UNIVERSITY OF BATH

Department of Mechanical Engineering

FACULTY OF ENGINEERING AND DESIGN

FINAL YEAR MEng PROJECT REPORT

Name: Jemima Poynton

Date of submission: 10/05/2024

Word count: 10,500

"I certify that I have read and understood the entry in the Student Handbook for the Department of Mechanical Engineering on Cheating and Plagiarism and that all material in this assignment is my own work, except where I have indicated with appropriate references."

Author's signature: …………Jemima Poynton……………….

Supervisor: *Aykut Tamer*

Assessor: *Carl Sangan*

Abstract:

A control-and-mission-centric conceptual design tool has been developed in the MATLAB/Simulink environment. The overarching aim was to build a framework for a comprehensive concept design tool to combine different phases of flight into an overall mission, and output control requirements to analyse performance, including comfort, optimal mission and transition profile, and to investigate the effect of iterating design parameters.

A 6DoF dynamic mathematical model was developed based on an object-oriented approach. The aerodynamics was characterized by a Vortex Lattice Method (VLM) software built specifically for the tool. The results from the VLM module were compared against established software including: XFLR5, Tornado, AVL, and VortexLattice.jl. The model has been built such that any number of rotors or wings in any mounting position and orientation may be included to meet the unique challenges presented by the highly variable configurations of UAM aircraft.

Particle swarm optimization has then been applied to reference aircraft to optimize the trim points through a mission. Analysis of the parameter space was performed, indicating several closely located global minima, hence a global optimization algorithm was required. The cost function has been defined to both enforce trim conditions, and a course angle and flight path derived from the mission definition. These equilibrium points have been used as references for a gain scheduling reference controller. The controller gains were tuning using both Linear Quadratic Regulator (LQR) and Linear Quadratic Integral (LQI), and the two were compared. The weighting functions corresponding to how well each state is driven toward the demand were tuned using particle swarm optimization.

Unfortunately, due to a loss of data, the final stage of the report that was intended to demonstrate the use of the tool for iterative concept design could not be completed. Instead, full descriptions are provided of this would be applied in each case. The areas of investigation included comparing two different UAM aircraft for power requirements over two mission, with and without the aerodynamic interference of the rotor on the wing. Similarly, the number of rotors and their position would be analyzed for UAM 2. Analysis would also be carried out to look at passenger comfort characterized by rotational acceleration and jerk for UAM 1 under LQI control. A sensitivity analysis would be carried out by varying the control authority of the control surfaces, and analyzing the effect on the step response of the aircraft.

Contents

Table of Figures

List of Symbols and Abbreviations

General Acronyms

UAM	Urban Air Mobility
VTOL	Vertical Take-Off/Landing
eVTOL	Electric Vertical Take-Off/Landing
UAV	Unmanned Aerial Vehicle
I/O	Input(s)/Output(s)
PSO	Particle Swarm Optimisation
VLM	Vortex Lattice Method
CG	Centre of Gravity
CoL	Centre of Lift
LQR	Linear Quadratic Regulator
LQI	Linear Quadratic Integral
MIMO	Multi-Input Multi-Output
VRS	Vortex Ring State

Transformation Matrices and Coordinate Systems

L_1	Transformation matrix between individual component and overall body frame (roll).
L_2	Transformation matrix between individual component and overall body frame (pitch).
L_3	Transformation matrix between individual component and overall body frame (yaw).
R_b^c	Rotation matrix from component to body frame
R_b^r	Rotation matrix from rotor to body frame
W	Body to Euler angular rate transformation
φ_c	Component plane 'yaw' angle relative to the body frame.
θ_c	Component plane 'pitch' angle relative to the body frame.
ψ_c	Component plane 'roll' angle relative to the body frame.
R_b^w	Rotation matrix from body to wind frame
R_b^e	Rotation matrix from body to Earth frame

Component Modelling

C_T	Thrust coefficient
C_p	Power coefficient
$C_{p,i}$	Induced power coefficient
$C_{p,0}$	Profile power coefficient
$C_{p_{net}}$	Net power coefficient (unsteady behaviour)
r	Distance from control point P to vortex filament
Γ_n	Vortex filament strength.
$\vec{V}$	Induced velocity vector in vortex lattice method
M_{cg}	Moment about the centre of gravity
F_w	Aerodynamic forces in the wind frame
S	Reference/planform area
m	Aircraft mass
L_n	Length of nacelle/distance of rotor from wing
R	Radius of rotor
T	Rotor thrust
M	Empirical stall constant
α_0	Empirical stall constant
S_{ss}	Area of slipstream zone
R_w	Rotor wake contraction radius
ρ	Air density
P	Vortex lattice method control point
dl	Length of vortex filament
l	Moment arm
l_m	Moment arm from panel m to control point
c	Reference/mean chord
b	Reference span
c_w	Wing chord at rotor mounting position
S_{ss0}	Slipstream area with rotor undeflected
v_{ss}	Induced velocity of rotor on wing
V_{xss}	Forward velocity in slipstream zone
V_{zss}	Vertical velocity in slipstream zone

Aerodynamic/Stability Derivative Convention

All stability and control derivatives follow the convention $C_{a_b} = \frac{\partial C_a}{\partial b}$.

Flight Dynamics

$\boldsymbol{C_L}$	Lift Coefficient	$\boldsymbol{C_l}$	Roll Moment Coefficient
$\boldsymbol{C_D}$	Drag Coefficient	$\boldsymbol{C_m}$	Pitch Moment Coefficient
$\boldsymbol{C_Y}$	Side Force Coefficient	$\boldsymbol{C_n}$	Yaw Moment Coefficient
$\boldsymbol{\alpha}$	Angle of Attack	$\boldsymbol{p}$	Roll rate
$\boldsymbol{\beta}$	Sideslip Angle	$\boldsymbol{q}$	Pitch rate
$\boldsymbol{V_a}$	Airspeed	$\boldsymbol{r}$	Yaw rate
$\boldsymbol{\delta_{c_i}}$	Deflection of i^{th} control surface	$\boldsymbol{g}$	Gravitational constant
$\boldsymbol{F_b}$	Total forces in body frame	$\boldsymbol{F_a}$	Aerodynamic forces in body frame
$\boldsymbol{F_g}$	Force of gravity in body frame	$\boldsymbol{F_T}$	Forces due to thrust in body frame
$\boldsymbol{M_b}$	Total moments in body frame	$\boldsymbol{M_a}$	Aerodynamic moments in body frame
$\boldsymbol{\tau_T}$	Torque/moment due to thrust	$\boldsymbol{C_{D0}}$	Zero angle of attack drag coefficient
$\boldsymbol{C_{Y_0}}$	Zero angle of attack side force coefficient	$\boldsymbol{C_{L_0}}$	Zero angle of attack lift coefficient
$\boldsymbol{\theta}$	Pitch Euler angle	$\boldsymbol{T_m}$	Thrust of m^{th} rotor
$\boldsymbol{\varphi}$	Roll Euler angle	$\boldsymbol{k_t}$	Constant relating thrust to rotational velocity squared
$\boldsymbol{\psi}$	Yaw Euler angle	$\boldsymbol{k_b}$	Rotor drag coefficient
$\boldsymbol{\omega}$	Rotational velocity	$\boldsymbol{p_e}$	Position of aircraft in Earth frame
$\boldsymbol{I_b}$	Inertia tensor		

Control and Trim

$\boldsymbol{x}$	State Vector	$\boldsymbol{O_{n\times m}}$	Zero matrix of dimensions m by n
$\breve{\boldsymbol{x}}$	Trimmed/Desired State Vector	$\boldsymbol{K}$	Gain Matrix
$\tilde{\boldsymbol{x}}$	State Vector Error wrt. Trim State	$\boldsymbol{K_x}$	Proportional Gain Matrix
$\boldsymbol{J}$	Cost/Objective Function	$\boldsymbol{K_i}$	Integral Gain Matrix
$\boldsymbol{H}$	Weighting Matrix	$\boldsymbol{k_b}$	Back Calculation Coefficient

1. Introduction

1.1. Research Motivation

Uber was one of the first to kickstart the air taxi concept with its Urban Air Mobility (UAM) initiative, Uber Elevate, in 2016[1-3]. Currently, the forecast profit has attracted both the interest of major aerospace companies, and extensive investment into eVTOL startups[1,3]. Major development projects include[3]: Vertical Aerospace's VX4[4], VoloCity[5], Lilium Jet[6], CityAirBus[7] and Boeing's NeXt PAV[2].

The future vision of UAM aircraft is a safer, quieter, and less expensive alternative to helicopters[3,8], alleviation of traffic congestion[1,9] and an environmentally friendly mode of transport that will improve connectivity in urban areas[9].UAM aircraft are required to take off from urban facilities such as parking garages, tall buildings, warehouses and underutilised heliports[1,3]. Due to these unique challenges, eVTOLs make up the majority of designs[1,3]. eVTOL technology has significant overlap with the rise of unmanned aerial vehicles in urban environments, for applications including connectivity and cargo delivery. Electric aircraft are well suited as they have low noise, a small footprint[3] and they require no petroleum-based refuelling facilities[3].

The two broad categories of configurations are multicopter, and compound fixed-wing VTOL aircraft. **Figure 1** summarises the most common classifications of eVTOL aircraft. 'Lift + Cruise' configurations have separate dedicated rotors for the VTOL and cruise mode. Thrust vectoring is achieved by tiltwing, tiltrotor, tilt-duct, or folding-wing configurations[1]. Tilt-wing and folding-wing configurations are less common due to slow, energy-intensive transition manoeuvres[1,3]. It is key to note that these designs feature highly variable geometry, making standard tools difficult to apply.

