figure 7.18 Weighing the fuselage on a digital scale

The fuselage was weighed with the motor (including mount), the propeller (including mount) + radio receiver and 2 internal servos for the elevator and rudder. The total weight was 649 grams. The CG of the fuselage in the absence of these extra modules was measured to be 316.75 mm from the nose by balancing the fuselage on a tube until horizontal and stable as shown in the following sections. Because the nose offset is 204 mm, the CG is located at 112.75 mm aft of the wing reference point (see XFLR5/X-Plane methods). In the vertical axis, it coincides with the propeller axis. This alignment is assumed as it cannot be measured easily.

7.5.2.4 Motor (including mount):

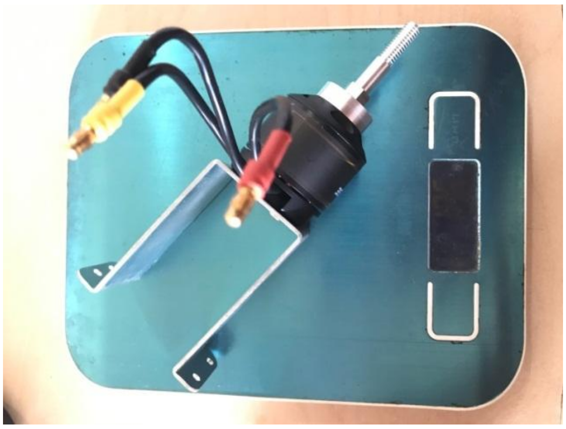


Figure 7.19 Weighing the Brushless DC Motor on a digital scale

The motor (including the mount) was weighed using a digital scale at 186 grams. The CG of the motor (including mount has been measured at 45 mm from the nose of the fuselage). That position is 204 mm ahead of the wing reference point (see XFLR5/X-Plane methods [3][4]). Which makes ita total offset of 159mm ahead. For the vertical axis, it aligns with the propeller axis.

7.5.2.5 Radio Receiver:

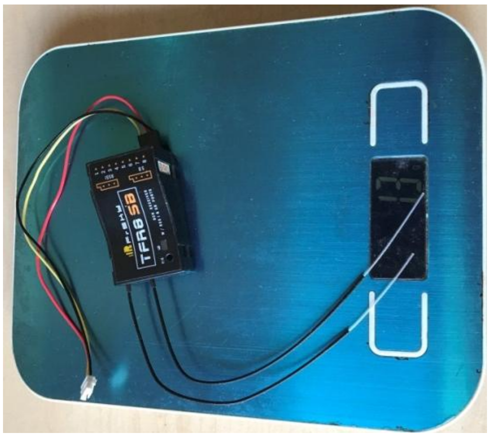


Figure 7.20 Weighing the RC receiver on a digital scale

The radio receiver was weighed using a digital scale at 16 grams. Its position is to be determined based on onboard avionics placements but the main unit is planned on being situated as close to the CG of the aircraft as possible.

7.5.2.6 Propeller (including mount)

Figure 7.21 Weighing the propeller on a digital scale

The propeller (including the mount) was weighed using a digital scale at 36 grams.

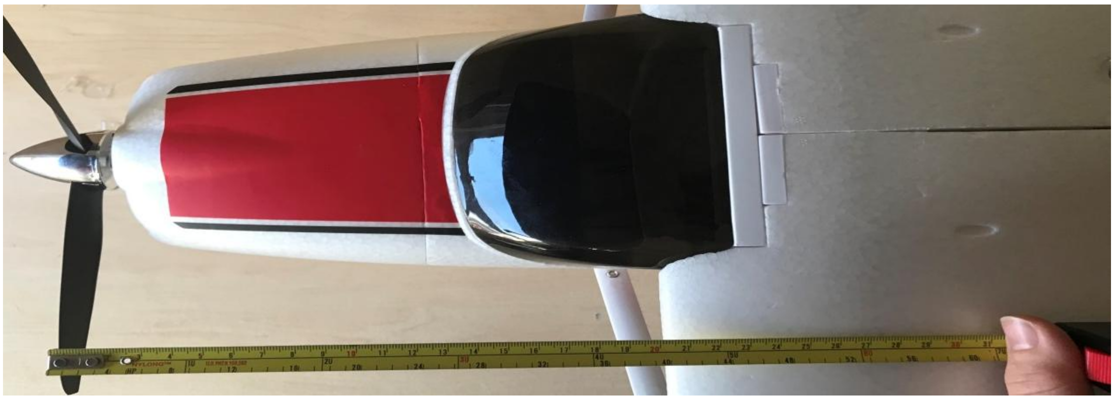


Figure 7.22 Measuring propeller distance from the wing tip

It's position has been measured to be 15 mm from the nose. It is positioned along the x-axis. It is situated 219 mm ahead of the wing tip and 75 mm below it (propeller axis).

7.5.2.7 Landing Gears:

The three landing gears were found to have different masses. As such they are subdivided into the following three components.

7.5.2.7.1 Left landing gear (port):

The left landing gear was weighed on a digital scale and the reading recorded as 47 grams.

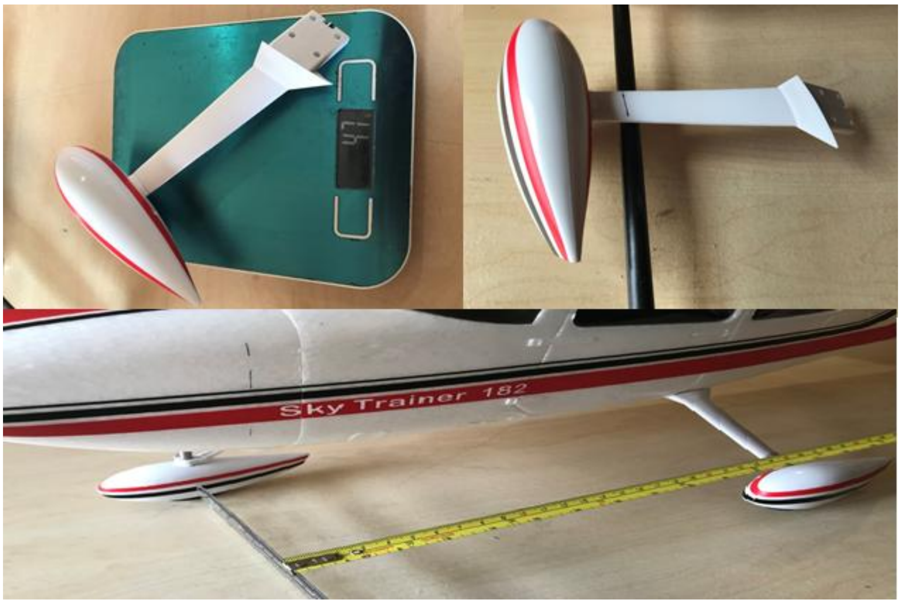


Figure 7.23 Weighing, localising and determining the C.G. of the gear

It's approximate CG was estimated by balancing it on a carbon fibre tube and marking the position with a marker. The position was adjusted until the landing gear was horizontal and stable. This position relative to appropriate reference frames were measured more precisely in the complete assembly as illustrated in the figure. It is noteworthy to mention that XFLR5 uses the leading edge of the wing and X-Plane uses the nose of the default aircraft as their respective reference points.

7.5.2.7.2 Right Landing gear (starboard):

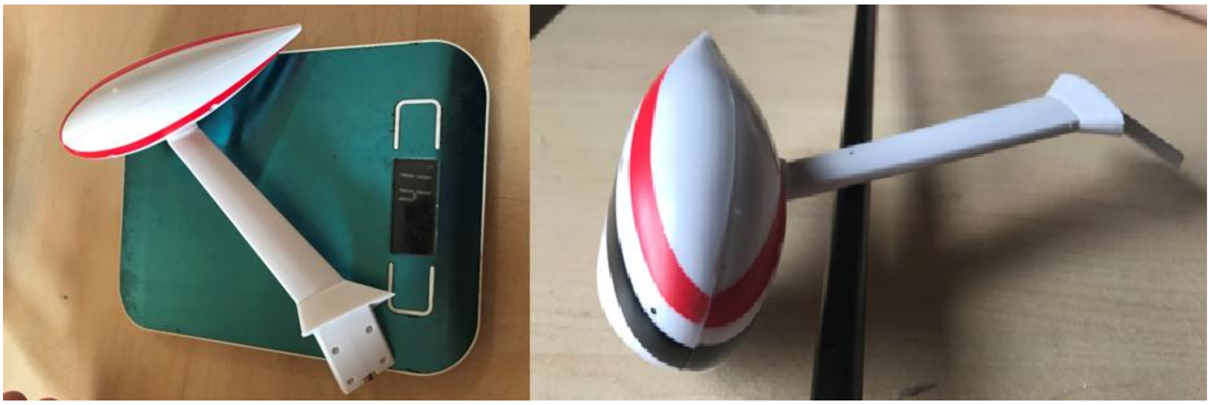


Figure 7.24 Weighing and balancing the right landing gear for C.G. determination

The right landing gear was weighed on a digital scale and the weight was recorded as 41 grams (note: different from the left landing gear). The same technique was used to get an approximation of the CG location via balancing it on a carbon fibre tube until it's stable and horizontal. The CG position relative to appropriate frames of references were noted using the same method as shown for the left landing gear.

The CG of the right landing gear was determined to be 125 mm, 142.5 mm from the centreline and 200mm vertically down from the wing chord.

7.5.2.7.3 Front Landing Gear:

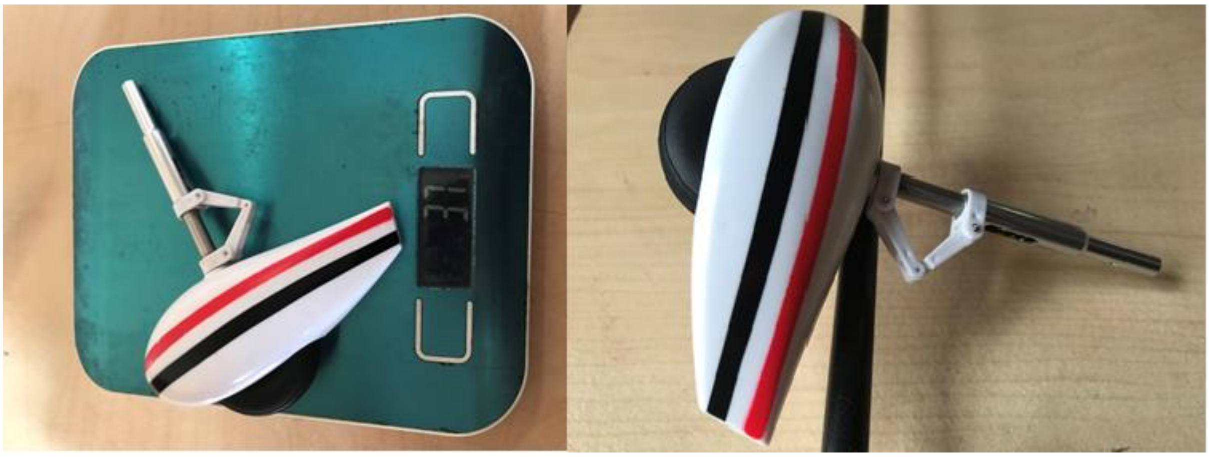


Figure 7.25 Weighing and determining the C.G. of the front landing gear

The front landing gear was weighed on a digital scale and recorded as 37 grams. The same balancing technique was utilised to find the CG of this gear. There were difficulties in finding the exact spot but it is so close to the point shown in the above image that it was just assumed to be there. The minute difference in the actual CG position is not going to impact any of the modelling processes or results. It's position relative to the wing was determined by marking a vertical line on the side of the fuselage (coincides with its axis) and measuring the distance of that line from the wing tip as shown in the image.

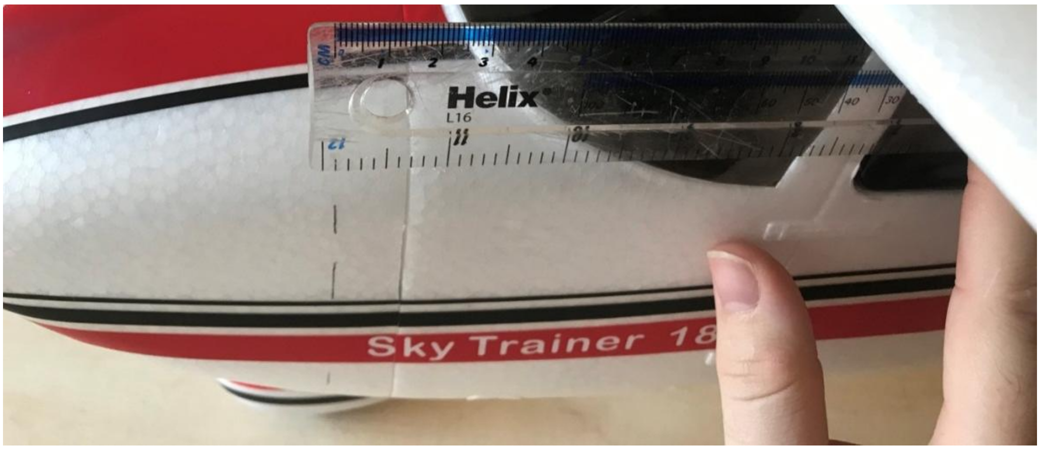


Figure 7.26 Measuring front landing gear location relative to the wing tip

Even though the nose wheel is in line with the centre axis of the aircraft, it's centre of mass is still an important consideration for the modelling because it is offset from the aircraft's roll, pitch and yaw axis. This translates to moment contribution to all 3 moments of inertia of the overall aircraft. The location of the front landing gear CG was determined to be 100mm ahead of the wing, 200 mm below the wing and in line with the root chord of the wing.

7.5.2.8 Wing struts:

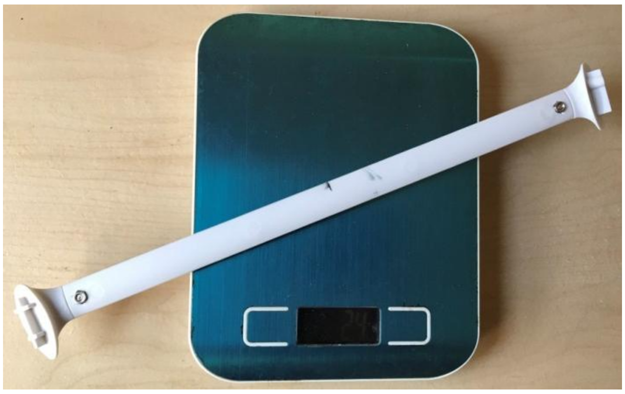


Figure 7.27 Weighing the wing strut

The wing struts are uniform and identical and their CGs are marked (as shown in the image). They were weighed at 25 grams each (with their attachments included as shown in the image). Their marked locations were used to find their COG location with respect to desirable coordinate frames. From the wing reference, the left strut is located at 25 mm from the leading edge, 190 mm from the centreline and 77.5 mm vertically down. The right strut is located 25 mm from the leading edge, -190 mm from the centreline and 77.5 mm vertically down.

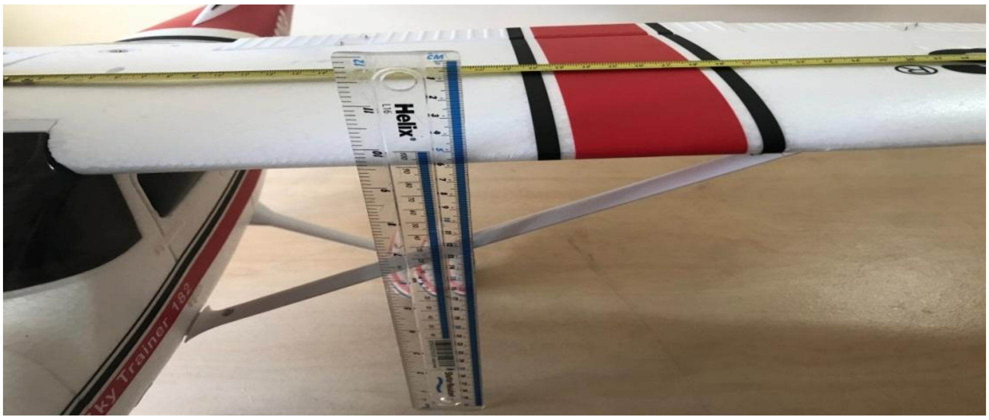
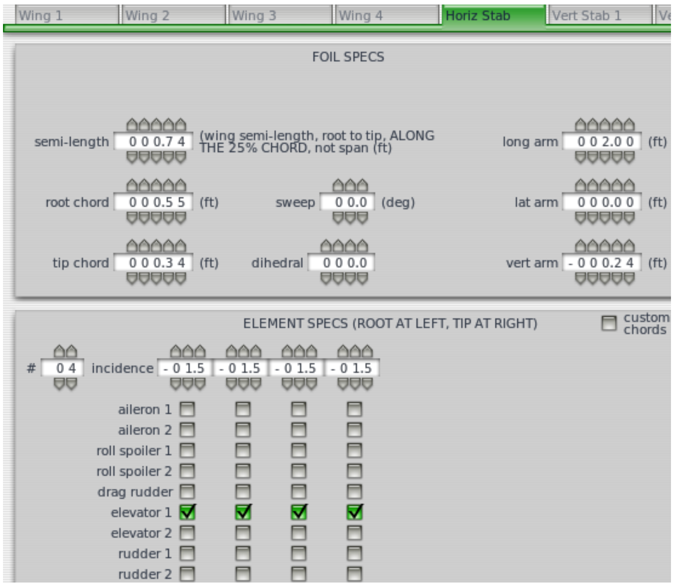


Figure 7.28 Measuring wing strut C.G. position relative to the reference points

7.5.3 Horizontal Stabiliser modelling process



FOIL SPECS

semi-length 0 0 0.7 4 (wing semi-length, root to tip, ALONG THE 25% CHORD, not span (ft)

long arm 0 0 2.0 0 (ft)

root chord 0 0 0.5 5 (ft)

sweep 0 0.0 (deg)

lat arm 0 0 0.0 0 (ft)

tip chord 0 0 0.3 4 (ft)

dihedral 0 0 0.0

vert arm - 0 0 0.2 4 (ft)

ELEMENT SPECS (ROOT AT LEFT, TIP AT RIGHT)

□ custom chords

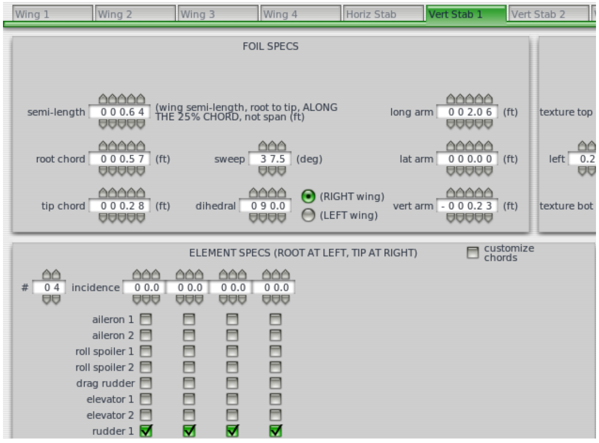
0 4

incidence	- 0 1.5	- 0 1.5	- 0 1.5	- 0 1.5
aileron 1	□	□	□	□
aileron 2	□	□	□	□
roll spoiler 1	□	□	□	□
roll spoiler 2	□	□	□	□
drag rudder	□	□	□	□
elevator 1	☑	☑	☑	☑
elevator 2	□	□	□	□
rudder 1	□	□	□	□
rudder 2	□	□	□	□

Figure 7.29 Horizontal stabiliser settings in Plane Maker for the FMS 182 model

For the creation of the Horizontal Stabiliser, the same procedure was used. However, instead of picking Misc Wings, the default wing setup menus were used and the appropriate measured values entered as shown in the above image. ELEMENT SPECS were used to assign control surfaces (elevators) to the section.