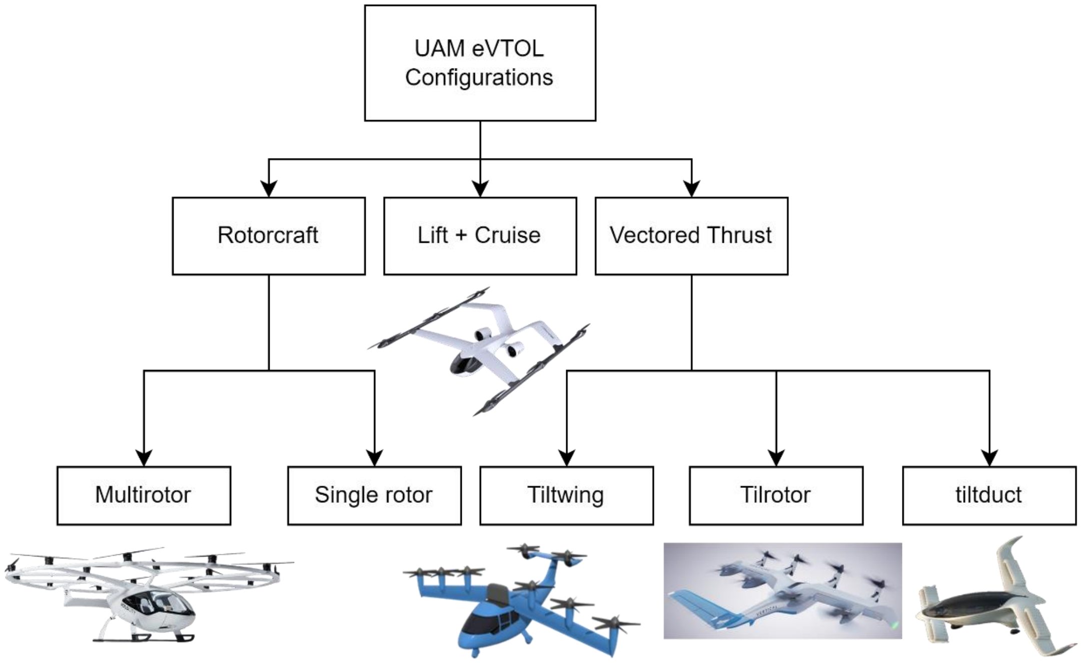


***Figure 1:** eVTOL configuration classifications.*

Hybrid fixed-wing and VTOL configurations are often favoured for UAM due to take advantage of the efficiency and higher speeds during the cruise phase[3,10]. The combination of multicopter, fixed-wing and transition phases results in a complex mission profile, such as that shown in **Figure 2**. The transition phases will vary based on aircraft configuration.

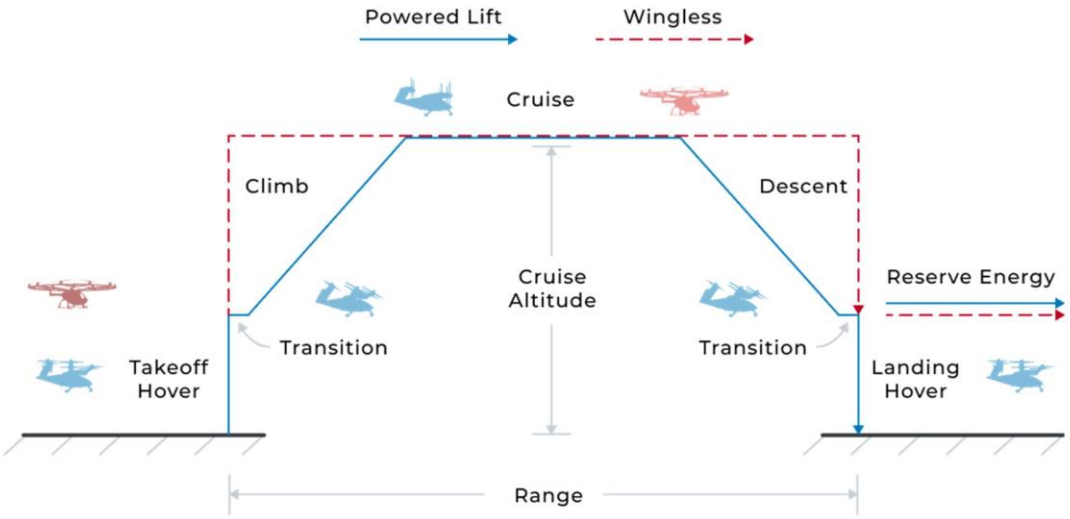


Figure 2: *UAM example mission profile for powered lift and rotorcraft types. Source: [11].*

Compound fixed-wing eVTOLs are still an emerging technology, requiring further research. This is particularly evident in the concept design stage, where there is heavy reliance on empirical, historical, and high-level modelling data (e.g. experimental, wind tunnel or CFD). Whilst UAM aircraft are being extensively researched in recent years, the lack of a standard platform limits the wide applicability of the research. Tilt-rotor aircraft are one of the most common approaches in the design space. The goal of this work is to provide a tool to provide insights into mission performance in the highly divergent early concept design stages.

1.2. Aims and Objectives

It has been established that the trajectory and mission profile of a UAM vehicle is vital to verifying that it meets certification requirements. The diversity of H-VTOL designs has created a need for a generalised and flexible tool for generating key mission performance data. The key areas of interest driving the design approach are:

1. Novelty: There is high variability in the designs and configurations of UAM aircraft, for which empirical data is limited[12].
2. Safety: Not only is safety critical for certification, public perception of safety is a key limiting factor in the adoption of UAM[8].
3. Trajectory: Trajectory modelling is vital to certification in an urban environment. The transition stage presents particular difficulty due to the highly dynamic nature and stringent safety requirements[13].
4. UTM/ATM: Significant infrastructure is required around urban traffic management (ATM)[8,14]. Trajectory-focused design tools should aim to be compatible.
5. Cost: Adoption of UAM is heavily cost dependent, putting power and weight considerations at the forefront[8].

This project work is not intended to replace high-level, higher fidelity modelling software, but rather to provide stability, comfort, and control insights earlier on in concept design. High-level commercial software presents 3 key difficulties at this stage: availability of data, computational cost, and skill/modelling development time. At these stages, the design is subject to significant and frequent change, where these difficulties become costly, and ideally should be subject only to detailed design changes, rather than complete iteration. **Figure 3** illustrates how the proposed tool will fit into the concept design workflow.

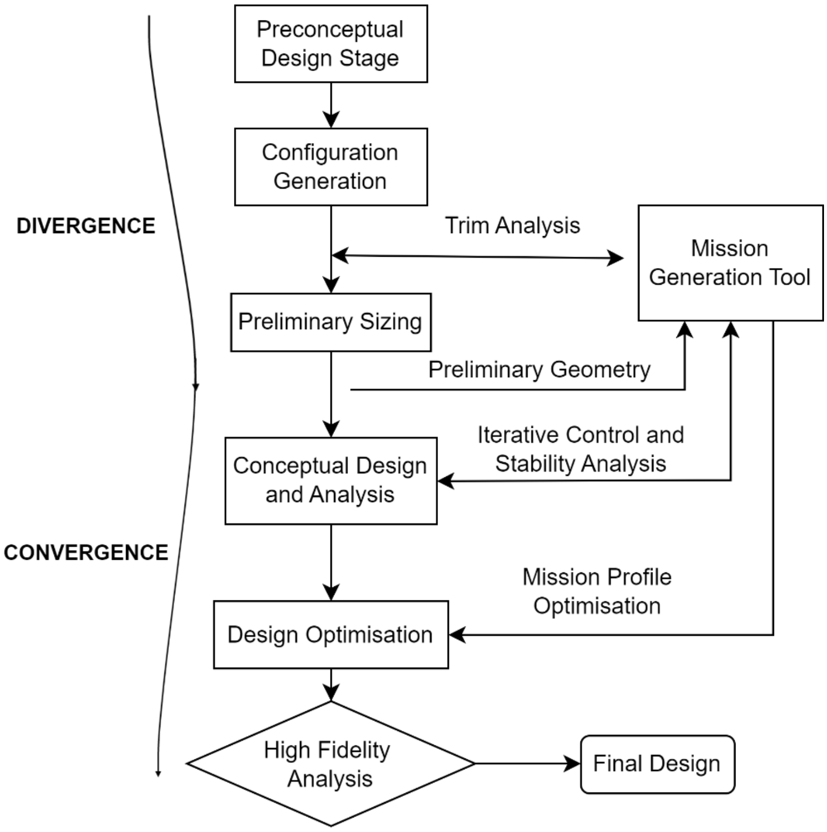


Figure 3: *Proposed concept design workflow. Adapted from: [15].*

Early on in the design process, the trim analysis may be used to verify early control surface sizing, with basic aerodynamic analysis being performed on the preliminary geometry. The tool should be quick and easy to use to make it most effective in the divergent stage of concept design. The mission profile generation may also be used for design optimisation to test and generate optimal mission profiles.

Modularity is key. Mission profile generation encompasses a wide scope of modelling, simulation, and control techniques. The framework should be such that further developments can be made, such as adapting the aerodynamic model to better represent the wide range of novel concepts. At this stage, the focus is placed on simple thrust vectoring designs such as Tilt-Duct or Tiltrotor aircraft but may developed to include folding-wing or tilt-wing designs in future.

The explicit objectives are outlined below.

Overarching Aim:

Build a framework for a comprehensive concept design tool to combine different phases of flight into an overall mission, and output control requirements to analyse performance.

Objectives

1. Develop a unified set of component models, without reliance on empirical or high-level data.
2. Create a 6DoF dynamic model that can output states given a set of mission parameters.
3. Develop a generalised transition model by blending cruise and VTOL phases.

4. Create a function to perform trim analysis on an over-actuated system (i.e., transition).
5. Develop functions to post-process mission data into key mission and point performance analyses.
6. Apply the model to 2 reference vehicles to demonstrate use.
7. Build the framework such that it can supplement/integrate with established tools.

2. Background Literature

Mission profile generation encompasses a wide scope of technical areas. This section serves as an overview of the methods used in UAM design, with in-depth technical detail being included later in the implementation of the described methods.

2.1. Currently Available Tools

Some key high-fidelity tools for rotorcraft/fixed-wing conceptual design are summarized in **Table 1**.

There is no widely accepted performance model structure due to the novelty of UAM configurations and unique requirements of the research requirements[3,16]. In [[16]], Hartman et al. applied current performance modelling methods to a set of conceptual UAM vehicles, and concluded that the performance modelling was limited by the absence of a fully defined operational environment, limited design details, and a lack of direct compatibility with UAM performance characteristics[3,16]. Hartman et. al. highlighted to need for future work to focus on more generalised performance modelling methods, and creating models with increased operational flexibility[3,16]. Sarkar et al.[17] integrated SUAVE with unmanned traffic management (UTM) software to evaluate performance based on a series of mission profile inputs. Infeasible mission profiles were identified to identify the feasibility of the design for realistic urban traffic management scenarios.

Table 1: *Summary of Currently Available Modelling Tools[3].*

Modelling Tool	Type	Method	Fidelity	eVTOL Functionality?
NDARC[15]	Rotorcraft conceptual design environment	Parametric	High	Adaptable
CREATION[15]	Design toolchain for flight performance analysis of rotorcraft	Suite of tools	Variable	Adaptable
CAMRAD II[15]	Aeromechanical analysis of helicopters and rotorcraft	Combination of advanced methods	High	-
HYDRA[15]	Rotorcraft design tool	Parametric /Physics-based	Variable	Adaptable
HETLAS[19]	Rotorcraft design tool	Unified modelling/ Component-based	High	Yes
FLIGHTLAB [19]	Aircraft (rotorcraft and fixed wing) modelling and analysis tool	Component-based	Selective	Yes
SUAVE[18]	Conceptual level aircraft design environment	Physics-based	Selective	Yes

As detailed in Section 1.2., the goal is not to replicate these tools, but rather to provide a tool to supplement early concept design. SUAVE is selective fidelity tool for concept design, similar to the goal of this work. However, SUAVE appears to direct focus towards optimising sizing, whereas this work takes a control-and-mission-centric approach.

2.4. Generalised/Unified Models

It has been highlighted that a generalised tool is required due to the high level of novelty of UAM aircraft configurations. Many current design tools rely on semi-empirical model of high-level data. This presents significant challenges for compound fixed-wing VTOL aircraft due the unconventional, complex and highly varied configurations[3]. Lee et. al.[19] developed a unified, high-fidelity flight dynamics modelling technique for eVTOL aircraft[3]. A component-based approach was taken, resulting in a model capable of analysing any aircraft, without modifying the mathematical model.

2.5. Lift Surface Modelling

Vortex Lattice Method (VLM) software such as XFLR5 is common for performing stability, control, and early aerodynamic analysis at the conceptual design stage[20-22]. VLM is an extension of Prandtl's lifting line theory applicable to a wider range of wing geometries[23]. This method is widely favoured in preliminary design stages due to its simplicity and low computational cost when compared to computationally demanding CFD methods[24]. However, while VLM provides an accurate estimate of C_L, it assumes inviscid flow, and does no account for thickness effects, leading to significant underestimation of drag[24]. Compressibility, unsteadiness, rotating frame effects, and other complex aerodynamic phenomena are also neglected, though versions of the mathematical model have been created to capture some of these effects[25].

In the conceptual design environment, designs will feature limited detail. Therefore, it is assumed that the increased accuracy of higher-fidelity software is outweighed by other concept level assumptions, and modularity for rapid prototyping is instead the priority for this design tool. Instead, key non-linear effects are captured by semi-empirical modelling (see Section 2.6.2). Hence, VLM is used here.

2.6. Rotor Modelling

Detailed models such as VLM or vortex particle methods may be employed for rotor modelling.[24,26] Lower-fidelity methods include Blade Element Momentum Theory (BEMT), Blade Element Theory (BET) or Actuator Disk Theory (ADT)[27]. Similar to VLM, the assumption is that not enough detail is available at the concept design stage to employ higher fidelity methods, hence ADT is applied here. There is a wealth of rotor modelling software available, where the aircraft states and thrust requirements from this work may be input.

2.7. Control and Trim Optimisation

To obtain control inputs required to perform manoeuvres, some kind of controller must be introduced. Due to the combination of fixed-wing and rotorcraft control inputs, transition presents and over-actuated optimisation problem[28,29]. D. Sagan et. al[28] constructed a 3DoF dynamics model for a notional VTOL aircraft and applied gradient-based optimisation methods to allocate control, with the aim of minimising the power required for trim. It this work it was found that the parameter space featured several local minima, necessitating further development of the optimisation problem formulation[28]. The problem of multiple global minima is often solved by either bounding the problem or employing global optimisation methods. MATLAB has suite of tools for this purpose (Global Optimization Toolbox).

A. C. Daud Filho et. al.[29] employed the Sequential Simplex algorithm, a gradient free method to derive trim points through a tiltwing transition manoeuvre. The work then went on to linearise about these trim points and create a reference scheduler, a type of gain scheduled control, to simulate transition behaviour. The gains were tuned using LQR, a form of optimal control. There are several control methodologies applied to eVTOLs including, but not limited to PID (Proportional Integral Derivative), LQR (Linear Quadratic Regulator), LQI (Linear Quadratic Integral), LQG (Linear Quadratic Gaussian) and MPC (Model Predictive Control). M. Okasha compared the response of a mini-drone under various stabilising controllers[30]. It was found that LQR both provided the best response and was the least effort to tune. PID does not offer a unified, systematic way to optimise system performance and often requires manual tuning[30]; it is not easily automated.

3. Pre-processing

The pre-processing module involves defining the geometry and carrying out aerodynamic analysis From this section onwards, all functions, Simulink models, and class definitions developed may be found here: https://github.com/JemimaPoynton/FYP-eVTOL-Mission-Profile-Generation. The overall module structure is shown in **Figure 4**. The workflow is designed such that outside software may be used to supplement the functionality at any stage, with minimal modifications.

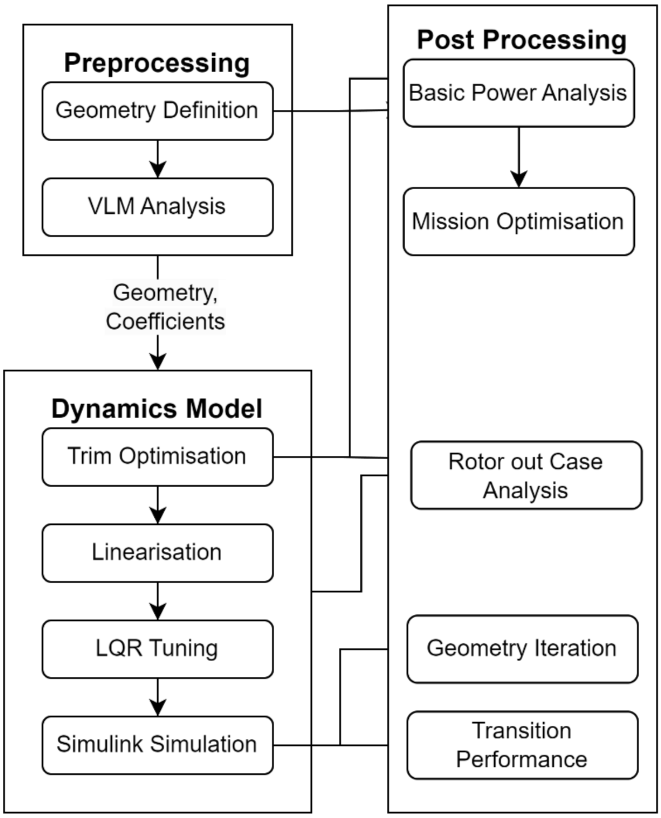


***Figure 4:** Top-level functional diagram.*

3.1. Object Oriented Component-Based Approach

A component-based object-oriented approach is adopted. Object-oriented programming enables a high level of modularity and improves data handling throughout the program. Classes provide a restricted structure that ensures the reusability of functions and collates all relevant data in one structure, reducing the number of inputs to functions. This structure is shown in **Figure 5**.