7.5.4 Vertical Stabiliser modelling process



FOIL SPECS

semi-length 0 0 0.6 4 (wing semi-length, root to tip, ALONG THE 25% CHORD, not span (ft)

long arm 0 0 2.0 6 (ft)

texture top

root chord 0 0 0.5 7 (ft)

sweep 3 7.5 (deg)

lat arm 0 0 0.0 0 (ft)

left 0.2

tip chord 0 0 0.2 8 (ft)

dihedral 0 9 0.0

● (RIGHT wing)

○ (LEFT wing)

vert arm - 0 0 0.2 3 (ft)

texture bot

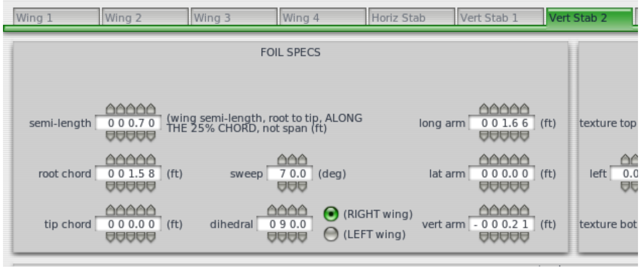
ELEMENT SPECS (ROOT AT LEFT, TIP AT RIGHT)

□ customize chords

0 4

incidence	0 0.0	0 0.0	0 0.0	0 0.0
aileron 1	□	□	□	□
aileron 2	□	□	□	□
roll spoiler 1	□	□	□	□
roll spoiler 2	□	□	□	□
drag rudder	□	□	□	□
elevator 1	□	□	□	□
elevator 2	□	□	□	□
rudder 1	☑	☑	☑	☑

Figure 7.30 Vertical stabiliser settings in Plane Maker for the FMS 182 model



FOIL SPECS

semi-length 0 0 0.7 0 (wing semi-length, root to tip, ALONG THE 25% CHORD, not span (ft)

long arm 0 0 1.6 6 (ft)

texture top

root chord 0 0 1.5 8 (ft)

sweep 7 0.0 (deg)

lat arm 0 0 0.0 0 (ft)

left 0.0

tip chord 0 0 0.0 0 (ft)

dihedral 0 9 0.0

● (RIGHT wing)

○ (LEFT wing)

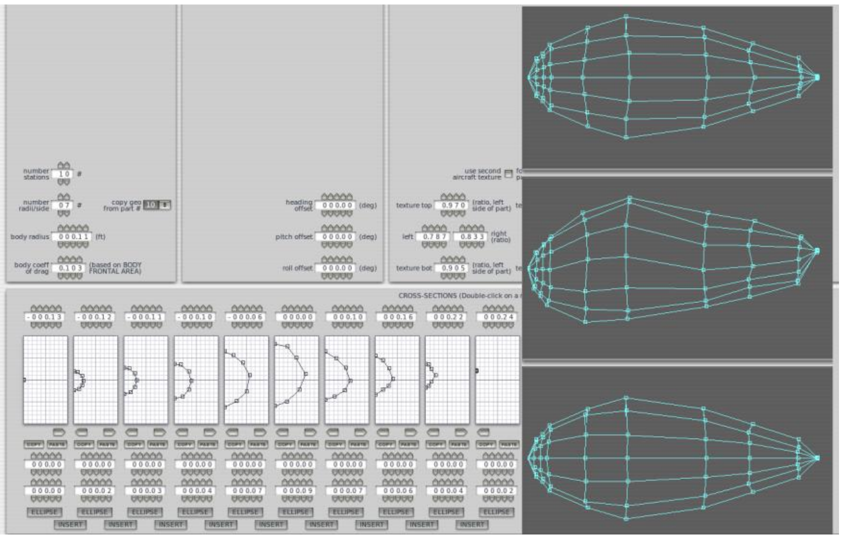
vert arm - 0 0 0.2 1 (ft)

texture bot

Figure 7.31 Vertical stabiliser 2 settings

The vertical stabiliser was divided into two sections as explained earlier in the target airframe measurements documentation process (Chapter 3.1). Based on the previously recorded measurements, these sections were defined in the same section of Plane Maker where the Horizontal Stabiliser was defined following identical procedures including assignment of the control surfaces (rudder) to Vert Stab 1.

7.5.5 Wheel Fairings Modelling



number stations 10 #

number radii/side 07 #

copy geo from part # 10

body radius 000.11 (ft)

body coeff of drag 0.103 (based on BODY FRONTAL AREA)

heading offset 000.00 (deg)

pitch offset 000.00 (deg)

roll offset 000.00 (deg)

use second aircraft texture □

texture top 0.970 (ratio, left side of part)

left 0.787 0.833 right (ratio)

texture bot 0.905 (ratio, left side of part)

CROSS-SECTIONS (Double-click on a

-000.13 -000.12 -000.11 -000.10 -000.06 000.00 000.10 000.16 000.22 000.24

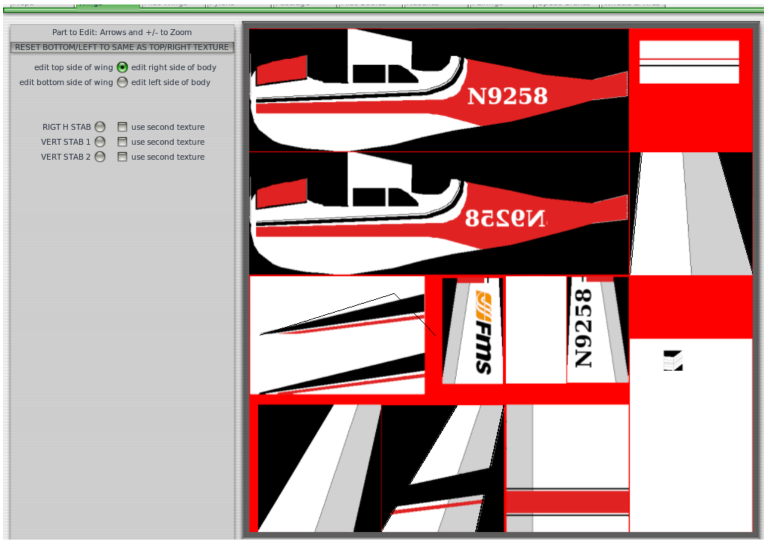
000.00 000.00 000.00 000.00 000.00 000.00 000.00 000.00 000.00 000.00

000.00 000.02 000.03 000.04 000.07 000.09 000.07 000.06 000.04 000.02

Figure 7.32 Wheel fairings modelling for the FMS 182 in Plane Maker

The wheel fairings were also modelled in by following the same method utilised by the fuselage and spinner modelling. The same attempt was made for the wing struts but unfortunately a bug in Plane Maker would not allow for that. As such, they were created using Misc Wing 1 and Misc Wing 2 with symmetrical aerofoils and the rest of the process was similar to wing modelling except that offsets were needed to place them in the angular fashion they are attached to the aircraft.

7.5.6 Texture Modelling



Part to Edit: Arrows and +/- to Zoom

RESET BOTTOM/LEFT TO SAME AS TOP/RIGHT TEXTURE

edit top side of wing ● edit right side of body

edit bottom side of wing ○ edit left side of body

RIGT H STAB ○ □ use second texture

VERT STAB 1 ○ □ use second texture

VERT STAB 2 ○ □ use second texture

Figure 7.33 Texture setting for the FMS 182 model

Once the geometry of the aircraft has been modelled in Plane Maker, Plane Maker can export a texture map for the various components that form the aircraft. The texture file is a Bitmap file with square dimensions. That Bitmap was edited using Gimp image processing tool and a more realistic looking rendering was therefore created.

7.6 Results Extra

The following are supplementary content based on the Results section of this thesis.