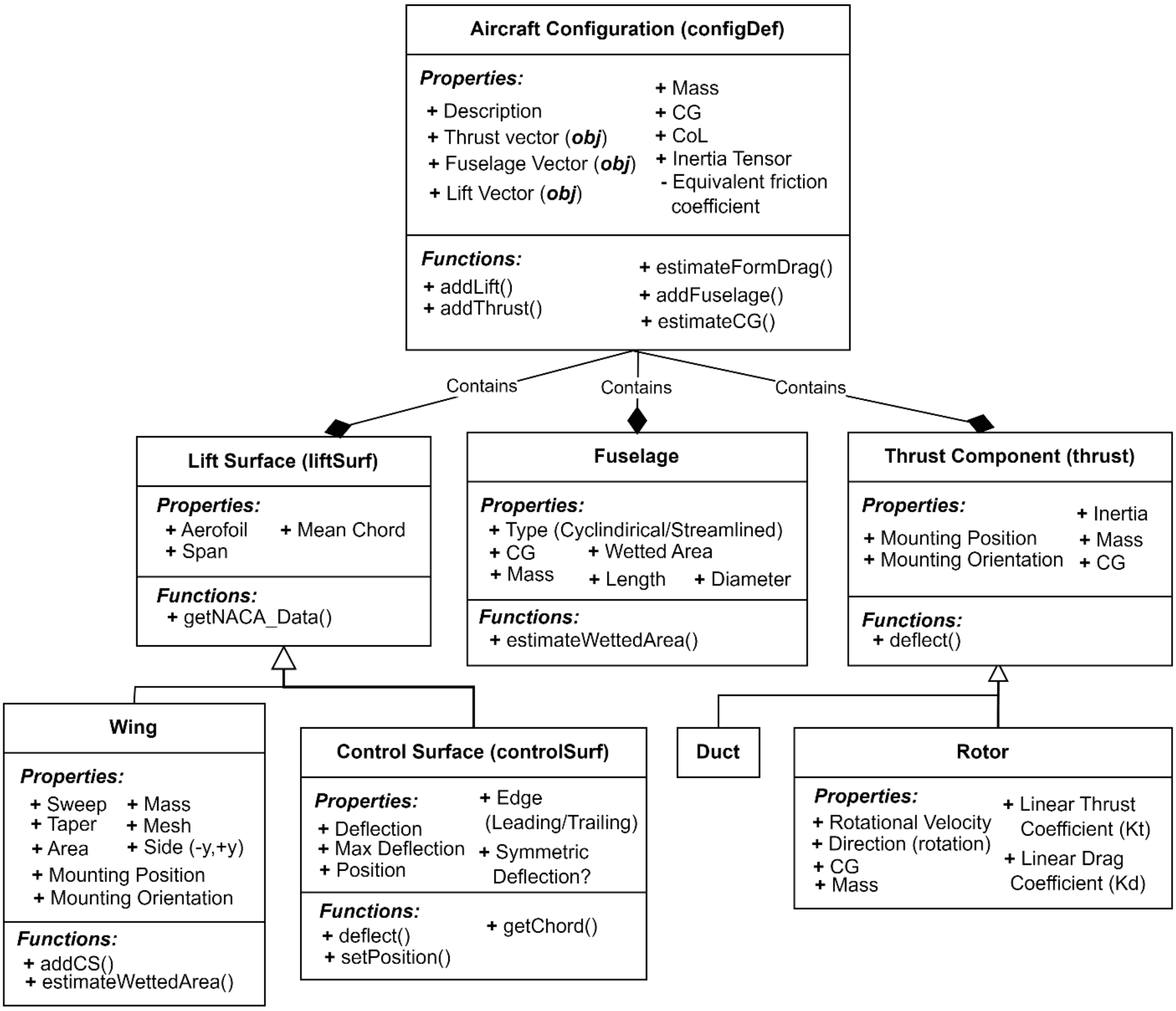
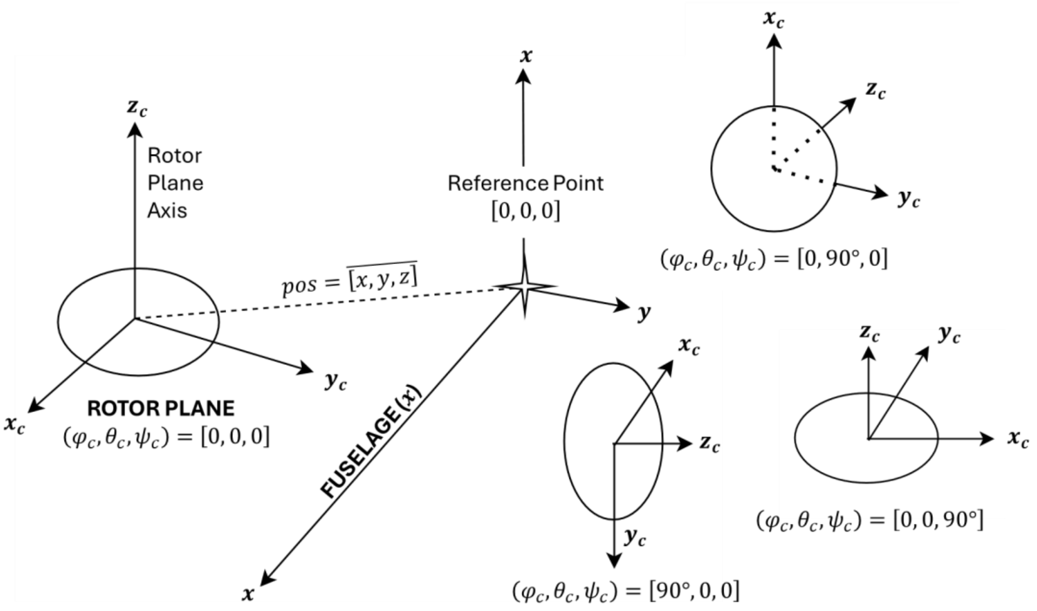


Figure 5: *Component class diagram. The class definitions/variable names have been simplified for readability. 'Duct' is currently a placeholder.*

Included in the class definitions are various standard equations for early estimations of required parameters, such as profile drag. These are based on semi-empirical estimates detailed in [31,32].

3.2. Geometry Definition

The unified component-based approach builds upon the work in [19]. The orientation of each component is characterised by the relative orientation of its plane to the main body frame (defined by the fuselage), and $[x\ y\ z]$ coordinates on the plane. The orientation is given in the form $[\phi\ \theta\ \varphi]$, corresponding to angles about the body coordinate system $[x\ y\ z]$. When representing a wing, θ is the angle of incidence, and ϕ is the dihedral. These are stored in the properties '[obj].ang' and '[obj].pos'. **Figure 6** illustrates the mounting coordinates for a rotor.



z_c
Rotor Plane Axis
x
Reference Point [0, 0, 0]
x_c
z_c
y_c
$(\varphi_c, \theta_c, \psi_c) = [0, 90°, 0]$
$pos = \overline{[x, y, z]}$
y
x_c
y_c
ROTOR PLANE
$(\varphi_c, \theta_c, \psi_c) = [0, 0, 0]$
FUSELAGE(x)
z_c
y_c
z_c
x_c
x_c
y_c
$(\varphi_c, \theta_c, \psi_c) = [0, 0, 90°]$
x
$(\varphi_c, \theta_c, \psi_c) = [90°, 0, 0]$

Figure 6: *Rotor orientation and mounting coordinate system.*

The transformation matrix between the component and body frame is therefore given by[19]

$$\boldsymbol{R_b^c} = L_1 L_2 L_3 \tag{3.2.?}$$

$$L_1(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi_c) & \sin(\varphi_c) \\ 0 & -\sin(\varphi_c) & \cos\,(\varphi_c) \end{bmatrix} \tag{3.2.?}$$

$$L_2(\theta) = \begin{bmatrix} \cos(\theta_c) & 0 & -\sin(\theta_c) \\ 0 & 1 & 0 \\ \sin(\theta_c) & 0 & \cos\,(\theta_c) \end{bmatrix} \tag{3.2.?}$$

$$L_3(\psi) = \begin{bmatrix} \cos(\psi_c) & \sin(\psi_c) & 0 \\ -\sin(\psi_c) & \cos(\psi_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2.?}$$

The model includes two different body coordinate systems: Geometric, and Dynamic. These are shown in **Figure 7**. The z and x axis are flipped. This is compensated throughout the program, but the user should be aware when defining input and output data. This is aided by visualisation of geometry. This arises from the 6DoF Euler block in Simulink requiring z is taken toward the earth reference[33], however when defining geometry, it is more intuitive to take z positive vertically.

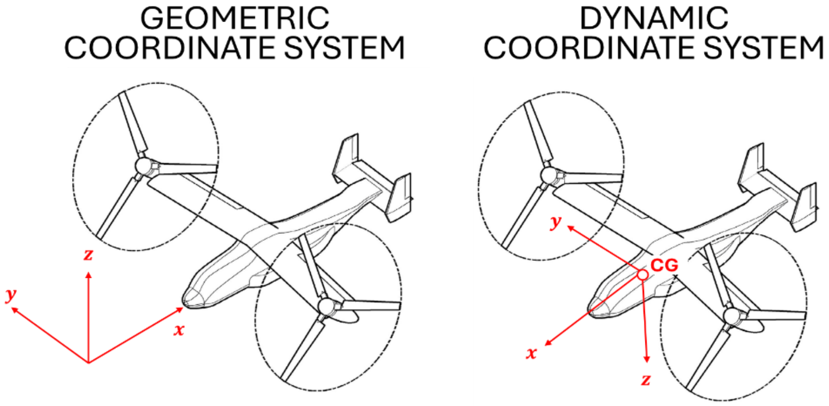


Figure 7: *Geometric and dynamic coordinate systems. Aircraft image source: [34].*

The properties defining wing and rotor geometry are given in **Figure 8**. Wing dihedral is represented by θ_c, and the mounting position is taken as the inboard trailing edge.

The geometry definition of lifting surfaces is limited to a single partition and aerofoil due to time constraints. Model complexity and functionality in terms of dynamics and control were prioritised instead. VLM is such that this is easily expanded by simply modifying the mesh function to obtain the relevant lattice points from the geometric properties, without changing any other aspect of the program. 'plotConfiguration' may be used to visualise a 'configDef' object as in **Figure 8:** Visualisation of aircraft geometry using 'plotConfiguration' function.