7.6.1 Aerofoil Polars

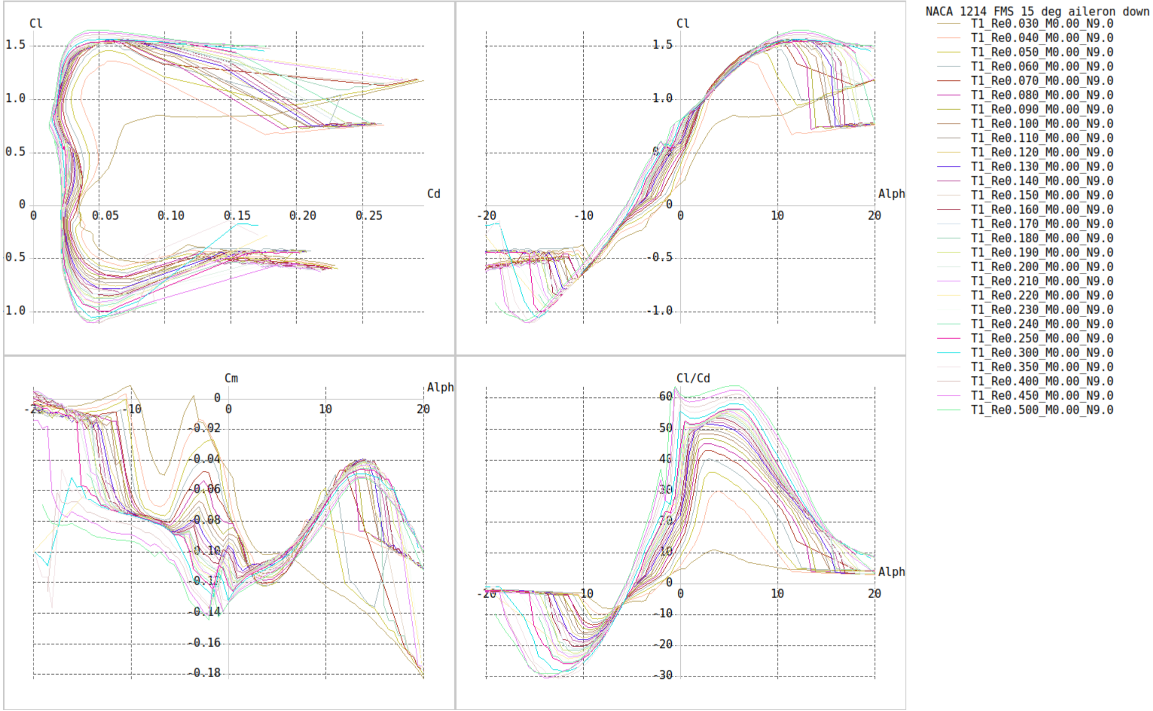


Figure 7.34 NACA 1214 FMS 15 deg aileron down

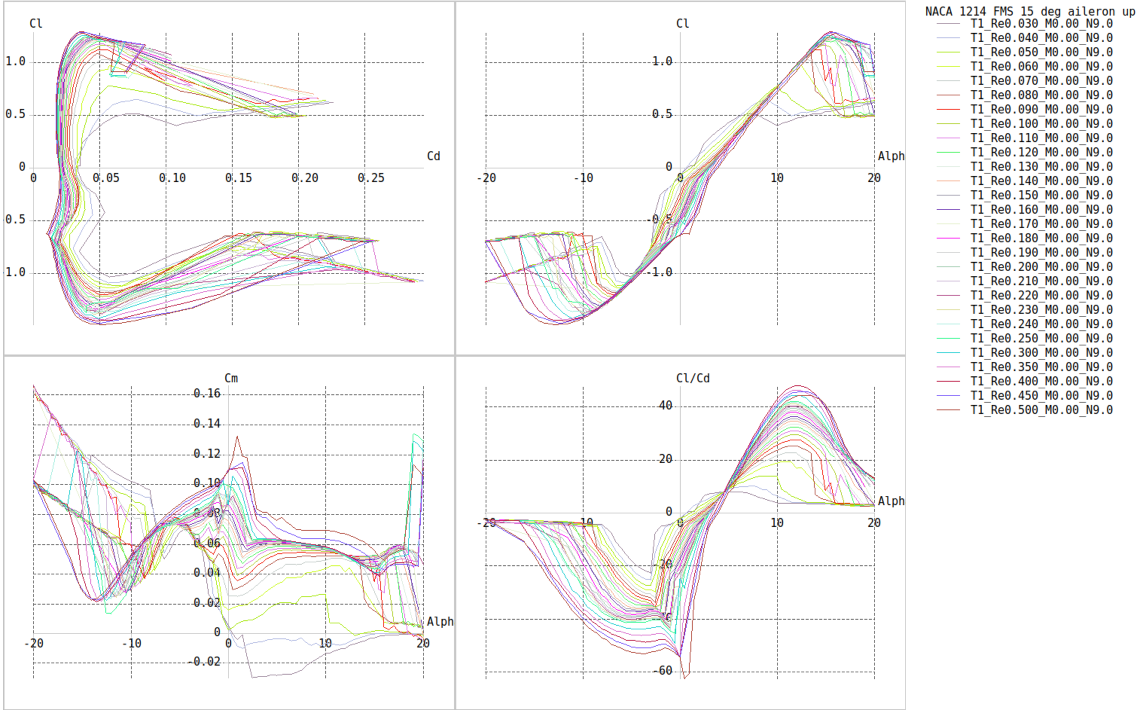


Figure 7.35 NACA 1214 FMS 15 deg aileron up

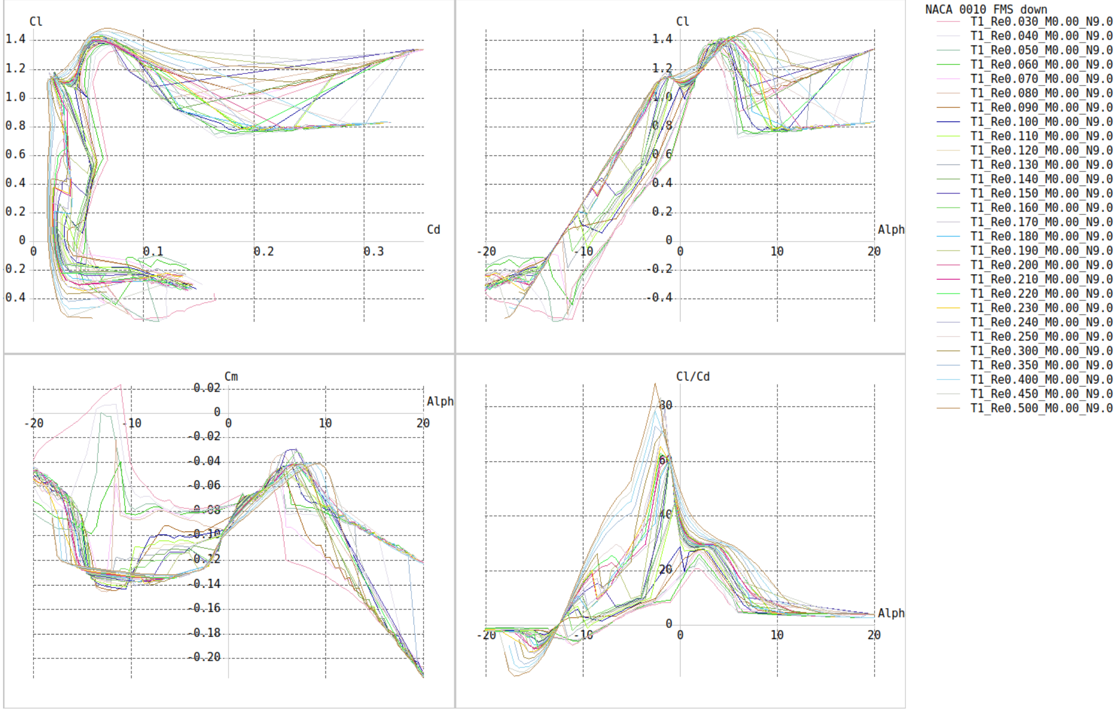


Figure 7.36 NACA 0010 FMS down

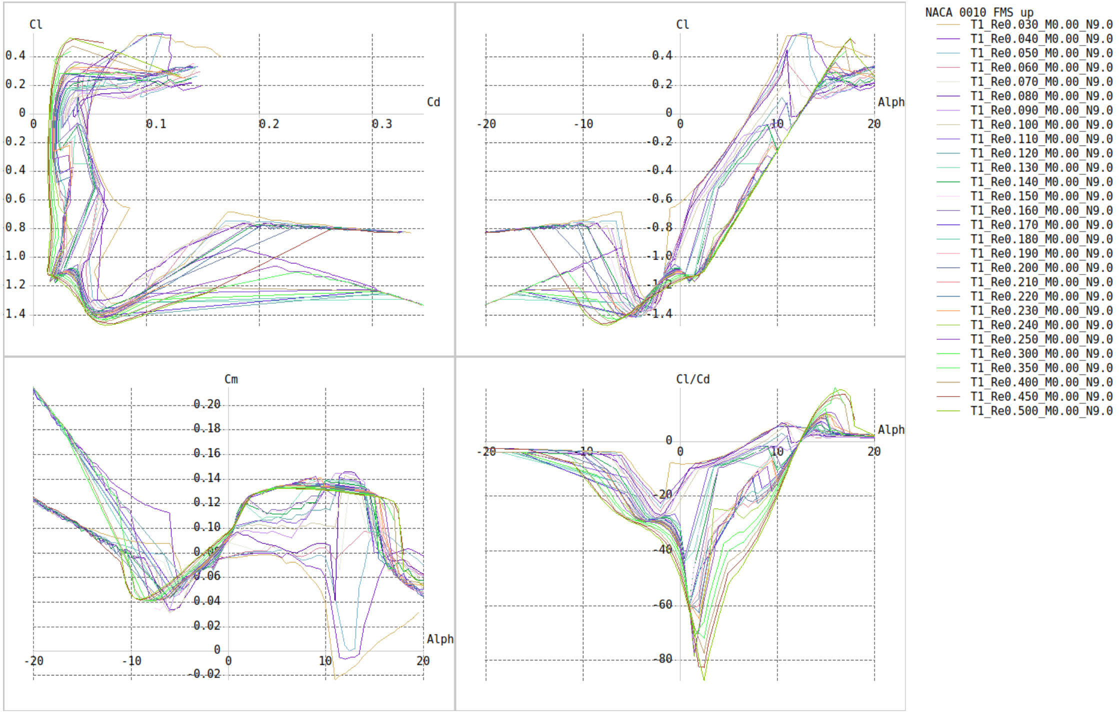


Figure 7.37 NACA 0010 FMS up

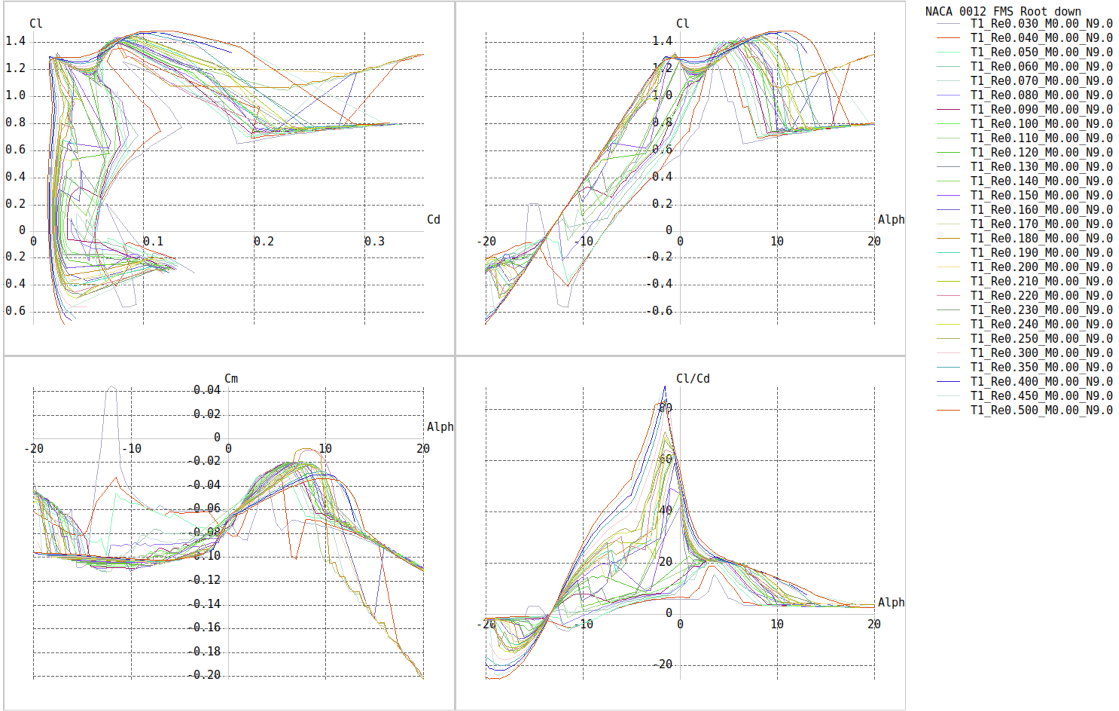


Figure 7.38 NACA 0012 FMS Root down

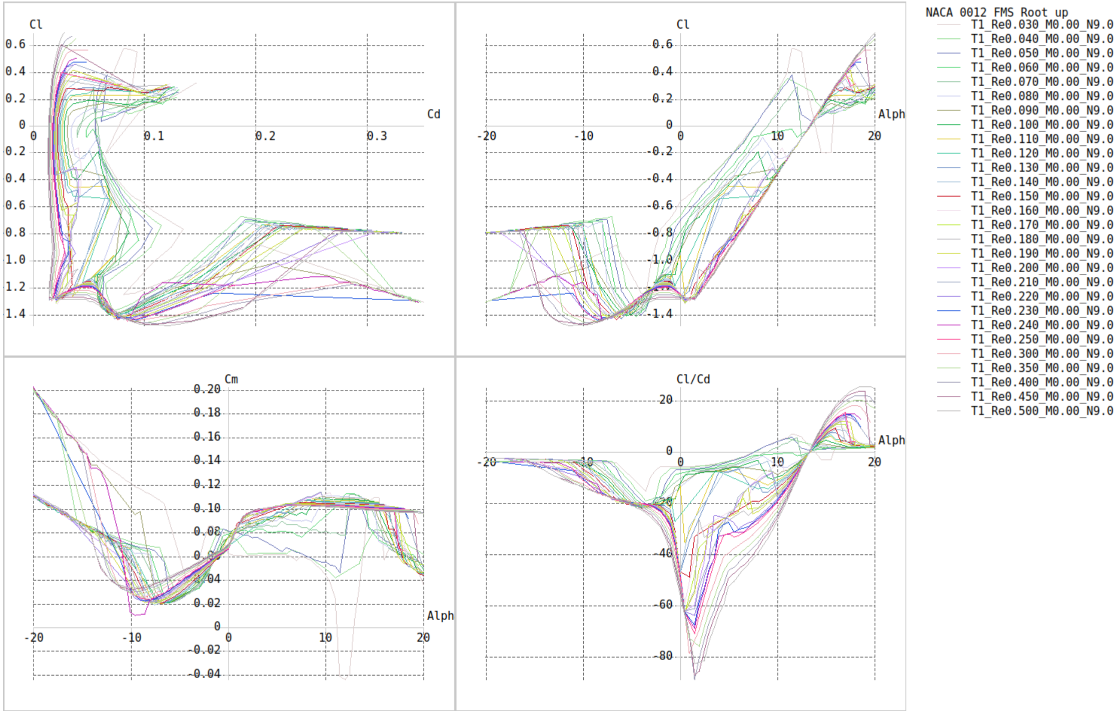


Figure 7.39 NACA 0012 FMS Root up

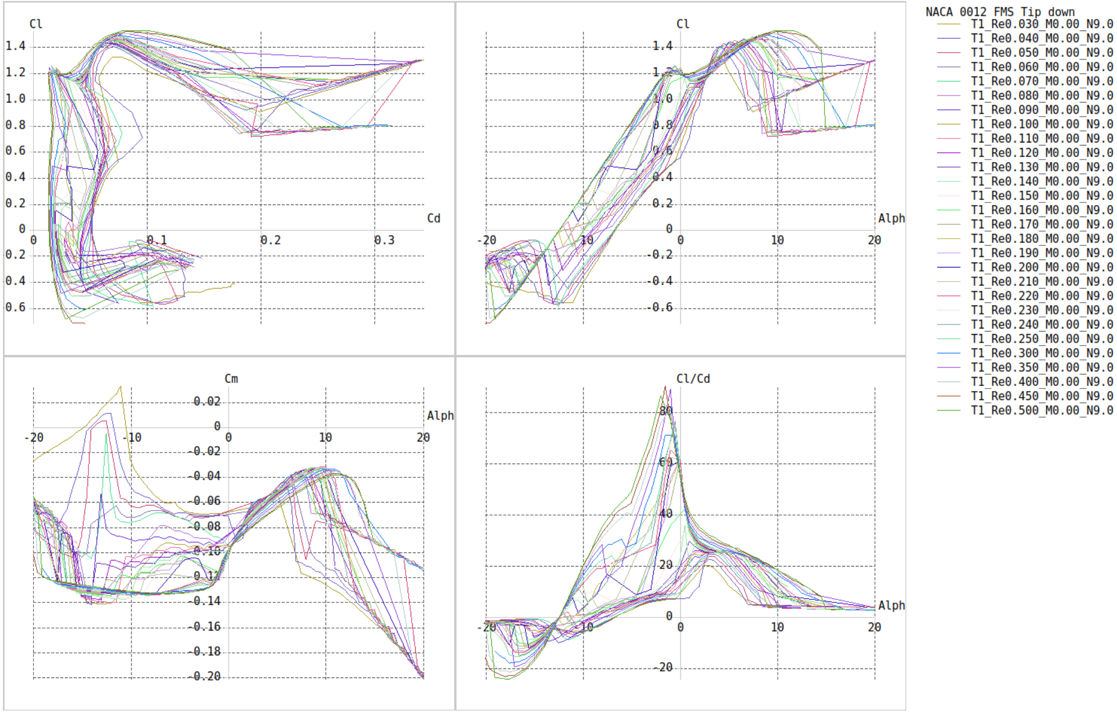


Figure 7.40 NACA 0012 FMS Tip down

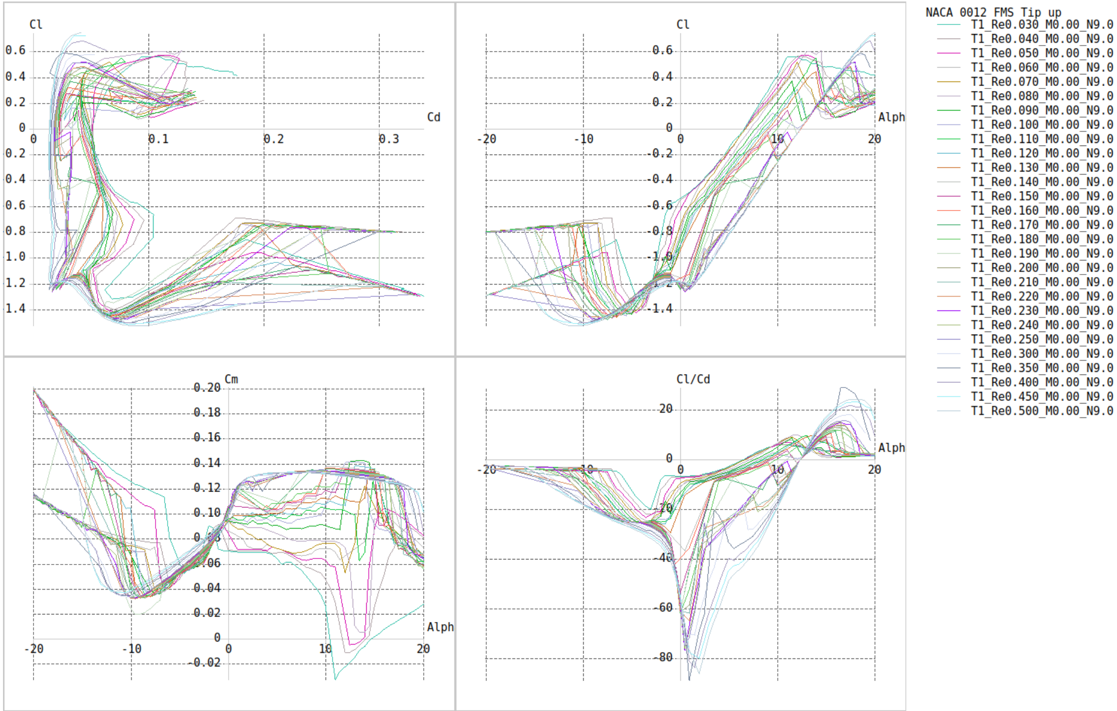


Figure 7.41 NACA 0011 FMS Tip up

7.6.2 Stability Test Flapped Model

The following results have been obtained based on the flapped version of the base model (FMS 182b SC0 Flap).

SPPO:

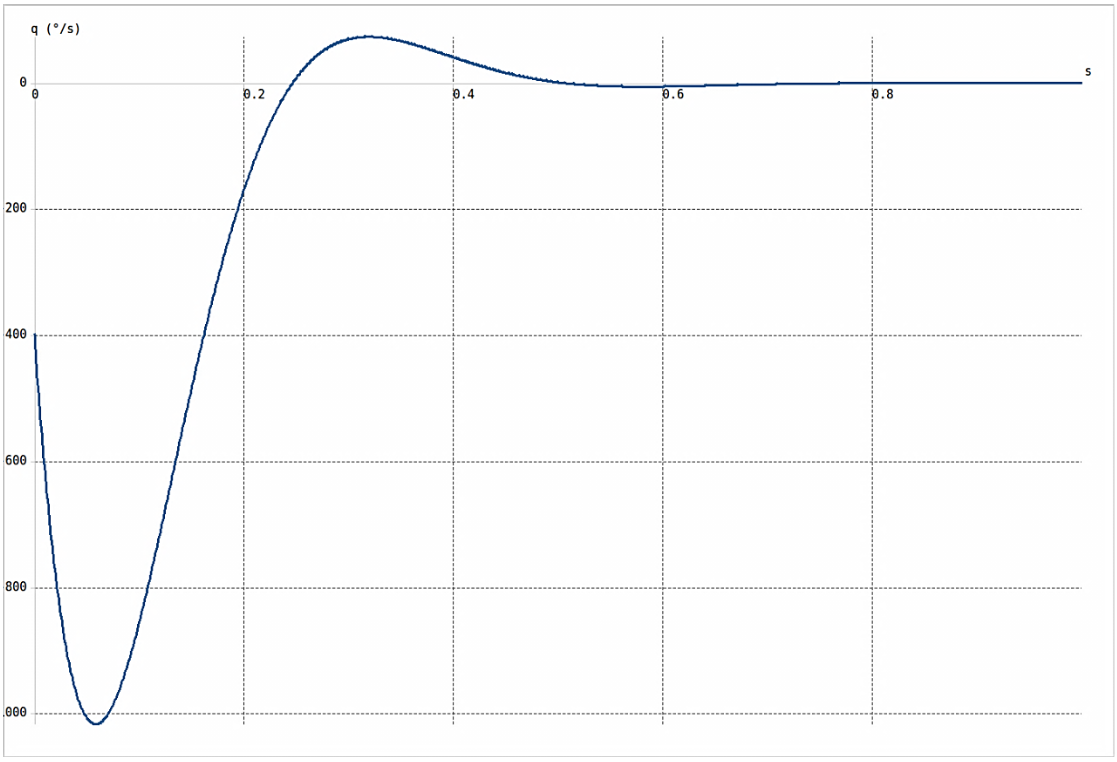


Figure 7.42 Time response: Pitch rate (q) against time (s)

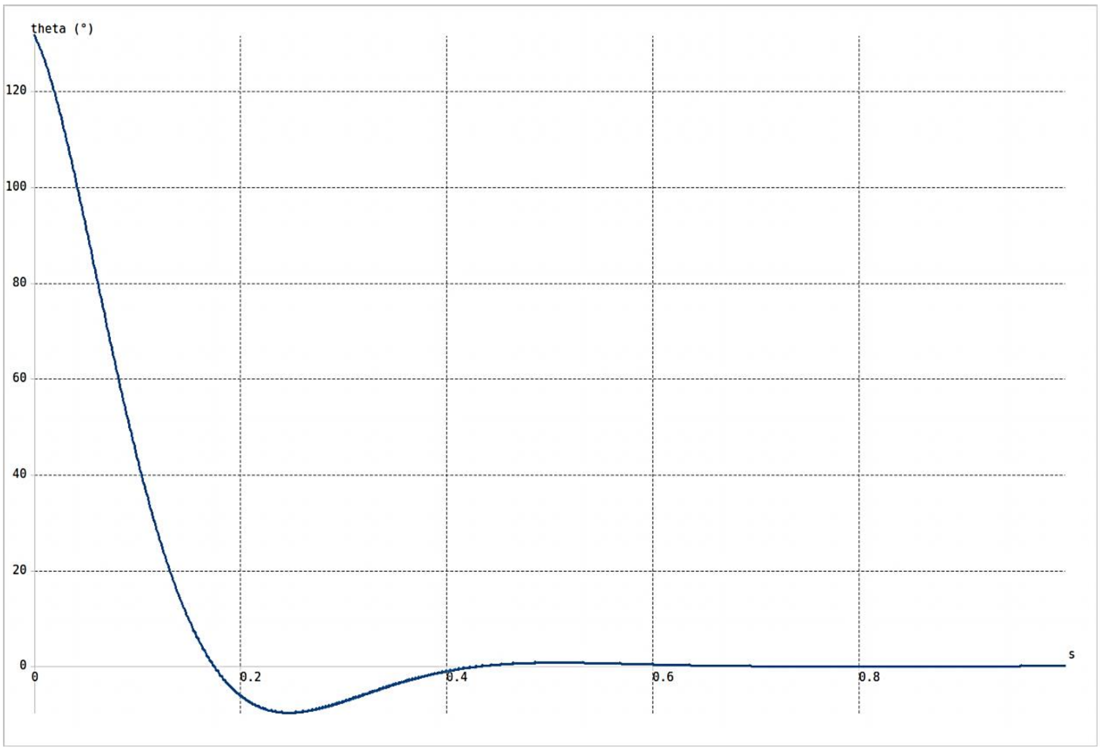


Figure 7.43 Time response: Pitch angle (theta) against time (s)

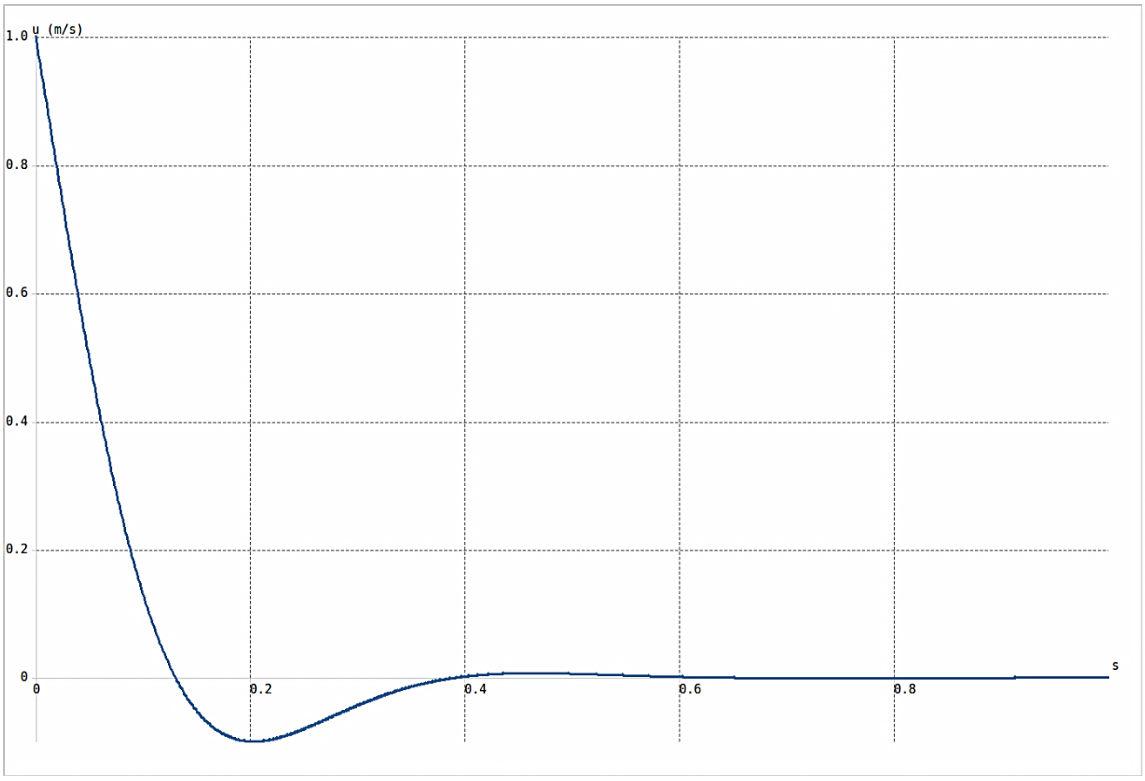


Figure 7.44 Time response: Forward speed (u) against time (s)

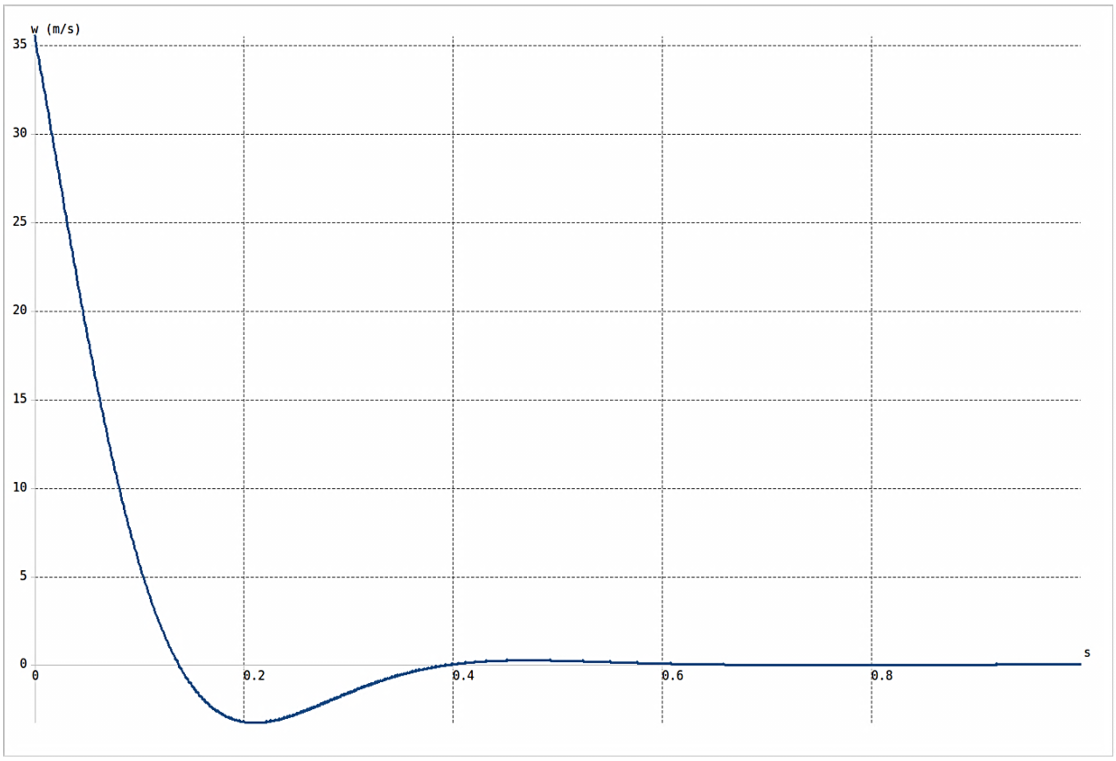


Figure 7.45 Time response: Vertical speed (w) against time (s)

Phugoid:

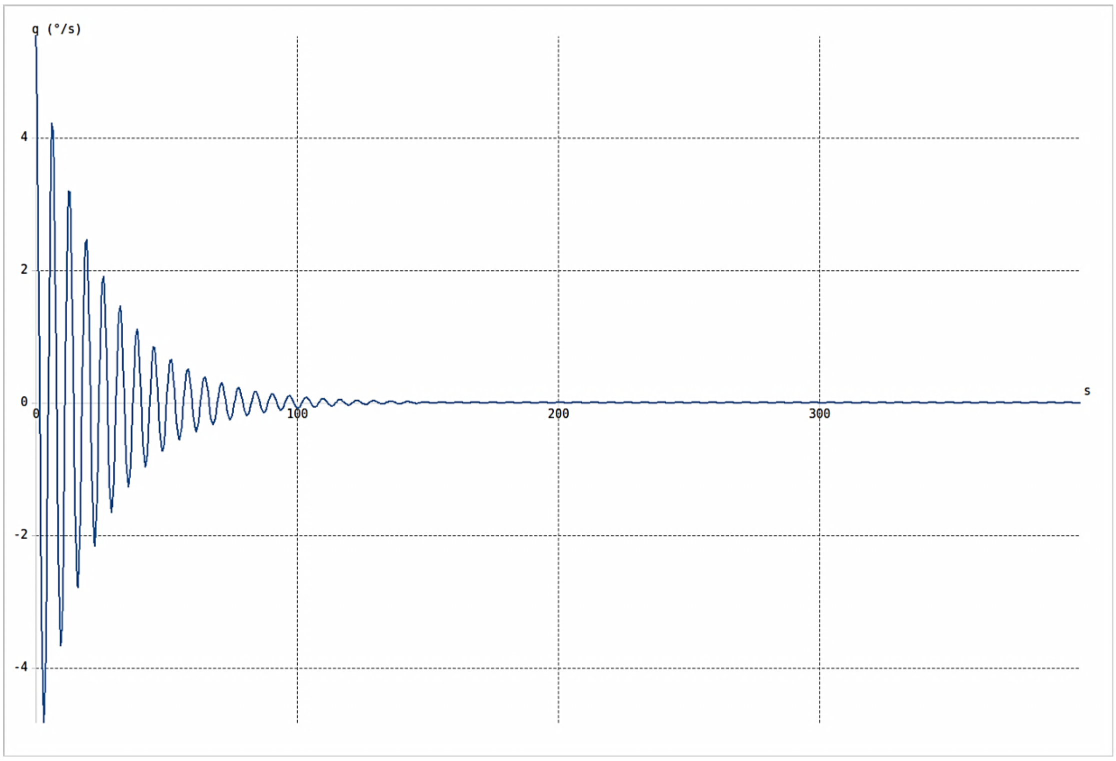


Figure 7.46 Time response: Pitch rate (q) against time (s)

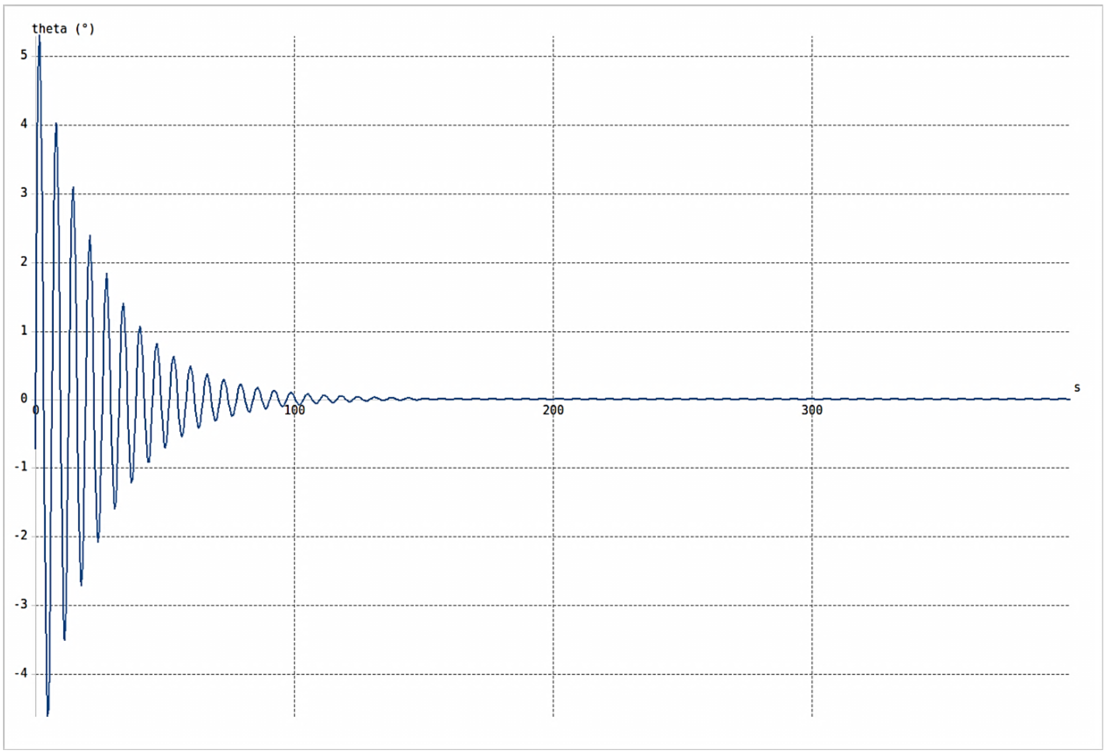


Figure 7.47 Time response: Pitch angle (theta) against time (s)

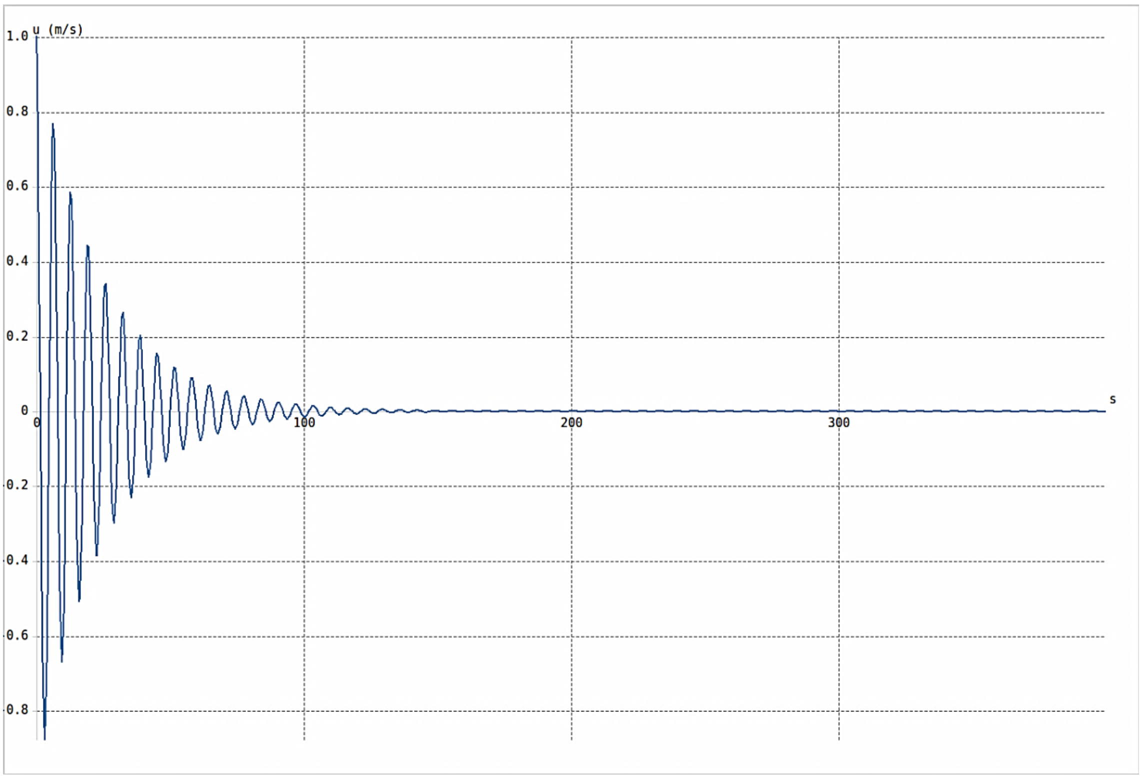


Figure 7.48 Time response: Forward speed (u) against time (s)

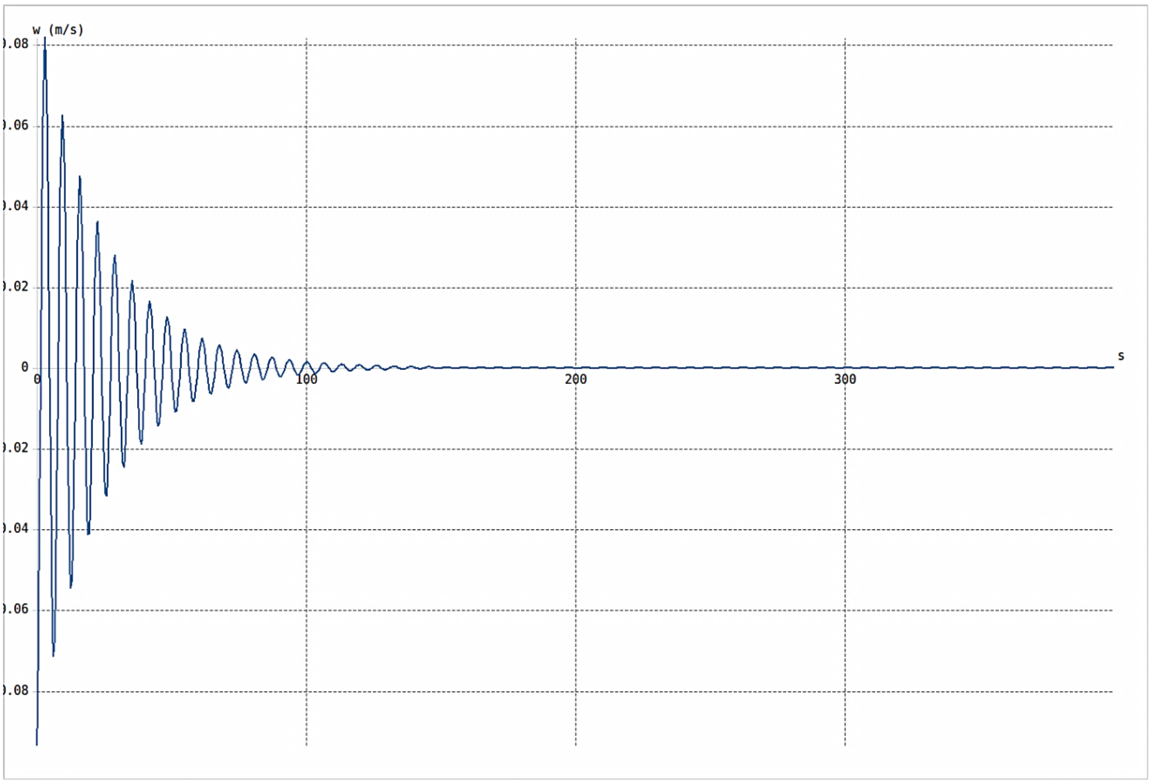


Figure 7.49 Time response: Vertical speed (w) against time (s)

Roll Damping:

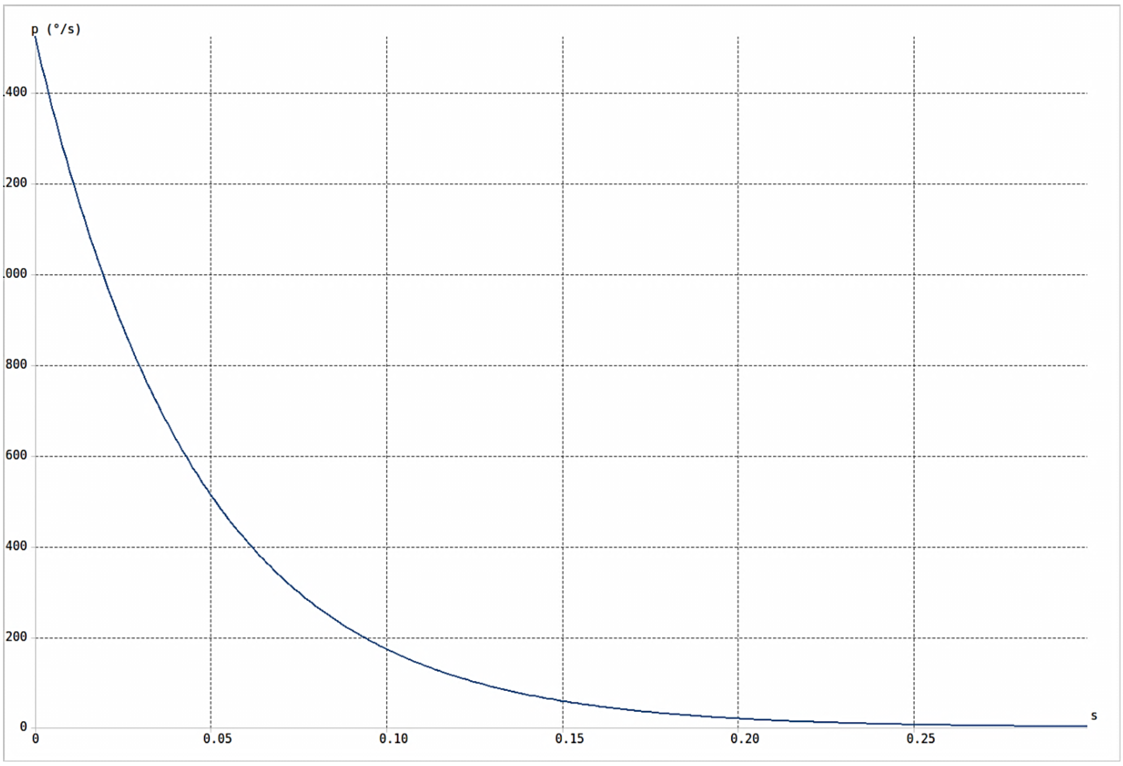


Figure 7.50 Time response: Roll rate (p) against time (s)

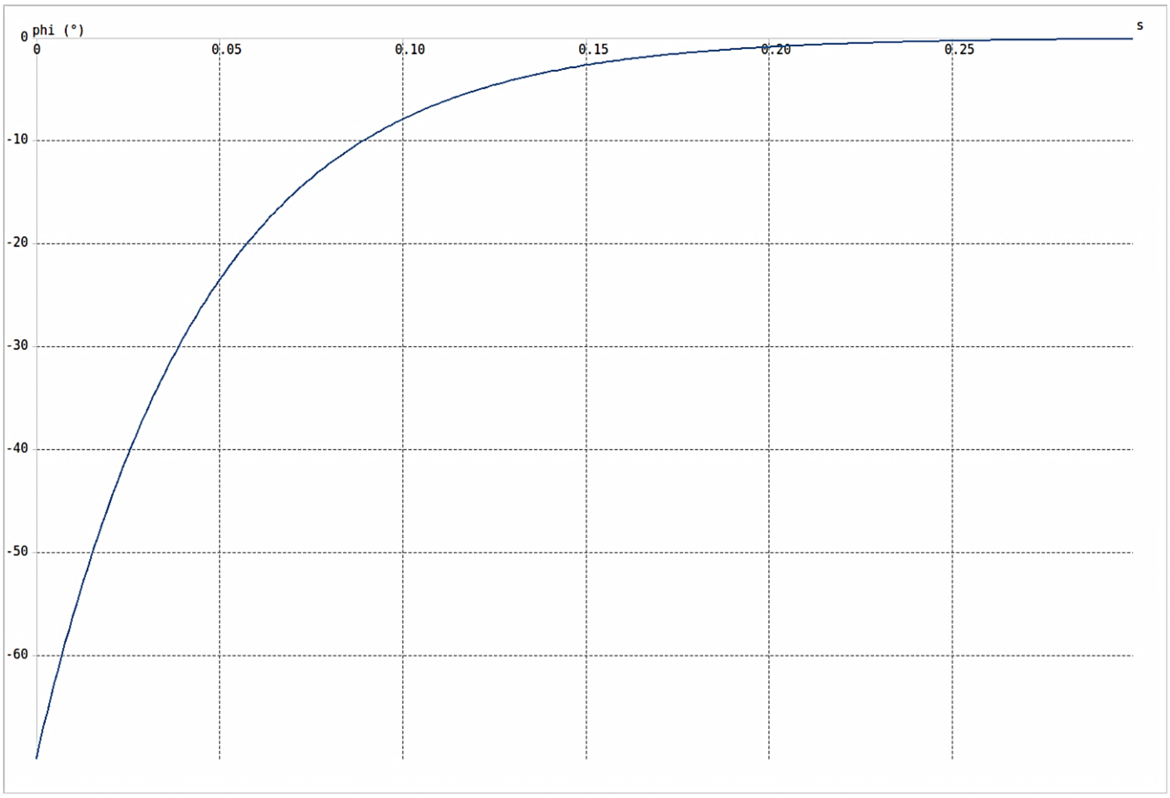


Figure 7.51 Time response: Roll angle (phi) against time (s)

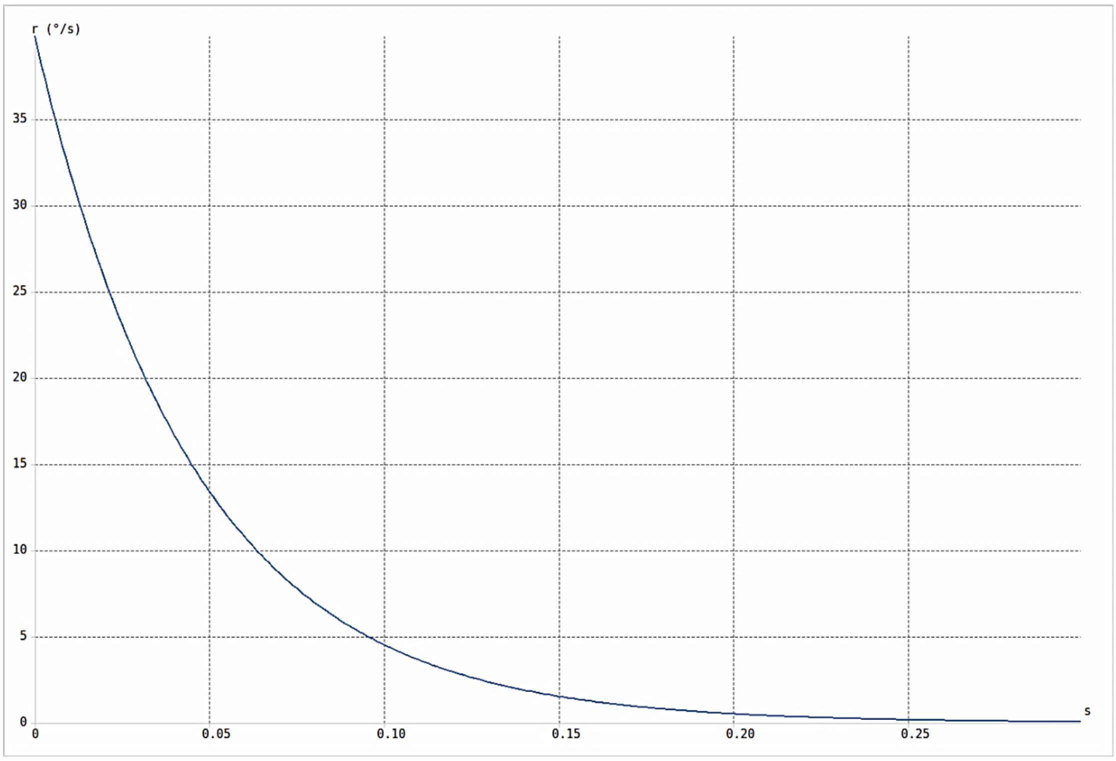


Figure 7.52 Time response: Yaw rate (r) against time (s)

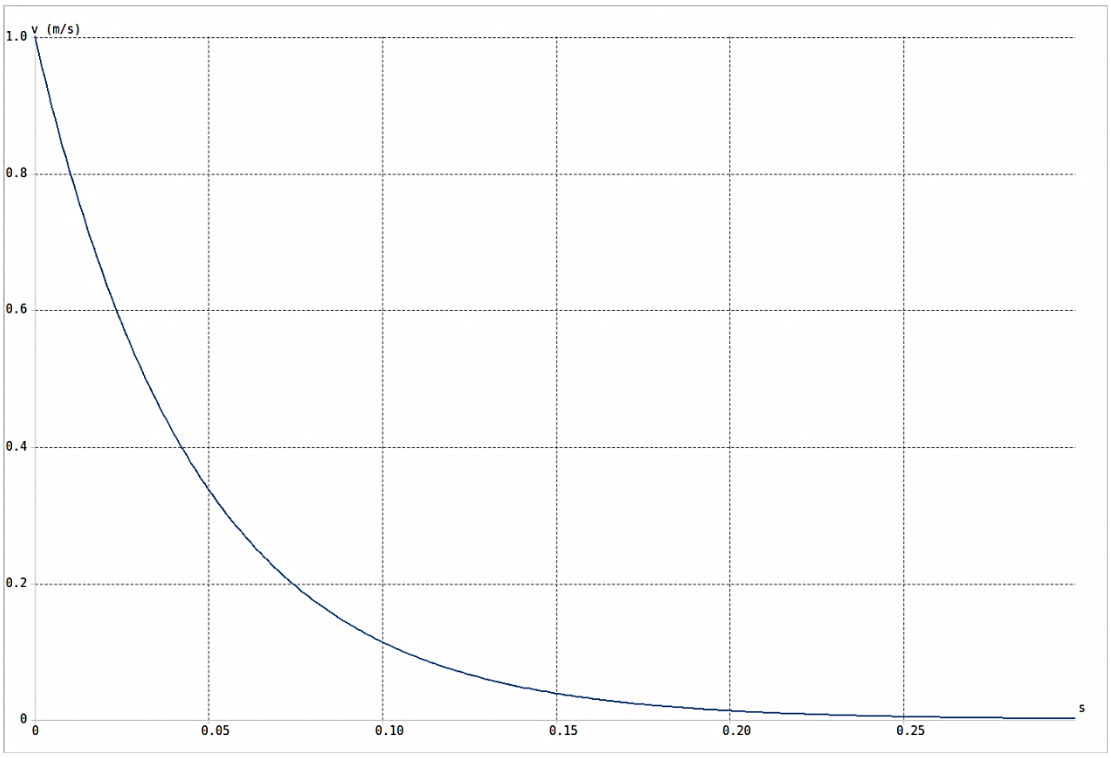


Figure 7.53 Time response: Lateral speed (v) against time (s)