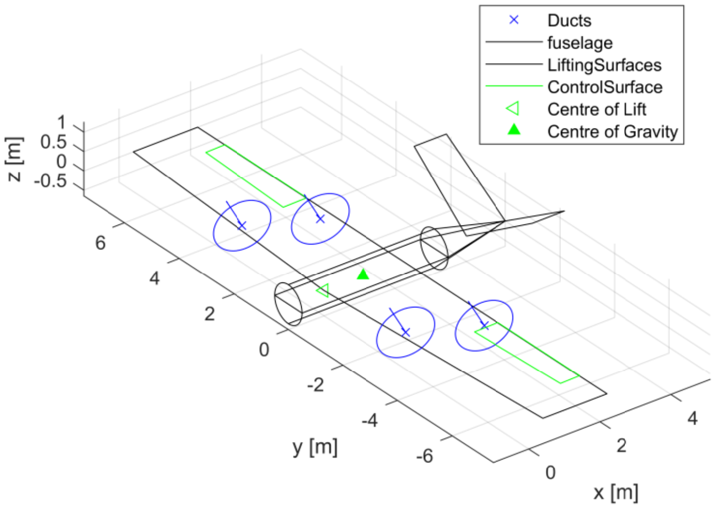


Figure 8: *Visualisation of aircraft geometry using 'plotConfiguration' function.*

3.3. Characterising Cruise Aerodynamics (VLM)

Currently, the available VLM software packages are a mix of open-source and commercial programs. XFLR5 is one example of a widely used open-source product, however, the wide capabilities of such programs can often inherently introduce a learning curve. Much of the available software is self-contained and difficult to directly integrate into the program. Vortex lattice method is also independent of the number of components and their position, unlike typical purely theoretical models.

A promising open-source software is Tornado, developed by Tomas Melin[35], however, it is heavily integrated into a UI, making automation within the class-defined system difficult and time consuming to implement. Hence, a version of VLM has been written to facilitate ease of use and rapid concept development, however, derivatives from more specialised or higher-level software may be manually input as required (see Section 4, **Figure 16**). The current VLM module is intended as a base framework, such that inputs and outputs are compatible with the whole tool, rather than a fully developed software package.

3.3.1. Implementation

Figure 9 illustrates the underlying concept of VLM. The lifting surface is represented as a grid of panels on which horseshoe vortices are superimposed[36]. The velocities induced by each vortex at a control point P, or n_i (see Figure), can then be calculated using Biot-Savart Law to derive aerodynamic forces and moments[23,36].

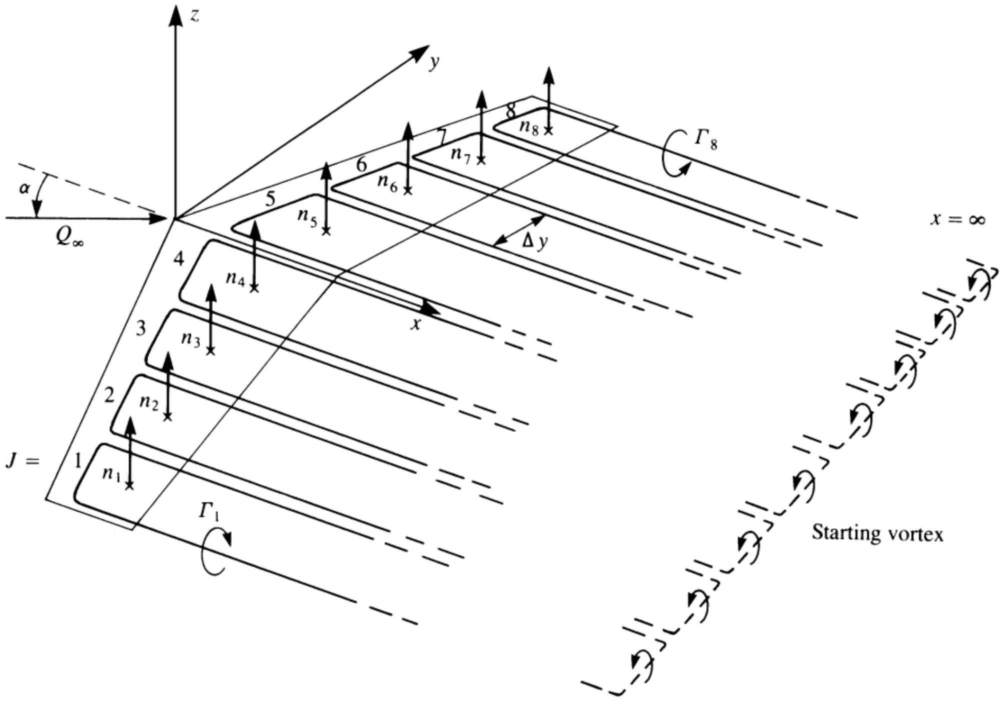


Figure 9: *Horseshoe vortex lattice model for solving the lifting-line problem. Source: [36].*

The velocity induced ($\vec{V}$) by a vortex filament of strength Γ_n is given by the Biot-Savart law as[23,35]

$$\overrightarrow{dV} = \frac{\Gamma_n(\overrightarrow{dl} \times \vec{r})}{4\pi r^3} \tag{3.3.1}$$

where r is the distance from P to the vortex filament, and dl is the length of the vortex filament. The solution of 3.3.1 yields a simpler equation to describe the induced velocity for a vortex segment of arbitrary length[23,35,37]

$$\vec{V} = \frac{\Gamma_n}{4\pi} \frac{\overrightarrow{r_1} \times \overrightarrow{r_2}}{|\overrightarrow{r_1} \times \overrightarrow{r_2}|^2} \left[\overrightarrow{r_0} \cdot \left(\frac{\overrightarrow{r_1}}{r_1} - \frac{\overrightarrow{r_2}}{r_2}\right)\right] \tag{3.3.2}$$

where $\overrightarrow{r_0}, \overrightarrow{r_1}$ and $\overrightarrow{r_2}$ are vectors as in **Figure 10**.

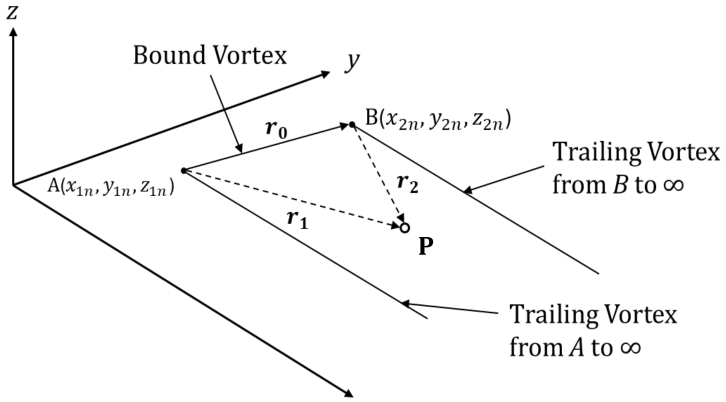


Figure 10: *Horseshoe vortex. Adapted from: [23,37].*

The total induced velocity at a control point, m, due to the horseshoe vortex representing the n^{th} panel is given by[23,37]

$$\begin{aligned}\overrightarrow{V_{m,n}} &= \overrightarrow{V_{AB}} + \overrightarrow{V_{A\infty}} + \overrightarrow{V_{B\infty}} \\ &= \overrightarrow{A_{m,n}}\Gamma_n\end{aligned} \tag{3.3.3}$$

where $\overrightarrow{A_{m,n}}$ is the influence coefficient. ∞ is represented by a point far downstream. This is denoted by C and D within the program. The total induced velocity at the m^{th} control point is therefore[23]

$$\overrightarrow{V_m} = \sum_{n=1}^{N} \overrightarrow{A_{m,n}}\Gamma_n \tag{3.3.4}$$

where N is the total number of panels. Applying boundary conditions forms a matrix of simultaneous equations to solve for[35,36] Γ_n

$$\begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & \vdots \\ w_{2,2} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & w_{m,n} \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \vdots \\ \vdots \\ \Gamma_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{bmatrix} \tag{3.3.5}$$

where b is the flow through each panel as a function of flight condition. w is the flow from each vortex to each panel. b_n is given by[26]

$$b_n = V_\infty \begin{bmatrix} \cos(\alpha)\cos(\alpha) \\ \cos(\alpha)\cos(\alpha) \\ \sin(\alpha) \end{bmatrix} + \omega \times r \tag{3.3.6}$$

where r is the distance from the control point to the CG. The force on the panel can then be calculated by applying the Kutta-Jukovski theorem[35]

$$F_b = \rho(\overline{V} \times \overline{\Gamma})l \tag{3.3.7}$$

where the aerodynamic forces and moments are resolved as

$$F_w = \begin{bmatrix} D \\ Y \\ L \end{bmatrix} = \boldsymbol{R_b^w} F_b \qquad M_{cg} = \begin{bmatrix} l \\ m \\ n \end{bmatrix} = \sum_{n=1}^{N} l_m \times dF_b \tag{3.3.8}$$

l_m is the moment arm from control point to reference point, and l is the length of the vortex filament. $\boldsymbol{R_b^w}$ is a rotation matrix from the body to the wind frame (see Appendix D). A sweep about the reference state is then performed to estimate linear aerodynamic and stability derivatives. The aerodynamic derivatives are also output for each individual surface to estimate structural loading.

Equations 3.3.2 to 3.3.8 are applied to a mesh, with the mesh density being specified by the user. **Figure 11** shows an example mesh. Wings with control surfaces are automatically partitioned either side of the flapped section; the panels are proportionally distributed, and sizing adjusted accordingly.