Spiral Mode:

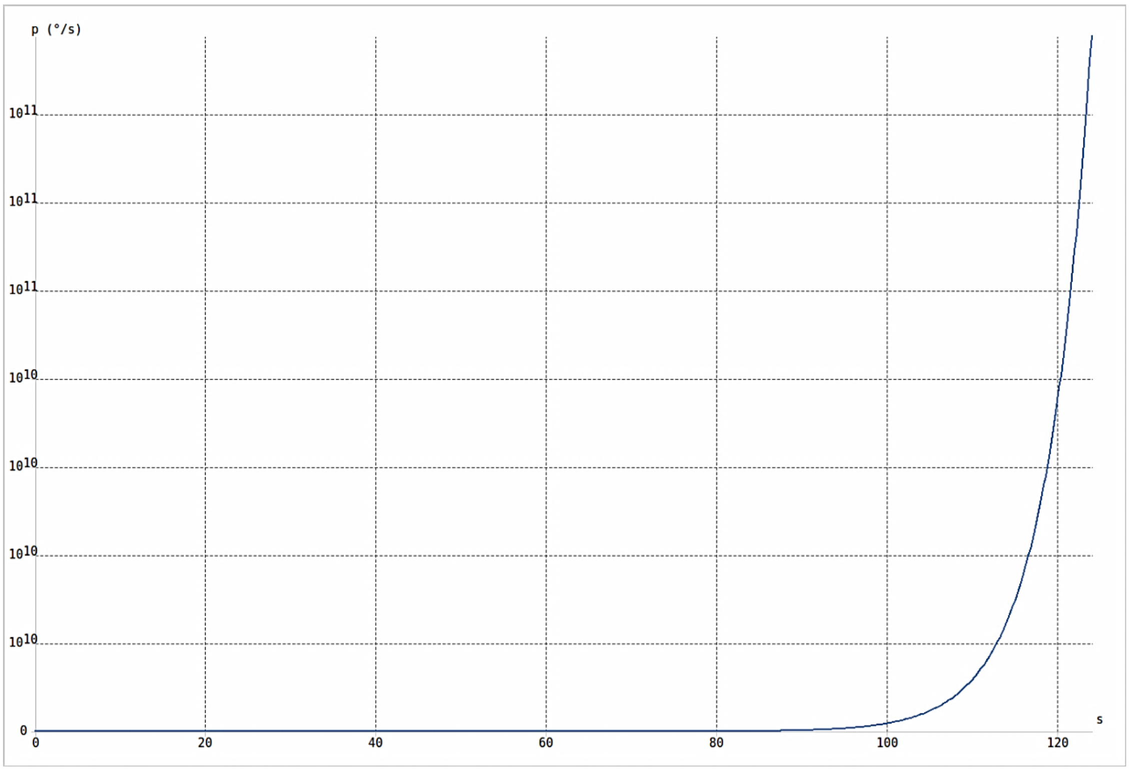


Figure 7.54 Time response: Roll rate (p) against time (s)

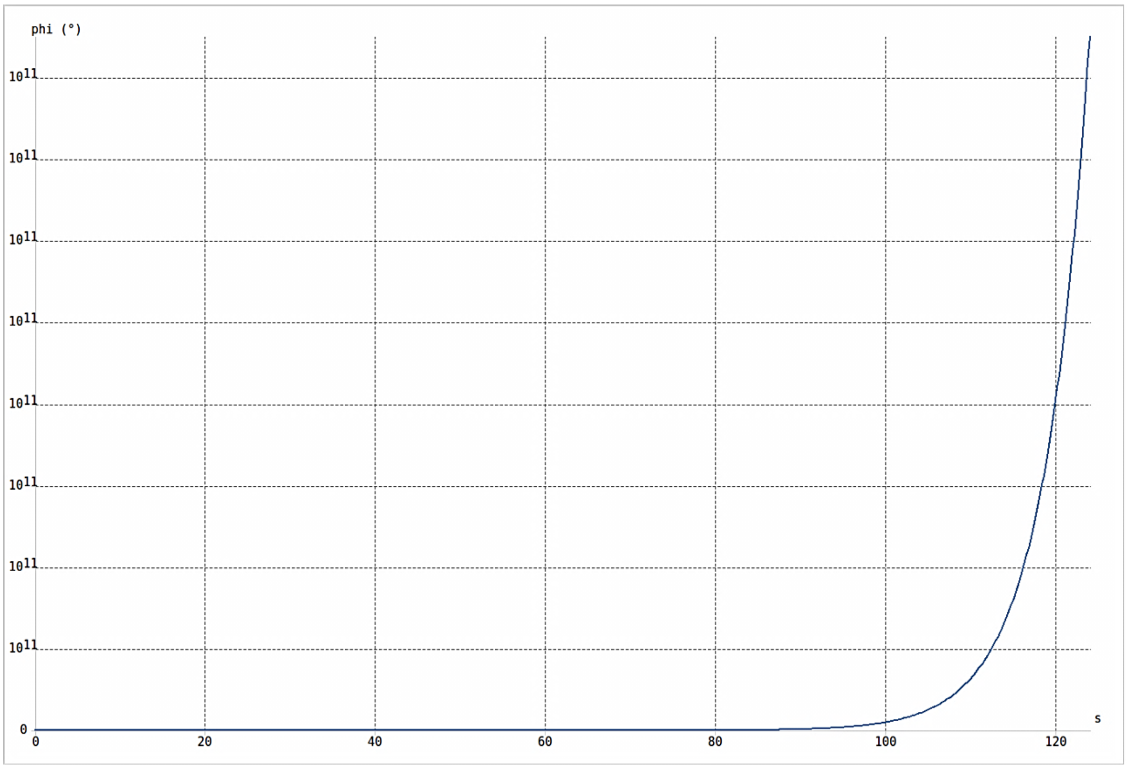


Figure 7.55 Time response: Roll angle (phi) against time

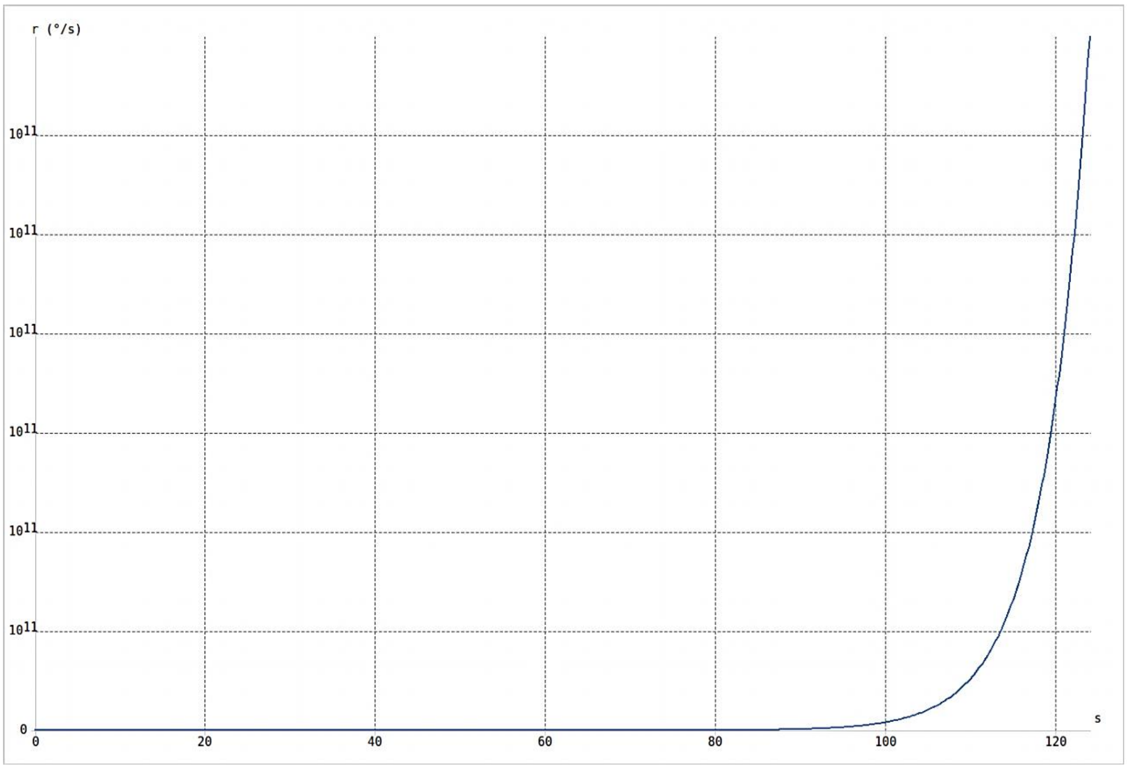


Figure 7.56 Time response: Yaw rate (r) against time (s)

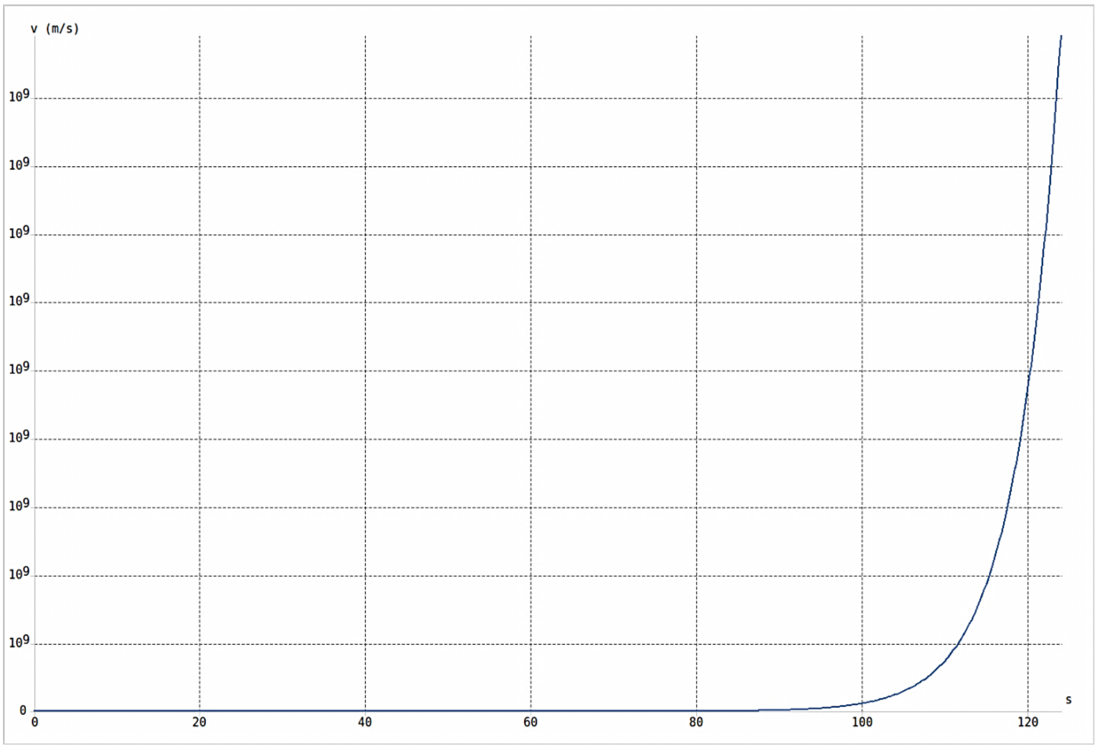


Figure 7.57 Time response: Lateral speed (v) against time (s)

Dutch Roll:

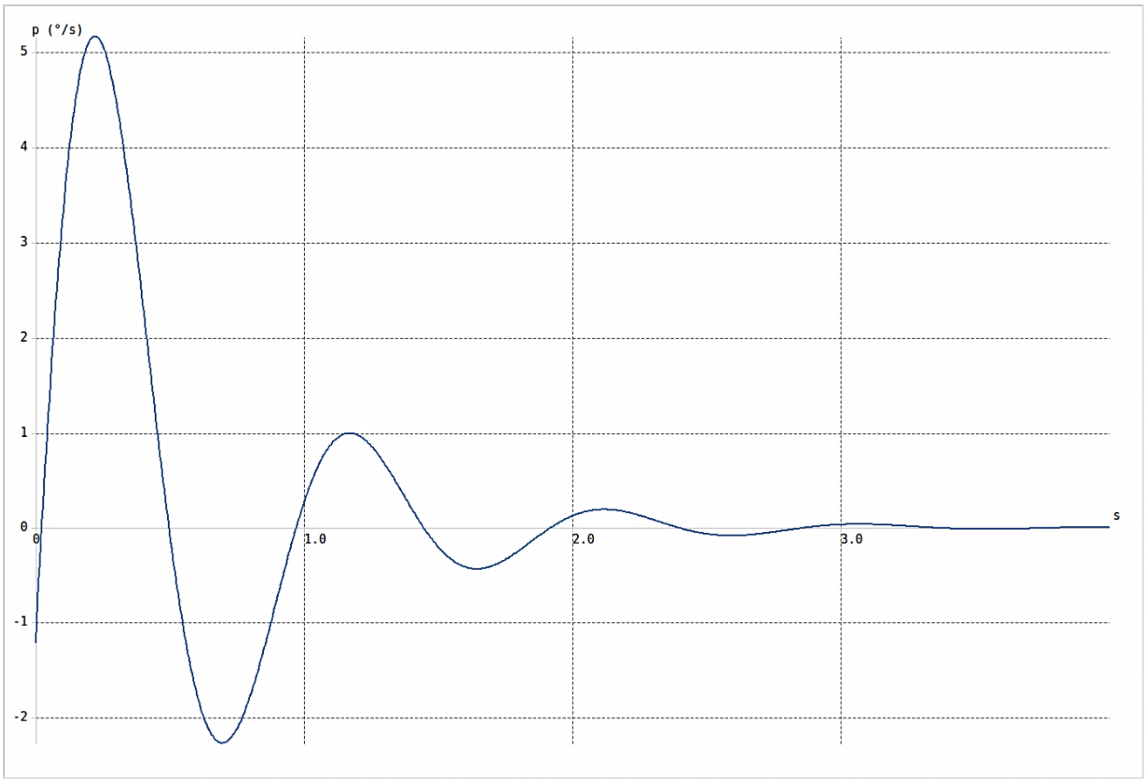


Figure 7.58 Time response: Roll rate (p) against time (s)

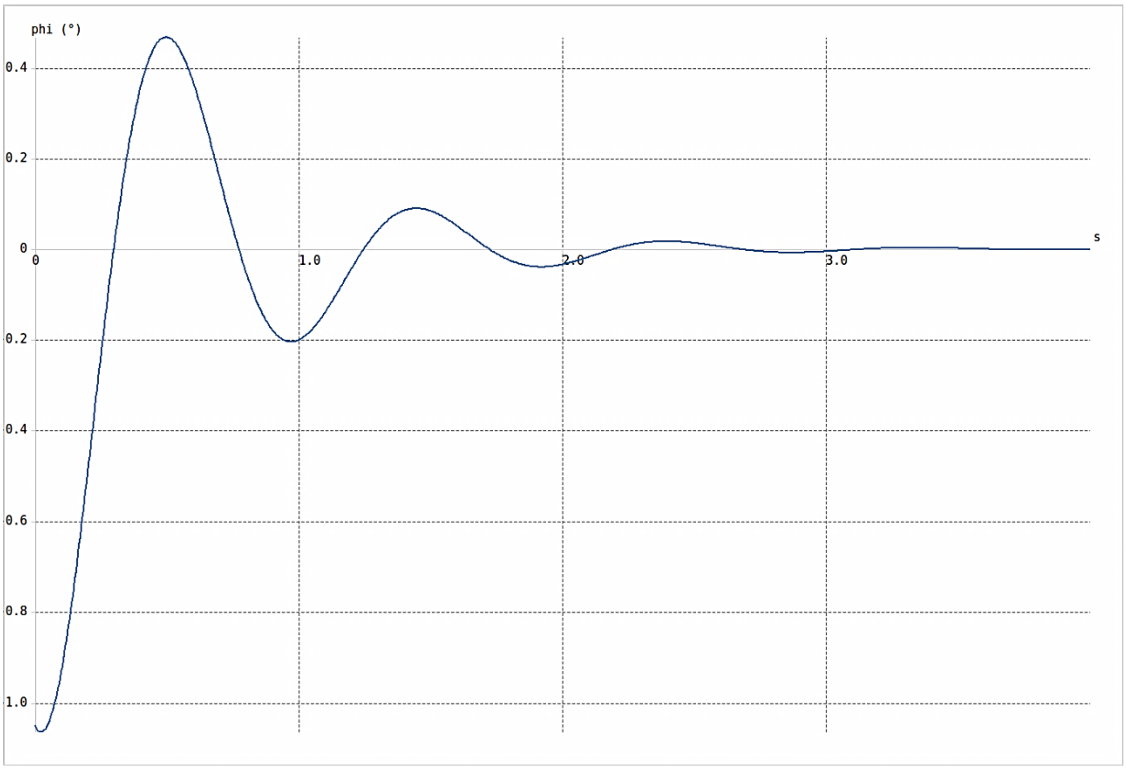


Figure 7.59 Time response: Roll angle (phi) against time(s)

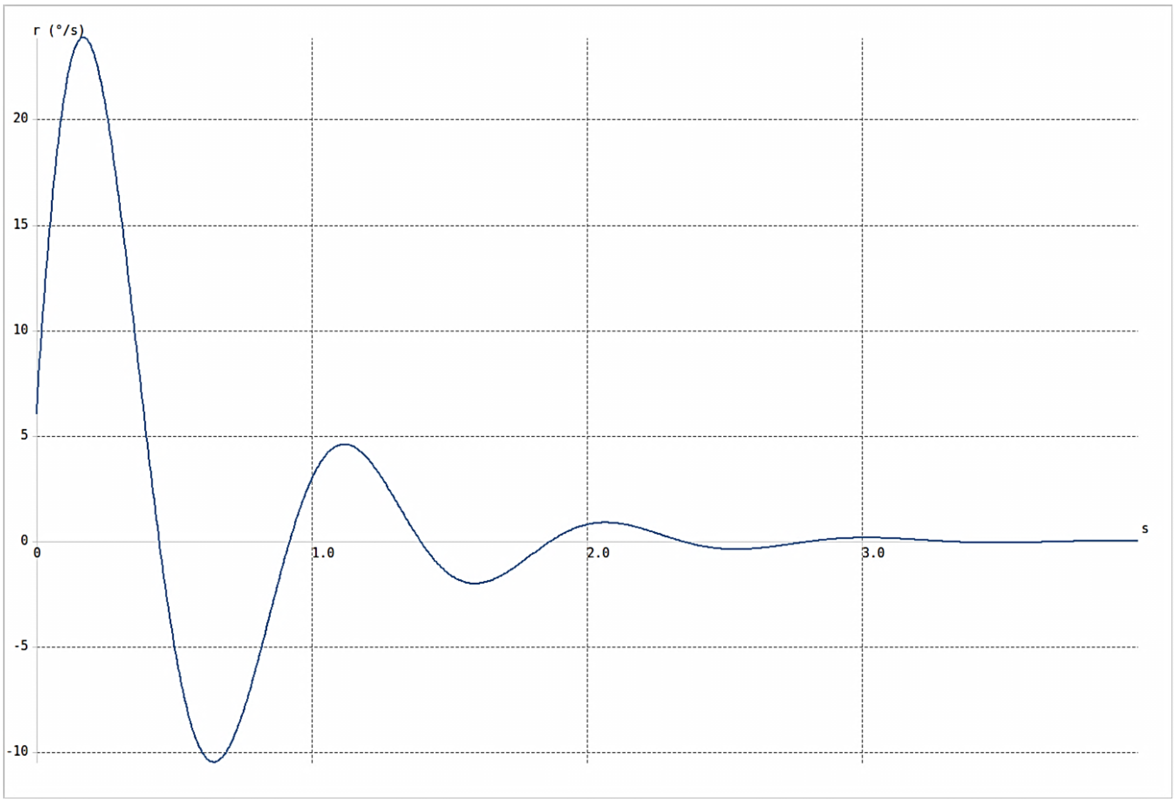


Figure 7.60 Time response: Yaw rate (r) against time (s)

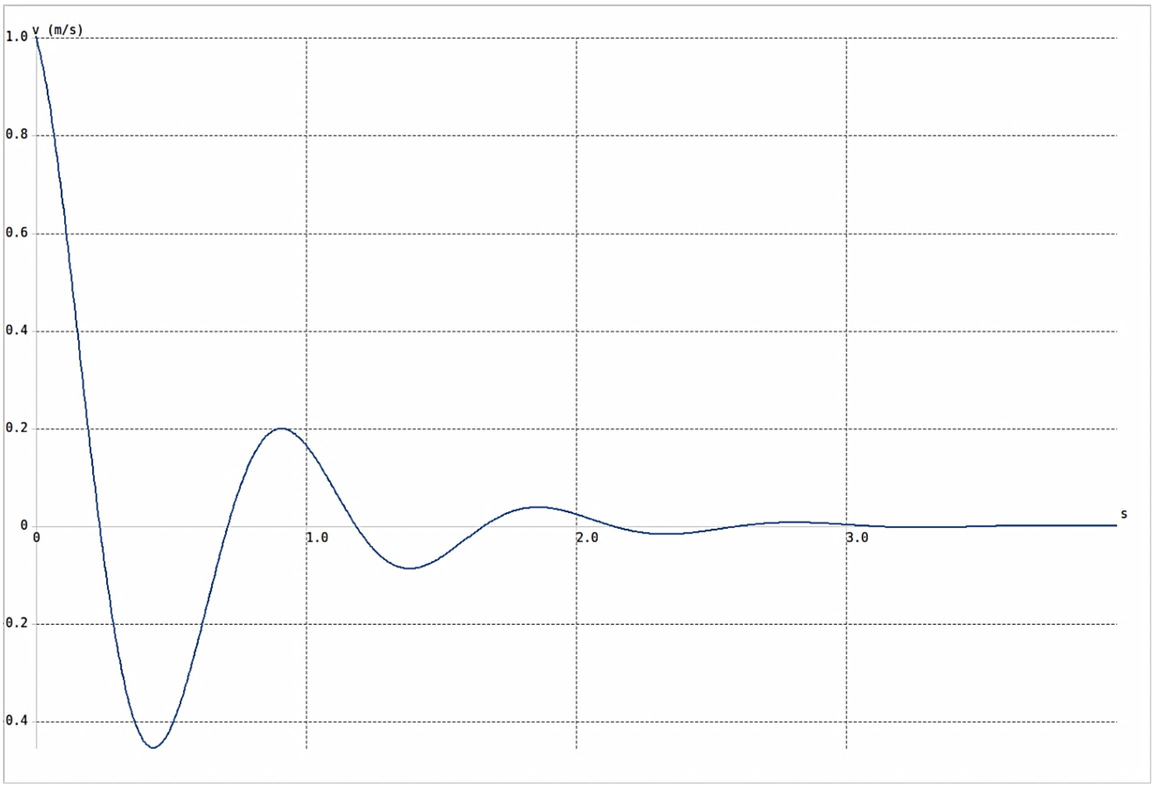


Figure 7.61 Time response: Lateral speed (v) against time(s)

Root locus:

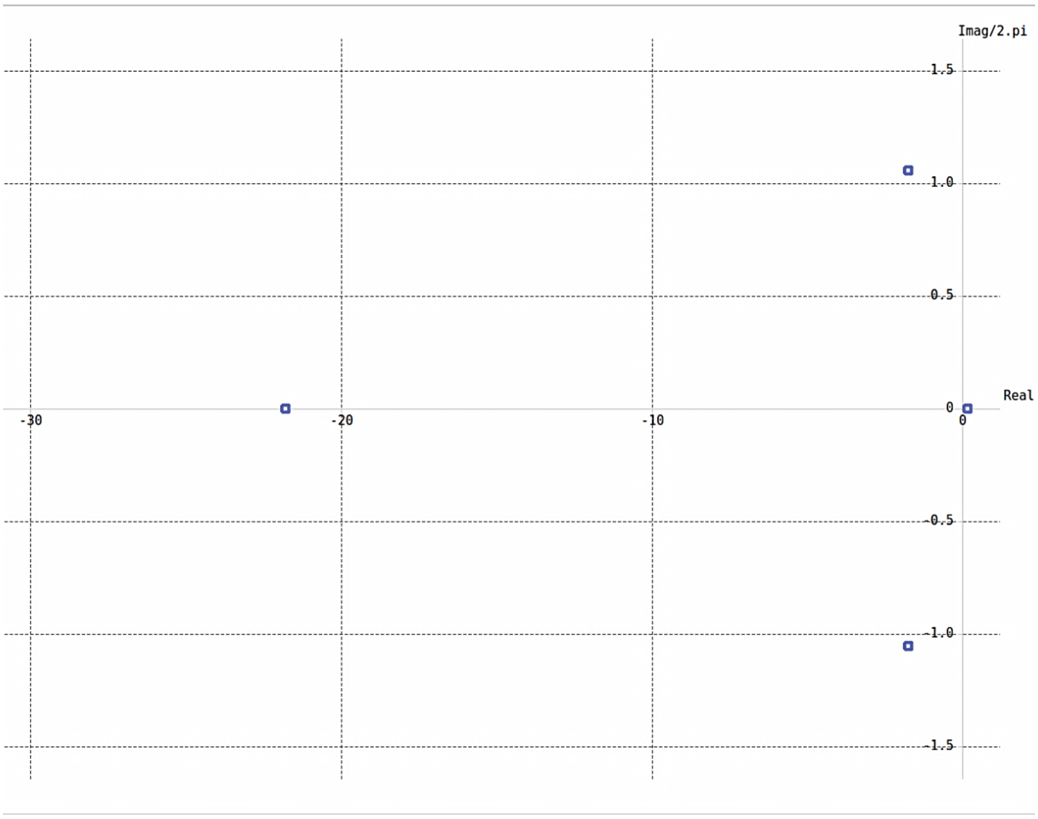


Figure 7.62 Root locus for FMS 182b Flap (lateral)

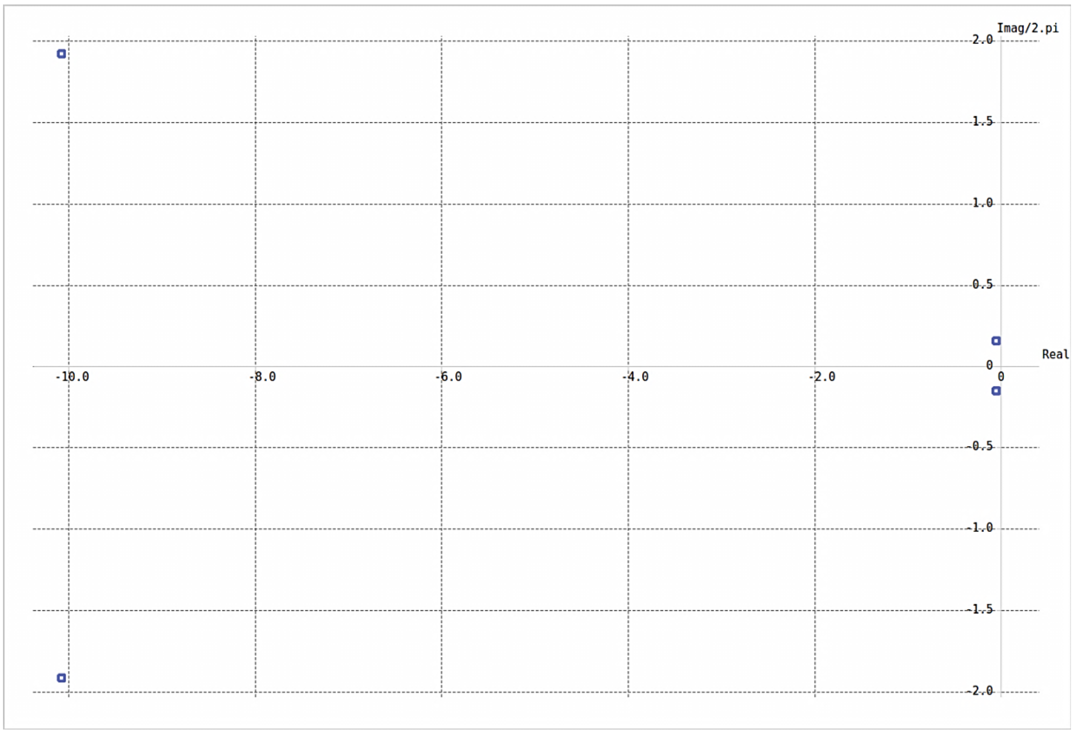


Figure 7.63 Root locus for FMS 182b Flap (longitudinal)

7.6.3 Control Test Results

T1 test results of the elevator discussed in Chapter 4.1.9.

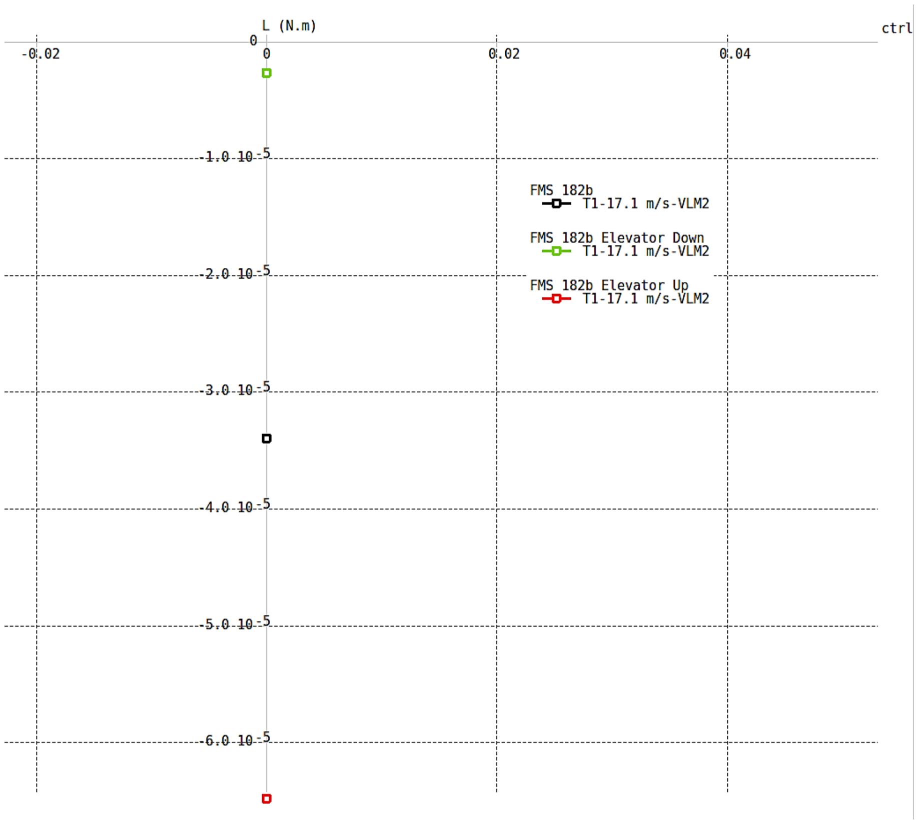


Figure 7.64 Rolling moment (L) against control gain

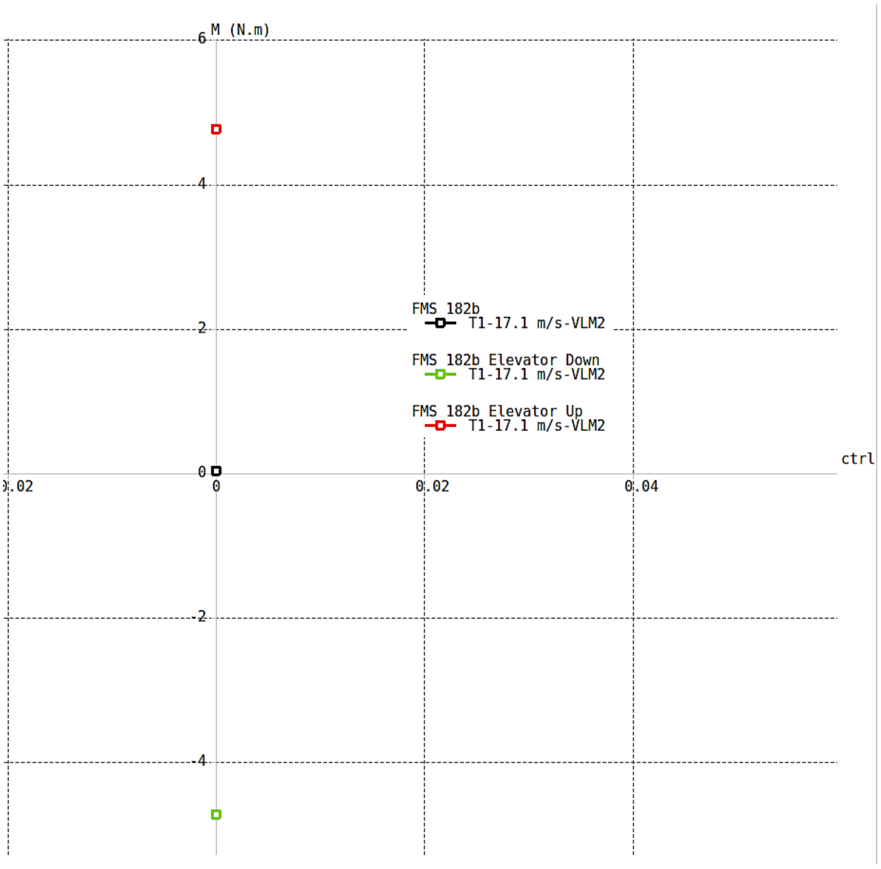


Figure 7.65 Pitching moment (M) against control gain

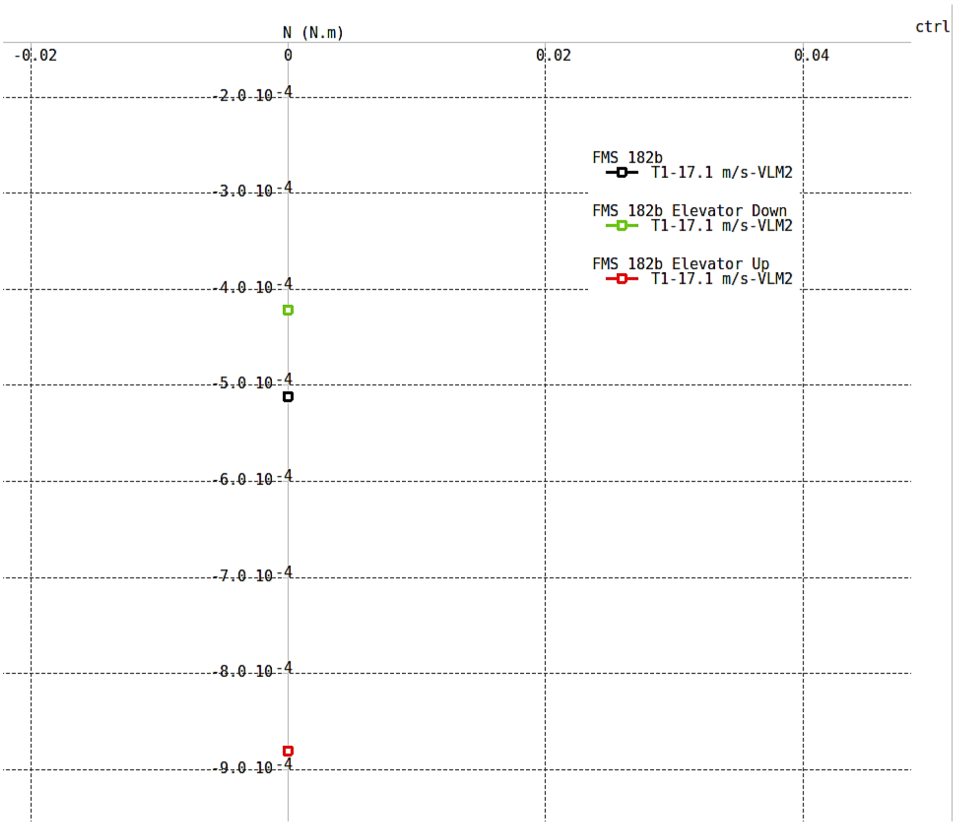


Figure 7.66 Yawing moment (N) against control gain

7.6.3.1 Variation of derivatives (flapped model)

controlPoints	Nde	CNde
-15	2.7811	0.081272
-14	2.6006	0.079518
-13	2.4788	0.078284
-12	2.4056	0.077522
-11	2.369	0.077123
-10	2.3575	0.076955
-9	2.3611	0.076899
-8	2.3724	0.076871
-7	2.3861	0.076818
-6	2.3989	0.07671
-5	2.409	0.076541
-4	2.4154	0.076318
-3	2.4184	0.076066
-2	2.4188	0.075827
-1	2.4181	0.075651
0	2.4176	0.075586
1	2.418	0.07565
2	2.4187	0.075824
3	2.4182	0.076061
4	2.4152	0.076311
5	2.4086	0.076533
6	2.3985	0.0767
7	2.3856	0.076806
8	2.3719	0.076859
9	2.3606	0.076885
10	2.357	0.076939
11	2.3685	0.077107
12	2.4051	0.077504
13	2.4784	0.078264
14	2.6003	0.079497
15	2.7809	0.081251

Table 7-18 Control test result from -15 to + 15 degrees

Based on the data set the averages:

Average Nde (NM)	Average Cnde
2.438513	0.077166

7.6.4 Plane Maker Aerofoil Test results

The following are a collection of the custom aerofoils designed for the FMS 182's X-Plane model via the methods described in the Methodology section of this thesis. Airfoil Maker has been used to visualise the foil characteristics. The x-axis on the plots are always +/- 20 degrees from the centreline.

7.6.4.1 Aerofoil polars for FMS 0010 low Reynolds number:

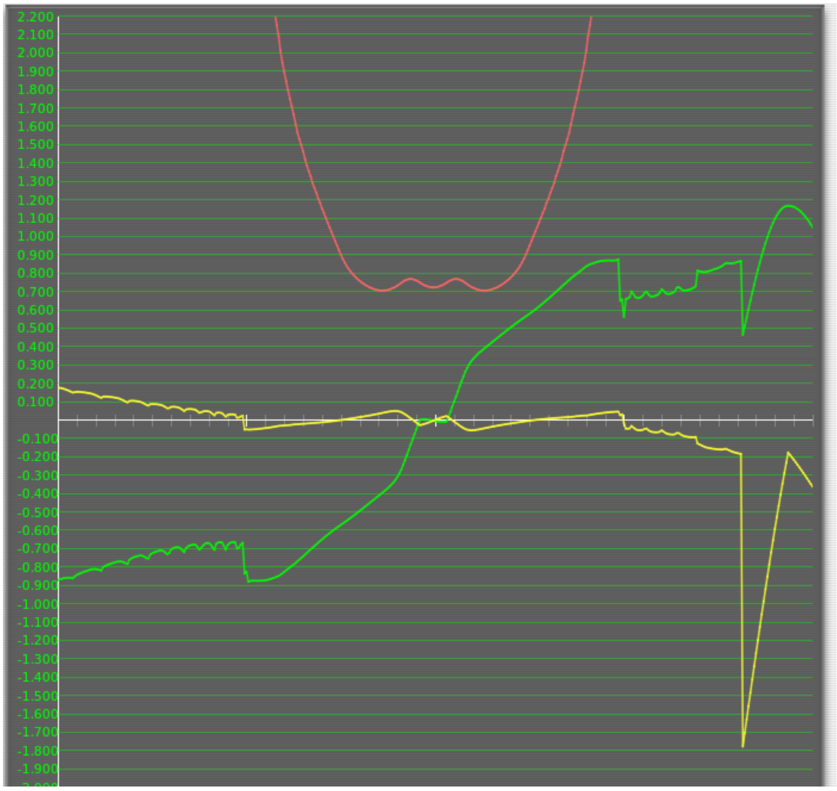


Figure 7.67 Aerofoil polar FMS 0010 Re 100k.afl (green = Cl, red = Cd, yellow = Cm)