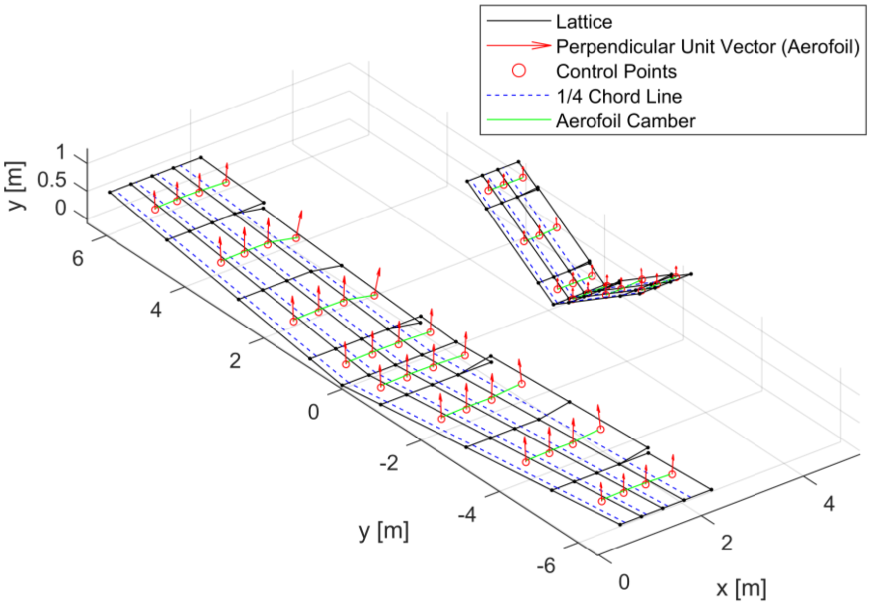


Figure 11: *Vortex lattice mesh for a simple wing and tail with deflected control surfaces. The key analysis points for Vortex Lattice Method (VLM) are marked.*

To allow for multiple surfaces of varying size, the mesh is initially defined by the base coordinates of each separate partition, but then reshaped such that the relationship between points is linearly mapped, as in [35,36]. This approach is beneficial to understanding the geometry of the mesh but may be an area of improvement for the computational efficiency of the method.

A key consideration when applying VLM is the convergence rate with the density of the mesh. For simple geometries a small number of panels is usually sufficient, but the user should evaluate the mesh density where a high level of accuracy is requires, or where geometry is complex. A convergence plot for the geometry in **Figure 11** is shown in **Figure 12**.

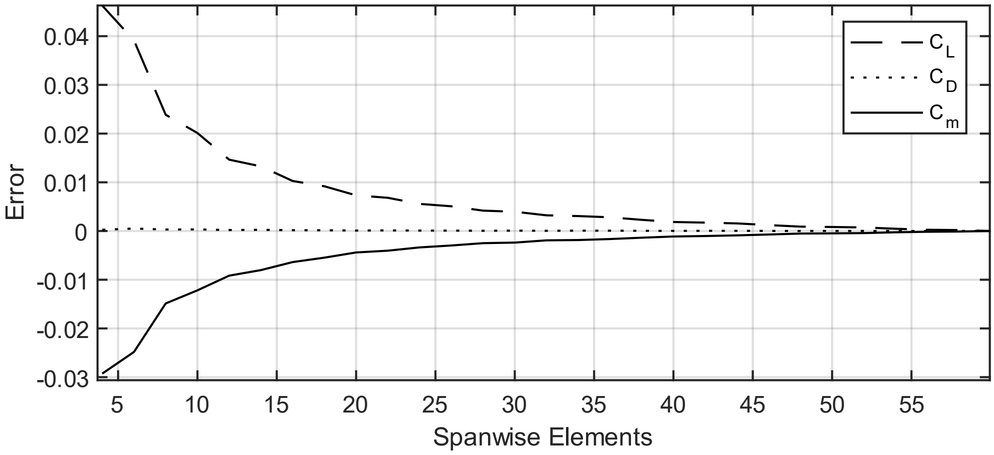


Figure 12: *Convergence of vortex lattice module.*

A similar figure may be used to assess the elements required to achieve the desired accuracy. The convergence is not purely exponential for low mesh density due the panels being proportionally distributed either side of the control surface boundary. This leads to dynamic mesh sizing.

3.3.2. Validation

To verify the accuracy of the VLM module, several standard geometries were compared against established software. A simple example is shown in **Figure 13**. The reference point is taken as [0.5 0 0]. Total spanwise and chordwise elements of 12 and 6 have been used in all cases, inclusive of control surfaces. **Table 2** summarises the results for key derivatives.

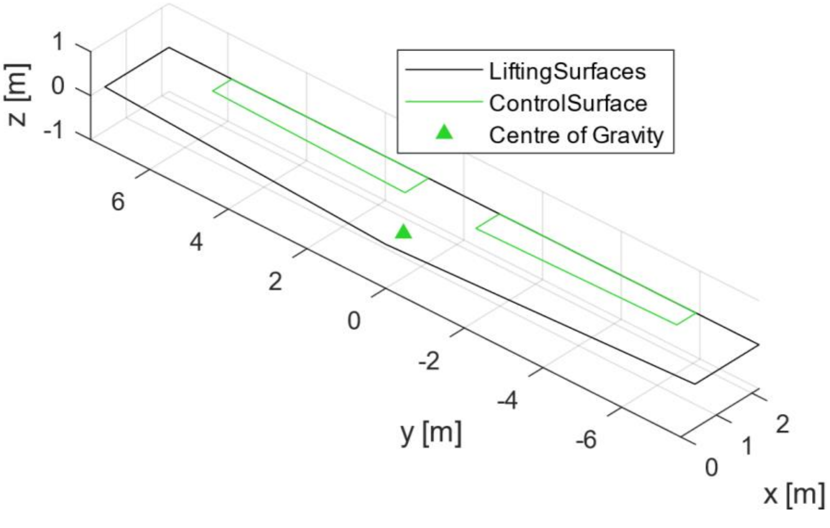


Figure 13: *Simple wing for VLM analysis*

Table 2: *Comparison of VLM Module with Commercial Software. The AVL and VortexLattice data is sourced from [38].*

Coefficient	VortexLattice[38]	AVL[38]	Tornado[35]	VLM Module	XFLR5
C_{L_α}	4.6632	4.6588	4.6917	4.6657	4.6696
C_{D_α}	-	-	0.0315	0.0355	0.0255

C_{Y_β}	0.00000	-0.00002	0.00000	0.00002	0.00001
C_{m_α}	-0.3957	-0.3978	-0.3955	-0.3983	-0.3982
C_{L_q}	5.6508	5.6494	5.6467	5.6920	-
C_{l_p}	-0.5239	-0.5248	-0.5314	-0.5421	-
C_{m_q}	-1.2478	-1.2702	-1.216	-1.2809	-
C_{n_p}	-0.01918	-0.01918	-0.0033	-0.0239	-
$C_{L_{\delta c}}$	-	-	2.2731	2.1465	-
$C_{m_{\delta c}}$	-	-	-0.5613	-0.6125	-

There is overall good agreement when compared with alternative VLM software. Errors may arise from differences in mesh distribution method, or slight differences in computational approach. Additional comparison with XFLR5 is included in Appendix A, along with the sample UAM aircraft.

However, the functionality has only been carried out for simple geometries. Further development of the VLM module is necessary but is not included here as focus is instead directed towards mission dynamics and control and building the overall framework. This is a recognised limitation. The results have been validated within the bounds of that used in this work. The aerodynamic assumptions inherent to the VLM have been discussed in Section 2.5.

3.4. Reference UAM Aircraft

To provide a reference for assessing and validating the generated data, and analyse a range of tiltrotor designs, 2 reference aircraft have been defined. These are based on designs and data found in literature ([39,40]). These have been selected to represent a variety of geometries, and 2 differently scaled applications. It should be noted that the geometry cannot be precisely represented within the constraints of the program as the data available is taken from the end of concept design.

UAM 1 is adapted from [40]. It is a highly manoeuvrable, high endurance blended-wing-body aircraft applicable for cargo delivery or connectivity applications. The original design has been modified to include two forward tilting rotors. UAM 2 is adapted from [39]. UAM 2 represents a passenger eVTOL with 6 rotors, the front two of which tilt forward during transition.

These are not exhaustive, but capture the key features included in the model. The aerodynamic properties have been kept as close as possible to the original sources within the restrictions of the VLM module to ensure functional concept designs (e.g. trim capability). It is assumed that a user will have made fundamental early concept-level assessments of stability and controllability, prior to the tool. These configurations are used throughout to demonstrate functionality of each module and qualitatively compare their performance. Section 6 then includes a detailed analysis of UAM 2, with concept iteration. Where aerodynamic characteristics are missing, they have been calculated using VLM, alongside the methods described in Section 3.1. The full dataset is included in Appendix A.

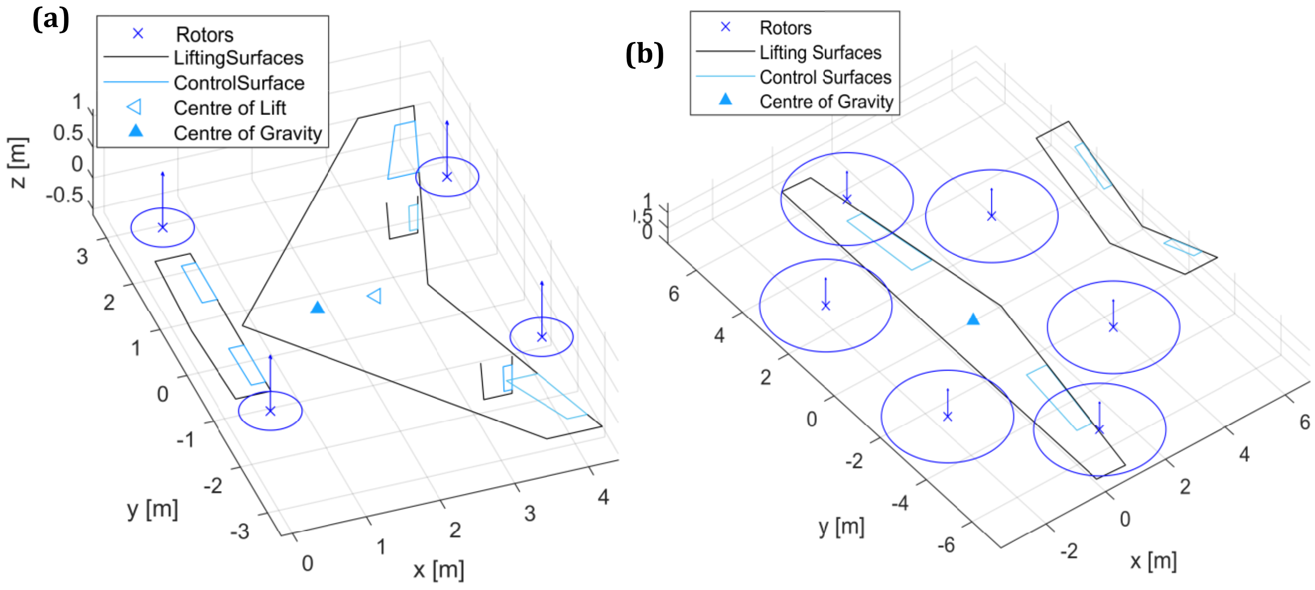


Figure 14: *Reference aircraft referred to as (a) UAM 1 and (b) UAM 2.*

4. Simulink Dynamics Model

Simulink is used to provide an intuitive model interface. Additionally, use of Simulink facilitates control system design which is a common requirement in conceptual design and simulation of eVTOLs, particularly in the area of UAVs.

The model interface is shown in **Figure 15**.

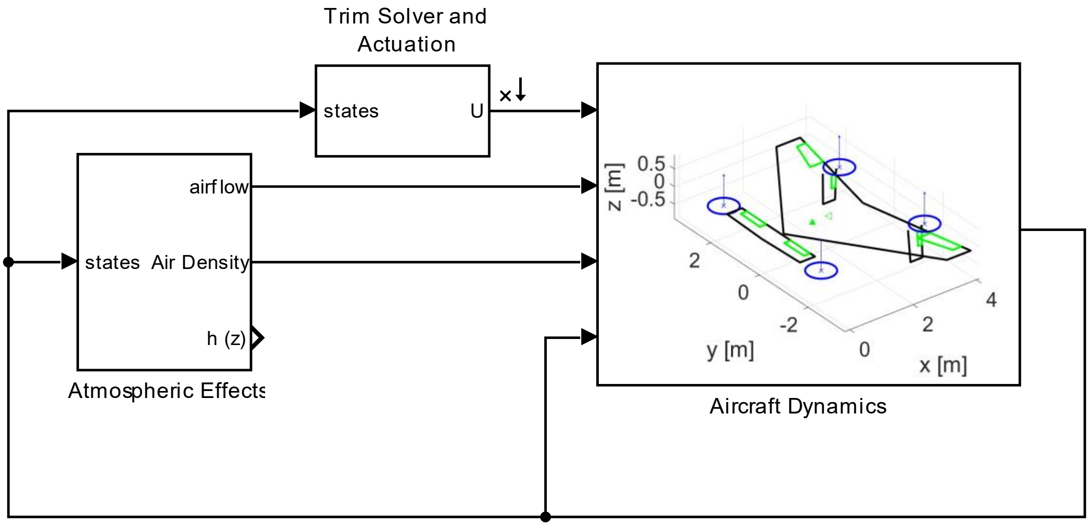
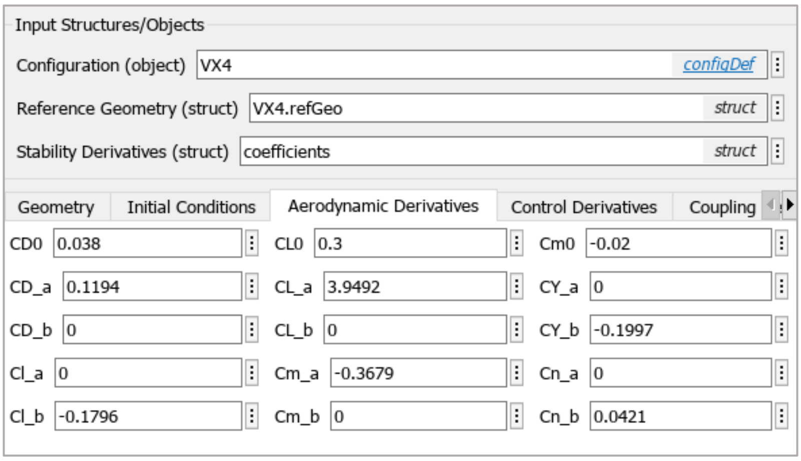


Figure 15*: Aircraft dynamics and control Simulink model (Top level).*

There are 3 key modules. Dynamics, Control, and Atmosphere. The dynamics module applies Equations 4.2.1. to 4.2.? to the aircraft specified within the mask, plotting the geometry as a visual aid. Control is a masked variant subsystem containing the control systems described in Section 4.3. The Atmosphere block is a variant subsystem allowing for inclusion of altitude effects (density), turbulence, and gusts by selection.

Figure 16 shows the masked ‘Aircraft Dynamics’ subsystem interface.



Input Structures/Objects

Configuration (object)	VX4	configDef
Reference Geometry (struct)	VX4.refGeo	struct
Stability Derivatives (struct)	coefficients	struct

CD0	0.038	CL0	0.3	Cm0	-0.02
CD_a	0.1194	CL_a	3.9492	CY_a	0
CD_b	0	CL_b	0	CY_b	-0.1997
Cl_a	0	Cm_a	-0.3679	Cn_a	0
Cl_b	-0.1796	Cm_b	0	Cn_b	0.0421

Figure 16: *'Aircraft Dynamics' masked subsystem interface.*

The interface includes two sections, one for bounding objects/structures in the workspace to parameters in the model, and one for parameter inputs. As discussed in Section 3.1, this allows for effective handling of a large amount of data, high modularity, and reusability by adhering to a standard format. The mask does, however, allow manual editing of parameters by simply leaving the sections blank. This may arise from the use of outside software, or general concept design iteration. MATLAB objects are not compatible with Simulink’s codegen, however, the functions stored in the objects are not required to run the dynamics module, so callback routines within the masks are used to extract the relevant data. The input objects can then be used again for post-processing.

A similar issue arises with the use of structures. This is circumvented by converting structures to buses, or directly assigning single variables as parameters in the MATLAB function block. Some buses are required to change size to allow any number of components to be input, which is handled by changing the bus definition ‘coeff’ and ‘thrust’ within the callback functions. These definitions are stored in ‘buses.mat’.

4.1. Aircraft Dynamics

4.1.1. Linearised 6DoF Equations of Motion

The aerodynamic and stability behaviour (in the linear region) may be represented by coefficients derived by VLM. This also serves to generalise the model. The forces and moments in the body frame and about the CG are given by[40-43]

$$F_b = F_a + F_g + F_T \tag{4.1.1}$$

$$M_b = M_a + \tau_T \tag{4.1.2}$$

where F_a, F_g, and F_T are the aerodynamic, gravity and thrust forces in the body frame. M_a and τ_T are the aerodynamic moments and torque due to thrust, respectively. The aerodynamic forces are given by[40-43]

$$\boldsymbol{F_a} = \begin{bmatrix} \boldsymbol{F_x} \\ \boldsymbol{F_y} \\ \boldsymbol{F_z} \end{bmatrix} = \boldsymbol{R_w^b} \begin{bmatrix} \boldsymbol{-D} \\ \boldsymbol{Y} \\ \boldsymbol{-L} \end{bmatrix} \tag{4.1.3}$$

where $\boldsymbol{R_w^b}$ is the transformation matrix from forces in the wind frame to the body axis given by[40-43]

$$\boldsymbol{R_w^b} = \begin{bmatrix} \cos(\alpha)\cos(\beta) & -\cos(\alpha)\sin(\beta) & -\sin(\alpha) \\ sin(\beta) & \cos(\beta) & 0 \\ \cos(\beta)\sin(\alpha) & -\sin(\alpha)\sin\,(\beta) & \cos(\alpha) \end{bmatrix} \tag{4.1.4}$$

The aerodynamic forces are calculated using the derivatives from the VLM module as[40-43]

$$\begin{bmatrix} D \\ Y \\ L \end{bmatrix} = \frac{1}{2}\rho V_a^2 S \begin{bmatrix} C_D \\ C_Y \\ C_L \end{bmatrix} \tag{4.1.5}$$

$$\begin{bmatrix} C_D \\ C_Y \\ C_L \end{bmatrix} = \begin{bmatrix} C_{D0} + C_{D_\alpha}\alpha + \sum_{i=1}^{n} C_{D\delta c_i}\delta_{c_i} + C_{D_q}\frac{c}{2V_a}q \\ C_{Y_0} + C_{Y_\beta} + C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + \sum_{i=1}^{n} C_{Y\delta c_i}\delta_{c_i} \\ C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{c}{2V_a}q + \sum_{i=1}^{n} C_{L\delta c_i}\delta_{c_i} \end{bmatrix} \tag{4.1.6}$$

where the C_{a_b} represents the change in coefficient C_a due to b. The full definitions may be found in the List of Symbols. α and β are the angle of attack and sideslip angle, respectively. p, q and r are pitch, yaw and roll rate. c and b are the reference chord and span. V_a is the net airspeed. δ_{c_i} is the i^{th} control surface deflection.