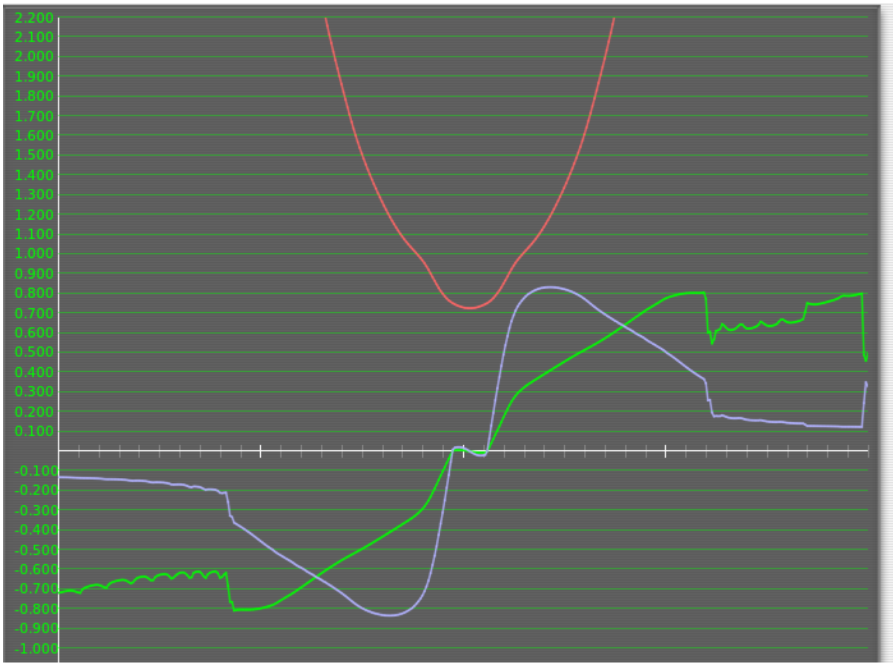


Figure 7.68 Aerofoil polar FMS 0010 Re 100k.afl (green = Cl, red = Cd, purple = L/D)

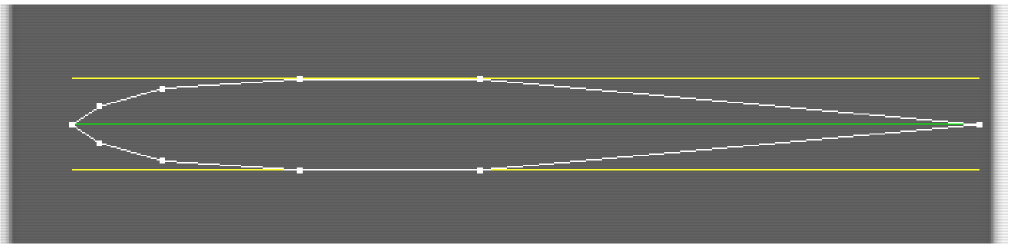


Figure 7.69 Aerofoil FMS 0010 Re 100k.afl

7.6.4.2 Aerofoil polars for FMS 0010 high Reynolds number:

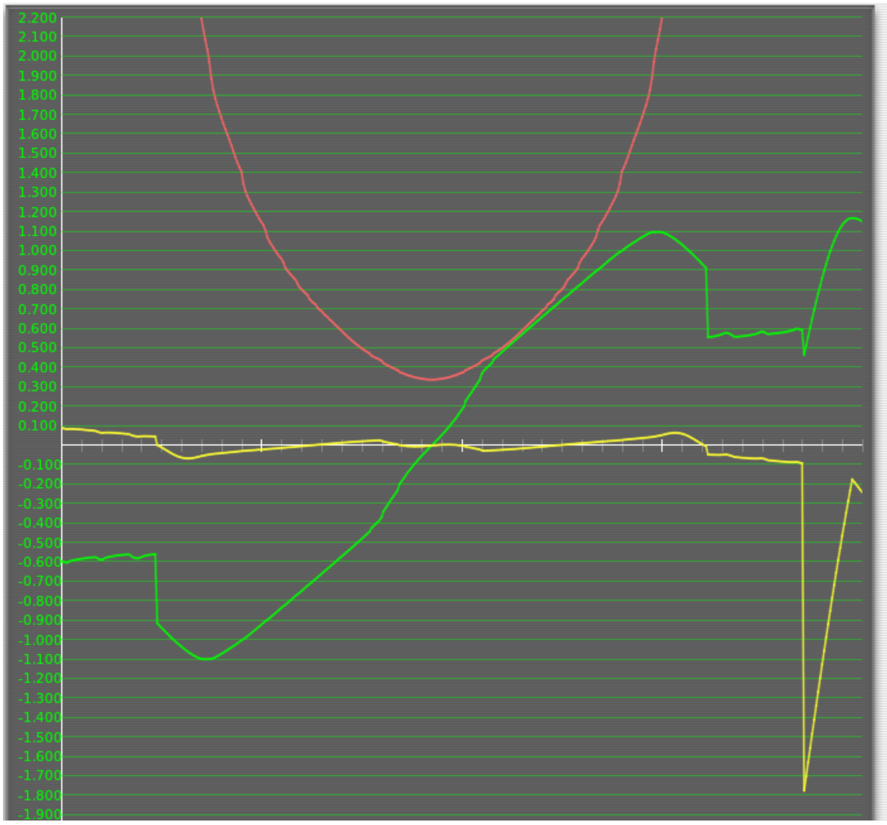


Figure 7.70 Aerofoil polar FMS 0010 Re 400k.afl (green = Cl, red = Cd, yellow = Cm)

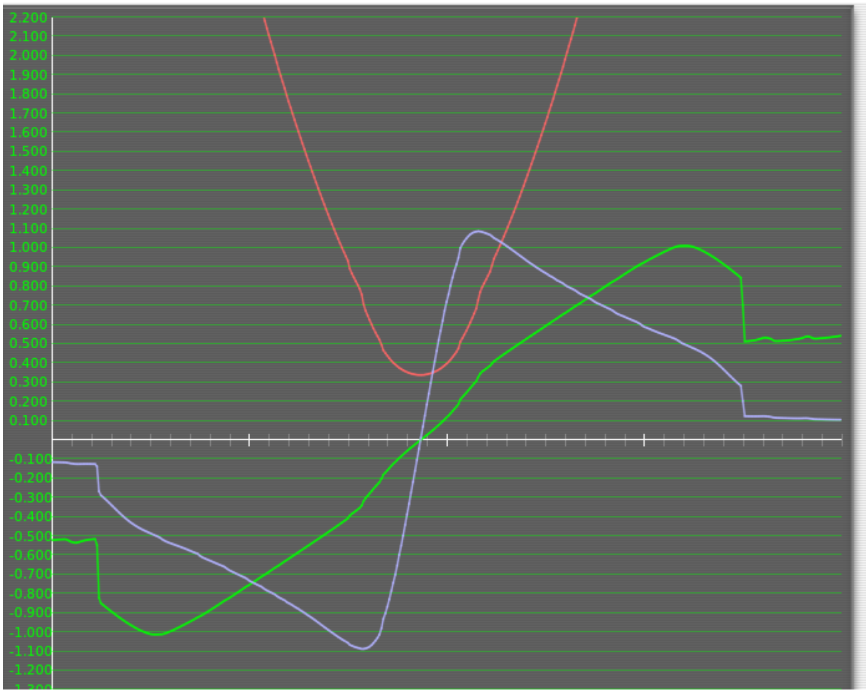


Figure 7.71 Aerofoil polar FMS 0010 Re 400k.afl (green = Cl, red = Cd, purple = L/D)

7.6.4.3 Aerofoil Polars for FMS 0012 (root) low Reynolds number:

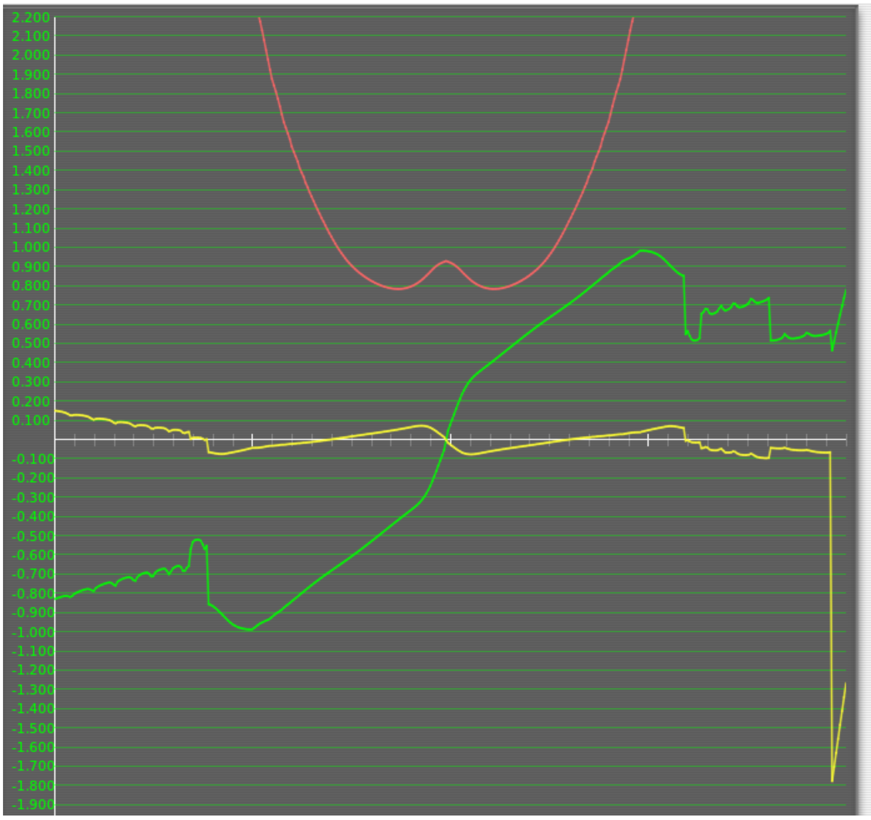


Figure 7.72 Aerofoil polar FMS 0012 Root Re 100k.afl (green = Cl, red = Cd, yellow = Cm)

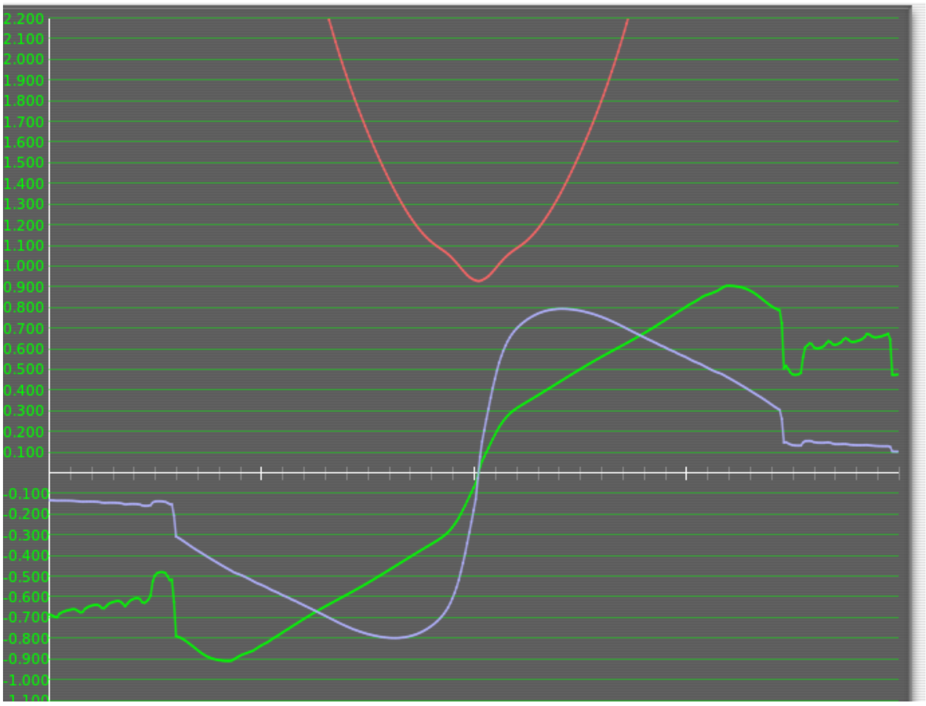


Figure 7.73 Aerofoil polar FMS 0012 Root Re 100k.afl (green = Cl, red = Cd, purple = L/D)

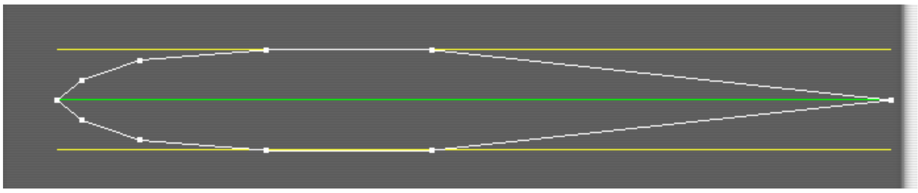


Figure 7.74 Aerofoil FMS 0012 Root Re 100k.afl

7.6.4.4 Aerofoil polars for FMS 0012 (root) high Reynolds number:

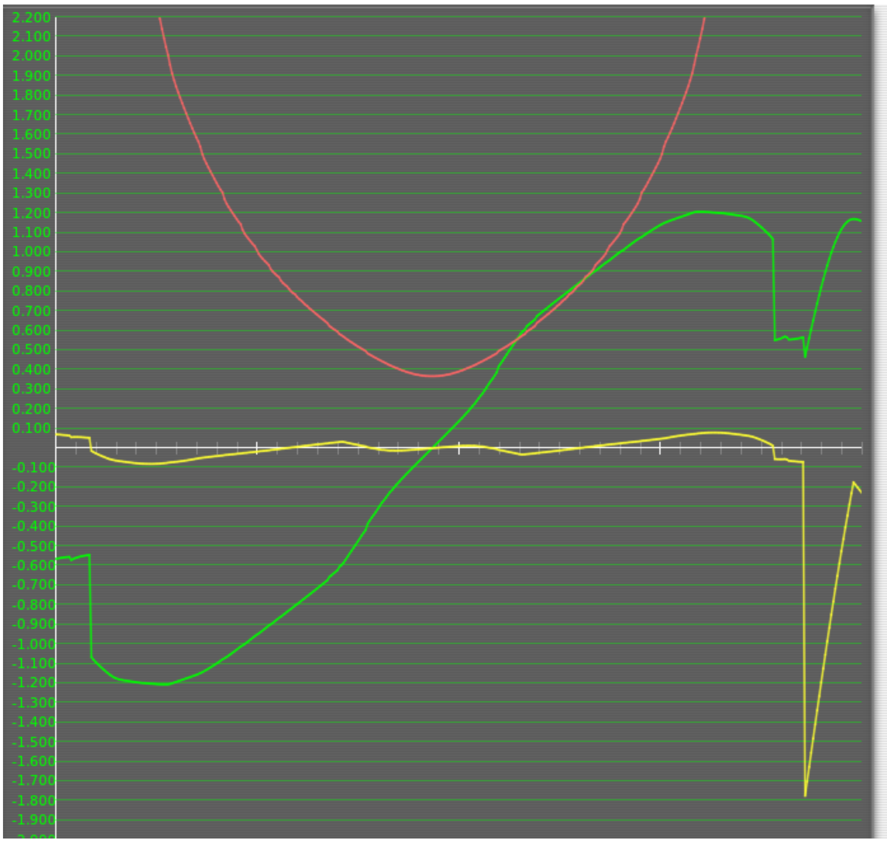


Figure 7.75 Aerofoil polar FMS 0012 Root Re 400k.afl (green = Cl, red = Cd, yellow = Cm)

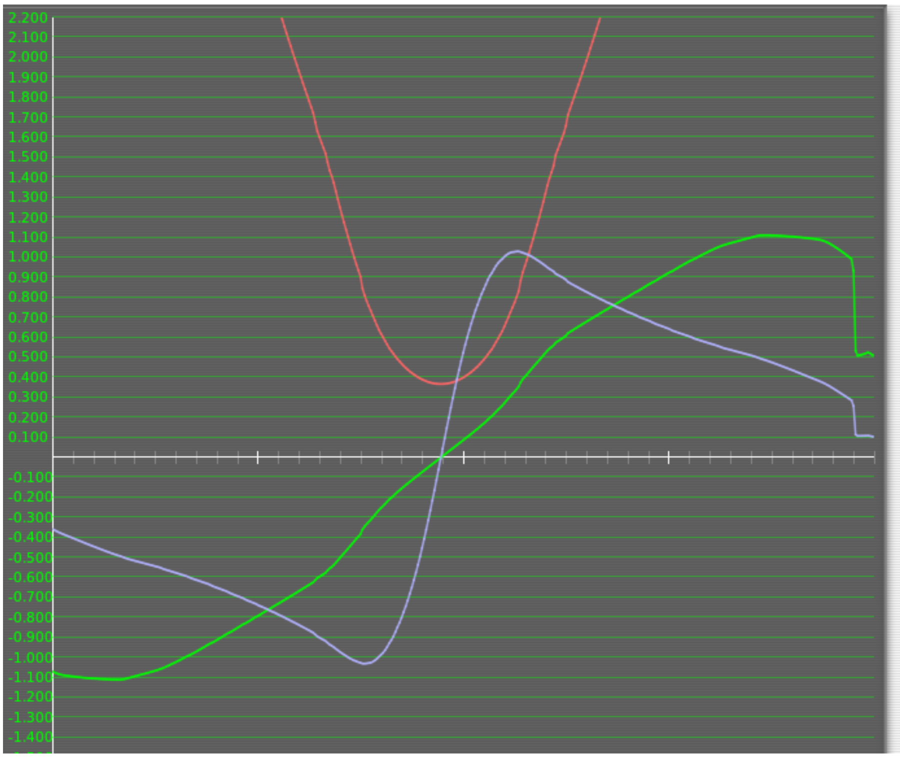


Figure 7.76 Aerofoil polar FMS 0012 Root Re 400k.afl

7.6.4.5 Aerofoil polars for FMS 0012 (tip) low Reynolds number:

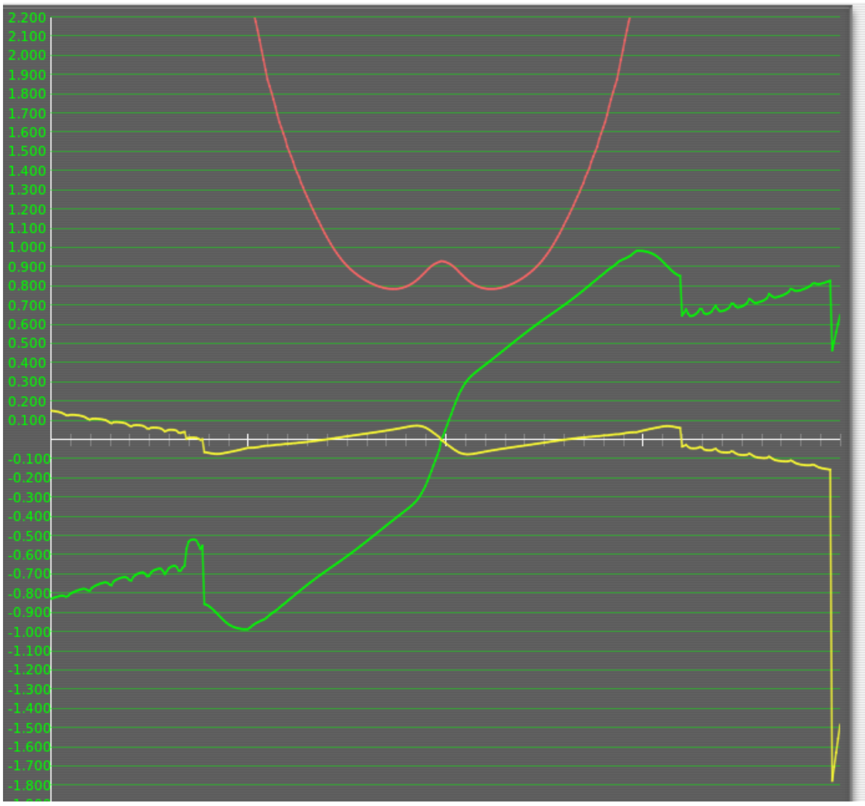


Figure 7.77 Aerofoil polar FMS 0012 Tip Re 100k.afl (green = Cl, red = Cd, yellow = Cm)

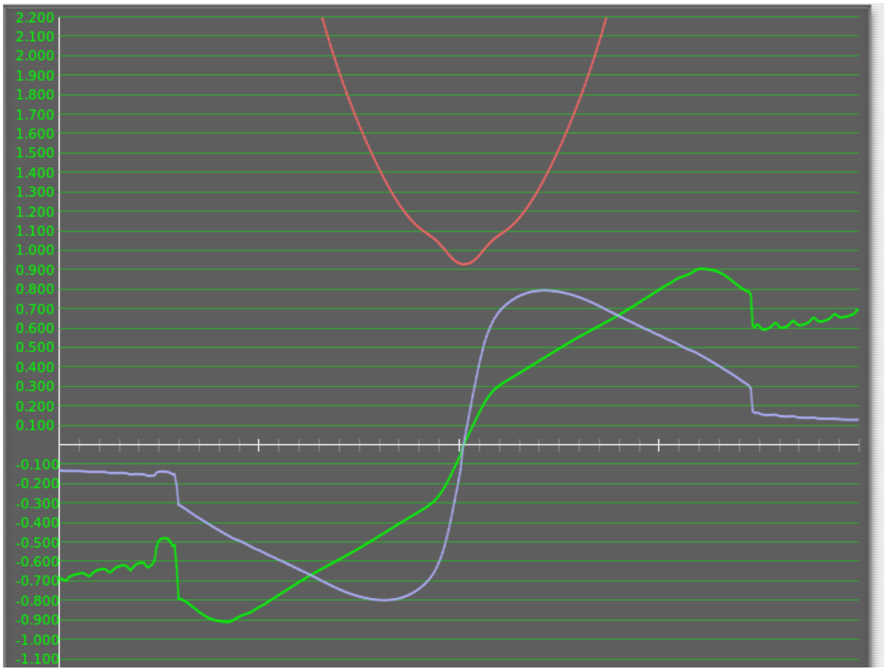


Figure 7.78 Aerofoil polar FMS 0012 Tip Re 100k.afl (green = Cl, red = Cd, purple = L/D)

7.6.4.6 Aerofoil polars for FMS 0012 (tip) high Reynolds number:

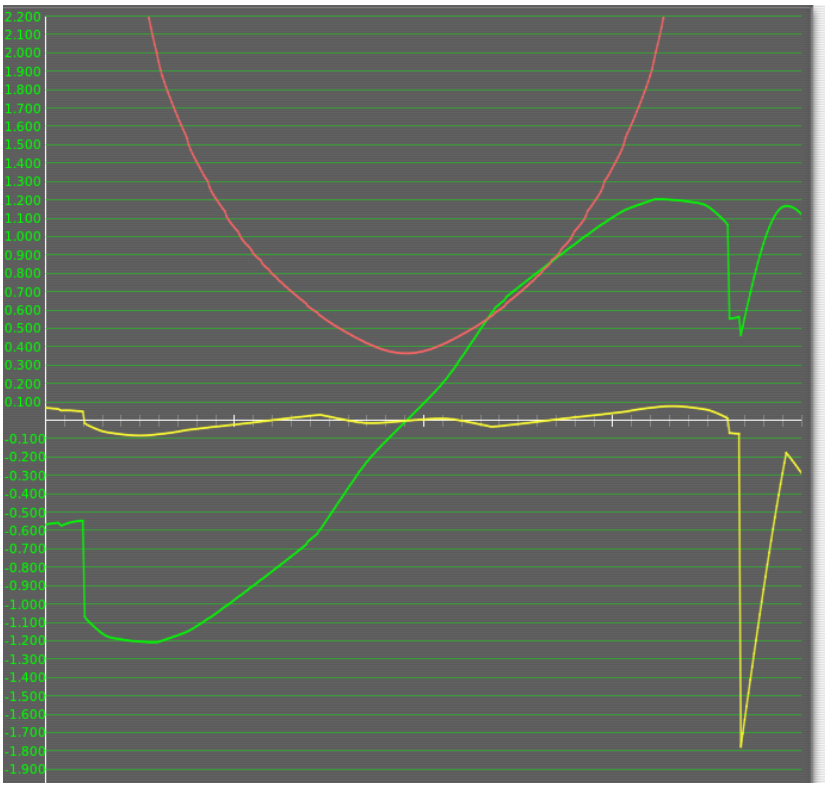


Figure 7.79 Aerofoil polar FMS 0012 Tip Re 400k.afl (green = Cl, red = Cd, yellow = Cm)

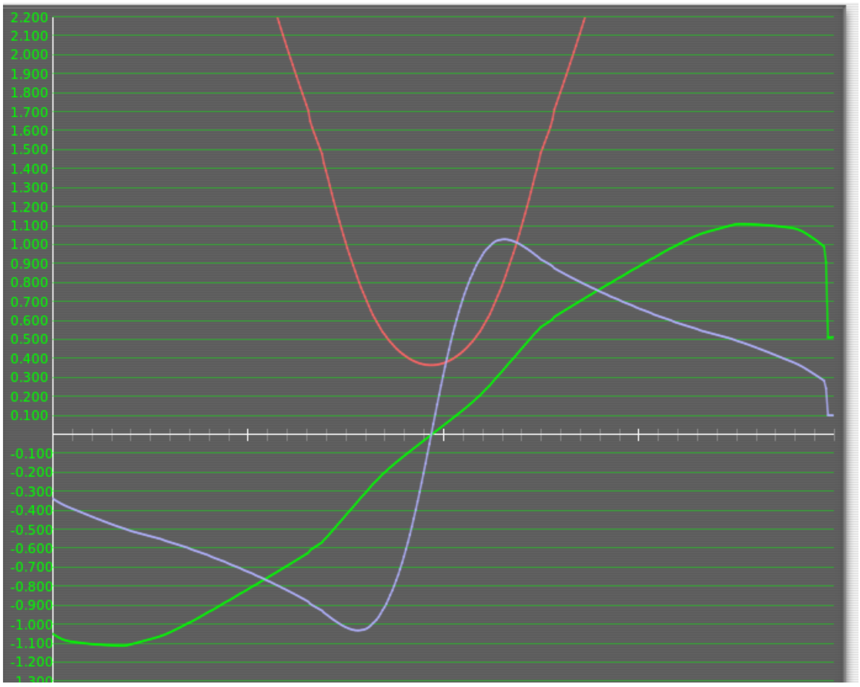
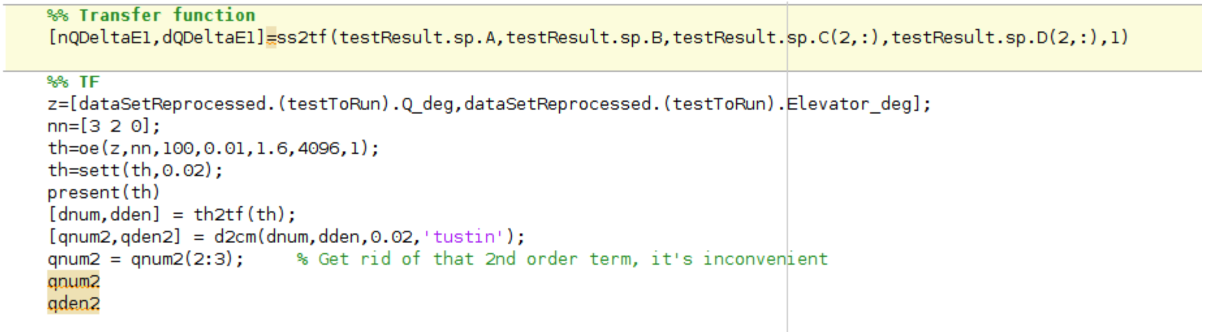


Figure 7.80 Aerofoil polar FMS 0012 Tip Re 400k.afl (green = Cl, red = Cd, purple = L/D)

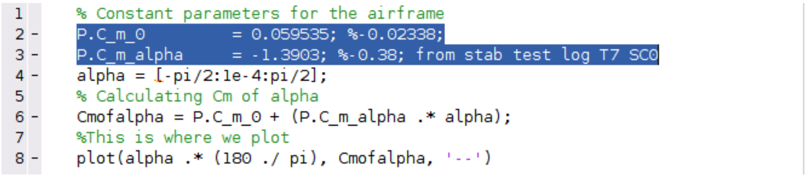
7.6.5 Modification of runscript function



```
%% Transfer function
[nQDeltaE1,dQDeltaE1]=ss2tf(testResult.sp.A,testResult.sp.B,testResult.sp.C(2,:),testResult.sp.D(2,:),1)

%% TF
z=[dataSetReprocessed.(testToRun).Q_deg,dataSetReprocessed.(testToRun).Elevator_deg];
nn=[3 2 0];
th=oe(z,nn,100,0.01,1.6,4096,1);
th=sett(th,0.02);
present(th)
[dnum,dden] = th2tf(th);
[qnum2,qden2] = d2cm(dnum,dden,0.02,'tustin');
qnum2 = qnum2(2:3);     % Get rid of that 2nd order term, it's inconvenient
qnum2
qden2
```

Figure 7.81 Addition to the script to calculate the elevator transfer fuctions based on the original code(see Chapter 3.3.13)

7.6.6 Linear lift code sample



```
% Constant parameters for the airframe
P.C_m_0         = 0.059535; %-0.02338;
P.C_m_alpha     = -1.3903; %-0.38; from stab test log T7 SC0
alpha = [-pi/2:1e-4:pi/2];
% Calculating Cm of alpha
Cmofalpha = P.C_m_0 + (P.C_m_alpha .* alpha);
%This is where we plot
plot(alpha .* (180 ./ pi), Cmofalpha, '--')
```

Figure 7.82 Using the XFLR5/X-Plane data to improve modelling accuracy

7.6.7 List of Aircraft Parameters

Parameter	Value	Description
g	9.81	Gravity

mass	1.322	Take-off weight of FMS 182
Jx	0.04119	x-component of Inertial Tensor from XFLR5 Stability Test
Jy	0.03961	y-component of Inertial Tensor from XFLR5 Stability Test
Jz	0.07279	z-component of Inertial Tensor from XRFL5 Stability Test
Jxz	-0.00057	Cross-coupling of x and z components of Inertial Tensor
Gamma	0.002998	Constant based on the J-matrix for moment calculation
Gamma1	-0.01416	Constant based on the J-matrix for moment calculation
Gamma2	0.805732	Constant based on the J-matrix for moment calculation
Gamma3	24.28038	Constant based on the J-matrix for moment calculation
Gamma4	-0.19043	Constant based on the J-matrix for moment calculation
Gamma5	0.797778	Constant based on the J-matrix for moment calculation
Gamma6	-0.01441	Constant based on the J-matrix for moment calculation
Gamma7	0.021817	Constant based on the J-matrix for moment calculation
Gamma8	13.73964	Constant based on the J-matrix for moment calculation
S_wing	0.26809	Wing surface area from XFLR5 and X-Plane
b	1.4097	Wing span based on measurements
c	0.19318	Mean Aerodynamic Chord from XFLR5 and X-Plane
S_prop	0.060207	Propeller Surface Area from X-Plane
rho	1.2682	Density of air at sea level
k_motor	83.1	Motor constant from Scorpion_calc
k_T_P	0	Motor constant from Scorpion_calc
k_Omega	0	Motor constant from Scorpion_calc
e	0.9	Oswald Efficiency Factor from XFLR5
AR	7.412638	Wing Aspect Ratio from XFLR5 and X-Plane
C_L_0	0.1064	Constant coefficient based MATLAB curve fitting
C_L_alpha	6.642	Constant coefficient based MATLAB curve fitting
C_L_q	8.7629	Constant from T2 test of XFLR5 at 6 degrees alpha
C_L_delta_e	-0.69867	Constant from T2 test of XFLR5 at 6 degrees alpha
C_D_0	0.02048	Drag coefficient calculated from XFLR5+X-Plane

		(profile+viscious+induced drag)
C_D_alpha	0.1265	Drag coefficient linearly approximated from the XFLR5
C_D_p	0.0437	Unable to estimate so a placeholder value from Aerosonde UAV was used
C_D_q	0	Unable to estimate
C_D_delta_e	0	Unable to estimate
C_m_0	0.059535	Moment coefficient constant from XFLR5 stability test
C_m_alpha	-1.3903	Moment Constant from XFLR5 stability test T7
C_m_q	-14.546	Moment Constant from XFLR5 stability test T7
C_m_delta_e	-1.95313	Moment Constant from XFLR5 control test
C_Y_0	-2.40E-05	Moment constant from XFLR5 stability test
C_Y_beta	-0.24388	Moment Constant from XFLR5 stability test T7
C_Y_p	-0.06015	Moment Constant from XFLR5 stability test T7
C_Y_r	0.23769	Moment Constant from XFLR5 stability test T7
C_Y_delta_a	0.01538	Aileron response constant from XFLR5 control test
C_Y_delta_r	0.171797	Rudder response constant from XFLR5 control test
C_ell_0	0	Moment constant
C_ell_beta	-0.00031	Moment constant (XFLR5 Stability Test -Dutch Roll)
C_ell_p	-0.49168	Moment constant (XFLR5 Stability Test -Dutch Roll)
C_ell_r	0.065902	Moment constant (XFLR5 Stability Test -Dutch Roll)
C_ell_delta_a	0.322494	Moment constant (XFLR5 Stability Test -Dutch Roll)
C_ell_delta_r	-0.00172	Rudder response constant (XFLR5 Control Test)
C_n_0	7.00E-06	Moment constant from XFLR5 T2 test
C_n_beta	0.10645	Moment constant (XFLR5 Stability Test -Dutch Roll)
C_n_p	-0.02498	Moment constant (XFLR5 Stability Test -Dutch Roll)
C_n_r	-0.10171	Moment constant (XFLR5 Stability Test -Dutch Roll)
C_n_delta_a	0.00375	Aileron response constant(XFLR5 Control Test)
C_n_delta_r	-0.0791	Rudder response constant (XFR5 Control Test)
C_prop	1	Propeller constant assumption
M	9.471	Transition rate(stall modelling) from MATLAB curve fitting.
alpha0	0.2657	Constant coefficient based MATLAB curve fitting
wind_n	0	Placeholder wind initialising values

wind_e	0	Placeholder wind initialising values
wind_d	0	Placeholder wind initialising values
Va0	10	Arbitrary Initial velocity
pn0	-800	Arbitrary initial position of the FMS 182 in simulated space (North)
pe0	500	Arbitrary initial position of the FMS 182 in simulated space (East)
pd0	-1000	Arbitrary initial position of the FMS 182 in simulated space (Down)
u0	10	Arbitrary initial velocity in the x-direction
v0	0	Arbitrary initial velocity in the y-direction
w0	0	Arbitrary initial velocity in the z-direction
phi0	0	Arbitrary initial roll angle
theta0	0	Arbitrary initial pitch angle
psi0	0	Arbitrary initial yaw angle
p0	0	Arbitrary initial roll rate
q0	0	Arbitrary initial pitch rate
r0	0	Arbitrary initial yaw rate

Table 7-19 FMS 182 MATLAB/Simulink modelling parameters based on XFLR5/X-Plane dataset

The above table contains all the constants used for the simulation in the form of a initialisation function. The methods and sources used to generate this datasheet for the FMS 182 is adequately described in the corresponding methods section.

7.7 Future Work Extra

This is where the additional material referred to in the Future work section is documented.

7.7.1 Arduino Related

Matlabl code (sweep_test.m) for checking an individual servo with a sweep test (For FMS 182):

```
clear all
a = arduino
s = servo(a,'D6')
pos = 0.5;
min_pos = 0.2;
max_pos = 0.9;
incre = 0.01;
direction = true;
writePosition(s, pos); % sends it to position half
while true
   pause(0.02);
```

```
if direction
   pos = pos+incre
   if pos >= max_pos
      direction = false;
   end
else pos = pos-incre
   if pos <= min_pos
      direction = true;
   end

end

writePosition(s,pos);
end
```

Matlab code (servo_test.m) for checking servo:

```
clear all
a = arduino
s = servo(a,'D6')

for angle = 0:0.02:1
   writePosition(s, angle);
   current_pos = readPosition(s);
   current_pos = current_pos*180;
   fprintf('Current motor position is %d degrees\n', current_pos);
   pause(0.1);
end
```

Additional code added in the Forces & Moments block m-file:

```
%% Code for aileron
map = 30/0.7;
ail_deg = (180/pi)*delta_a;
map_ail = ail_deg/map;
zero_pos = 0.55;

if map_ail == 0
   map_ail=zero_pos; % zero position of the servo
   writePosition(s,map_ail);
elseif map_ail<0
      map_ail=zero_pos+map_ail
      writePosition(s,map_ail);
else map_ail = zero_pos+map_ail
   writePosition(s,map_ail);
end
%% Code for rudder
map2 = 30;
rud_deg = (180/pi)*delta_r;
map_rud = rud_deg/map2;
zero_pos2 = 0.5;
```

```
if map_rud == 0
   map_rud=zero_pos2; % zero position of the servo
   writePosition(s,map_rud);
elseif map_rud<0
      map_rud=zero_pos2+map_rud
      writePosition(s,map_rud);
else map_rud = zero_pos2+map_rud
   writePosition(s,map_rud);
end
```

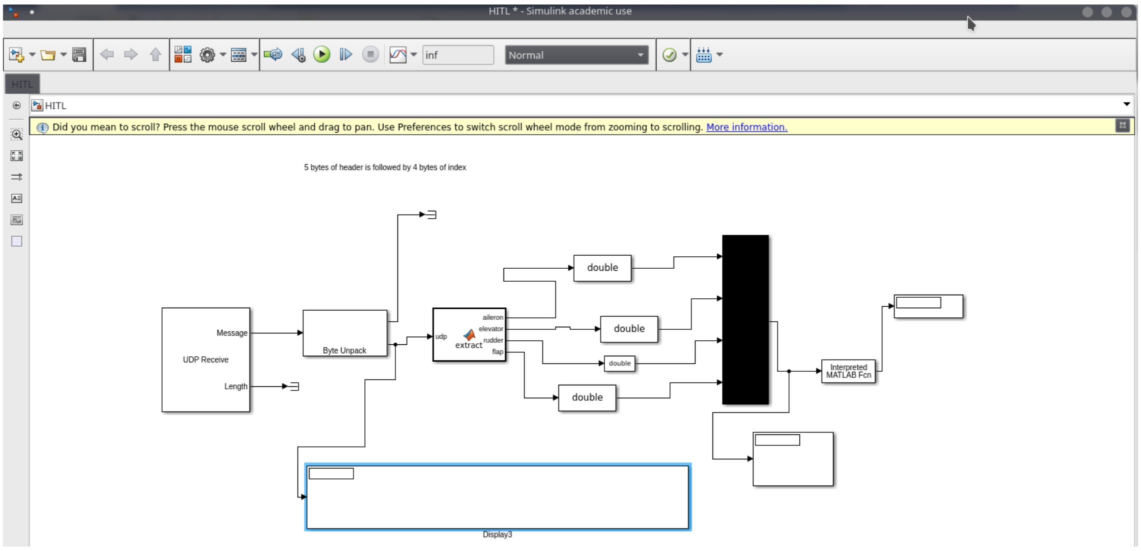
X-Plane HITL codes:

```
% Code for the interpreted function ardu
function [aileron, elevator, rudder, flap] = ardu(uu)
% function [aileron, elevator, rudder] = ardu(aileron,elevator,rudder)
global s
global s_a
global s_e
global s_f

aileron = uu(1);
elevator = uu(2);
rudder = uu(3);
flap = uu(4);

%% For flap
center = 0.5
if flap == 0
   writePosition(s_f, center);
else
   writePosition(s_f,0.9);
end


%% For aileron

map = 2/0.7;
map_ail = aileron/map;
zero_pos = 0.55;

if map_ail == 0
   map_ail=zero_pos; % zero position of the servo
   writePosition(s_a,map_ail);
elseif map_ail<0
      map_ail=zero_pos+map_ail
      writePosition(s_a,map_ail);
else map_ail = zero_pos+map_ail
   writePosition(s_a,map_ail);
end
%% For elevator
map3 = 0.9/2;
```

```
map_elev = elevator*map3;
zero_pos3 = 0.5;

if elevator == 0
   map_elev = zero_pos3;
   writePosition(s_e,map_elev);

elseif elevator <0
      map_elev = zero_pos3 + map_elev;
      writePosition(s_e,map_elev);
else map_elev = zero_pos3 + map_elev;

   writePosition(s_e,map_elev);
end



%% For rudder

map2 = 1/2;
map_rud = rudder*map2;
zero_pos2 = 0.5;

if rudder == 0
   map_rud = zero_pos2;
   fprintf('map_rud')
   writePosition(s,map_rud);

elseif rudder <0
      map_rud = zero_pos2 + map_rud;
      fprintf('map_rud')
      writePosition(s,map_rud);
else map_rud = zero_pos2 + map_rud;
   fprintf('map_rud')
   writePosition(s,map_rud);
end
end
```

Custom MATLAB function to import UDP data from X-Plane and extract them for use:

```
function [aileron, elevator, rudder, flap] = extract(udp)
% global a
% global s
% a = arduino
% s = servo(a,'D6')

coder.extrinsic('round');

elevator = udp(2,1); % first packet 2nd value
elevator = round(elevator,1);

aileron = udp (3,1);
aileron = round(aileron,1);

rudder = udp (4,1);
rudder = round(rudder,1);
rudder = -1*rudder;

flap = udp(6,2);
flap = round(flap,1);

uu = [elevator, aileron, rudder, flap];

end
```



HITL * - Simulink academic use

Did you mean to scroll? Press the mouse scroll wheel and drag to pan. Use Preferences to switch scroll wheel mode from zooming to scrolling. More information.

5 bytes of header is followed by 4 bytes of index

Figure 7.83 Screenshot of X-Plane HITL Simulink model