The aerodynamic moments are given by[40-43]

$$\begin{bmatrix} l \\ m \\ n \end{bmatrix} = \frac{1}{2}\rho V_a^2 S \begin{bmatrix} bC_l \\ cC_M \\ bC_n \end{bmatrix} \tag{4.1.7}$$

$$\begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} = \begin{bmatrix} C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a} + C_{l_r}\frac{b}{2V_a} + \sum_{i=1}^{n} C_{l\delta c_i}\delta_{c_i} \\ C_{M_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{c}{2V_a}q + \sum_{i=1}^{n} C_{m\delta c_i}\delta_{c_i} \\ C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + \sum_{i=1}^{n} C_{n\delta c_i}\delta_{c_i} \end{bmatrix} \tag{4.1.8}$$

F_g is given by[40-43]

$$F_g = m\begin{bmatrix} -g \cdot \sin(\theta) \\ g \cdot \cos(\theta)\sin(\varphi) \\ g \cdot \cos(\theta)\, cos(\varphi) \end{bmatrix} \tag{4.1.9}$$

The rotor forces and moments are defined by

$$F_T = \boldsymbol{R_b^r}\begin{bmatrix} 0 \\ 0 \\ -\sum_{m=1}^{m} T_m \end{bmatrix} \tag{4.1.10}$$

where $\boldsymbol{R_b^r}$ is the transformation matrix from the rotor to body plane (Equation X), and T_m is the thrust of the m^{th} rotor, given by T_m =$k_t\omega_m^2$, where k_{t_m} is a constant. The rotor induced moment is given by[40,44,45]

$$\tau_T = \boldsymbol{R_b^r}\begin{bmatrix} 0 \\ 0 \\ \sum_{m=1}^{m} k_{b_m}{\omega_m}^2 \end{bmatrix} + F_T \times l \tag{4.1.11}$$

l is a vector containing the moment arm in $[x, y, z]_b$. Linear variables k_t and k_b are used for computational efficiency. The thrust requirement, T_m, is then recalculated for power estimation. This is deemed acceptable as these variables effectively serve to represent the balance between the ω differential required for yaw control, and the overall thrust. Yaw control requirements contribute minimally to the total power required for VTOL.

Equations of motion can then be used to derive the state derivatives and characterise the behaviour over time. The state vector x takes the form $[u, v, w, p, q, r, \varphi, \theta, \psi]^T$, or $[V_b, \omega, \Theta]^T$. V_b is the velocity in the body frame in $[x\ y\ z]$. Similarly, ω represents the rotational velocities, and Θ contains the Euler angles in roll, pitch, and yaw.

$\dot{V_b}$ is derived from the force equation, given as[33,41]

$$\dot{V_b} = \frac{1}{m}F_b - \omega \times V_b \tag{4.1.12}$$

$\dot{\omega}$ is derived from the moment equation[33,41]

$$\dot{\omega} = I_b^{-1}[M_b - (\omega \times I_b\omega)] \tag{4.1.13}$$

where I_b is the inertia tensor. Θ is taken in the Earth frame calculated using the navigation equation[33,41]

$$\dot{\Theta} = \boldsymbol{W} \cdot \omega \tag{4.1.14}$$

Similarly, the position of the UAM in the earth frame (p_e) is given by[33,41]

$$\dot{p_e} = \boldsymbol{R_b^e}\, V_b \tag{4.1.15}$$

$\boldsymbol{R_b^e}$ and $\boldsymbol{W}$ are transformation matrices. These are defined in Appendix C.

A function has been created to model equations 4.1.1 to 4.1.15. ('Dyn.m'). This function may be applied to any aircraft geometry that follows the class definitions shown in Figure X. However, highly novel aircraft may exhibit complex dynamics or aerodynamic effects that cannot be accounted for by these equations. To maintain modularity and enable future development, the dynamics function is passed into other key functions anonymously. This has the advantage that the function may be modified/replaced, provided only that the key inputs and outputs (F_b, M_b) take the same form.

4.1.2. Non-Linear Effects

Section 4.1.1 assumes linear aerodynamics. However, there are several non-linear effects[46] that are especially critical for tiltrotor design. A common approach to implementing non-linear aerodynamics is the use of look-up-tables. However, this has a high reliance on either experimental or CFD modelling data. One approach considered was to use several data points generated by the VLM module, however, VLM is does not account for viscous effects and tends to underestimate drag, thus yielding little additional accuracy.

Expanding upon Equation 4.2.?, the effects of stall can be incorporated into the lift coefficient by blending the linear stability derivative model with a nonlinear function such that[42,47]

$$C_L(\alpha) = \big(1 - \sigma(\alpha)\big)\big[C_{L_0} + C_{L_\alpha}\alpha\big] + \ \sigma(\alpha)[2\text{sign}(\alpha)\sin^2(\alpha)\cos(\alpha))] \tag{4.1.16}$$

where[42,47]

$$\sigma(\alpha) = \frac{1 + e^{-M(\alpha-\alpha_0)} + e^{M(\alpha+\alpha_0)}}{\left(1 + e^{-M(\alpha-\alpha_0)}\right)\left(1 + e^{M(\alpha+\alpha_0)}\right)} \tag{4.1.17}$$

where M and α_0 are positive constants. This resolves to the behaviour of a flat plate at high α.[42] The aircraft is assumed to act as a flat plate perpendicular to the flow for a very high angles of attack, i.e. $w \gg u$. This is a common assumption in conceptual eVTOL flight dynamics modelling[42refs]. This is implemented by setting C_{D0} as 1.28[48] for high values of α ($\alpha > \alpha_{stall}$).

During transition, the rotors downwash flow creates aerodynamics interference on the wings, forming a "fountain effect" at low speeds[49-51]. This effect may be modelled by first dividing the wing into two sections: the slipstream zone effected by the rotor wake, and the free flow zone[50,52]. The area of the slipstream may be approximated as[52]

$$S_{ss} = \begin{cases} S_{ss0} + \dfrac{2R_w c_w - S_{ss0}}{60}\theta_r, & 0 \leq \theta_r \leq 60° \\ 2R_w c_w, & \theta_r \geq 60° \end{cases} \tag{4.1.18}$$

where S_{ss0} is the area of the slipstream in hover. This is a linear approximation derived from CFD in [52]. This relationship may not be accurate for all configurations, and instead [46], [53], and [50] may be consulted, however these also rely on semi-empirical relationships. Rotor wake contraction is complex to model[ref], and this is intended only to give a qualitative indication of the impact on dynamics. The contraction radius (R_w) of the rotor wake is approximated by[51-53]

$$R_w = R(0.78 + 0.22e^{(-0.3-2L_n\sqrt{C_T}-60C_T)}) \tag{4.1.19}$$

θ_r is the forward rotation of the rotor. L_n is the length of the nacelle, or distance of the rotor from the wing, nondimensionalised with respect to rotor radius[53]. C_T is the thrust coefficient, and R is the radius of the rotor This approach is illustrated in **Figure 17**[52].

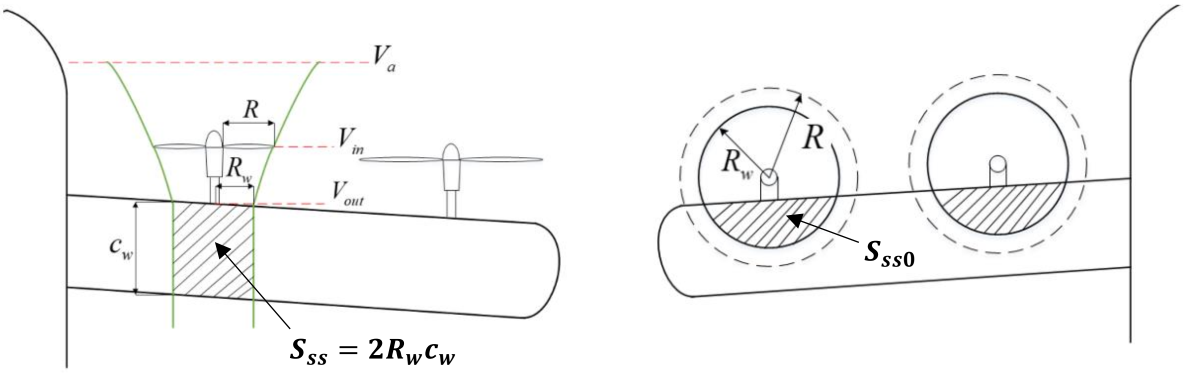


Figure 17: *Schematic diagram of slipstream area if fixed-wing (left) and copter (right) modes. Source: [52]. Annotations have been added.*

The induced velocity of the rotor on the wing is given by[52]

$$v_{ss} = \sqrt{\frac{2T}{\rho\pi {R_w}^2} + (V_a\cos(\alpha)\sin(\theta_r))^2} - V_a\cos(\alpha)\sin(\theta_r) \tag{4.1.20}$$

The resulting airflow velocity in the slipstream zone may be obtained by the superposition of v_{ss} and the aircraft airspeed (V_a)[52]. The forward and vertical velocities in the slipstream are calculated as[50,52,53]

$$\begin{aligned} V_{xss} &= v_{ss}\sin(\theta_r) + \ V_a\cos(\alpha) \\ V_{zss} &= -v_{ss}\cos(\theta_r) + \ V_a\sin(\alpha) \end{aligned} \tag{4.1.21}$$

The angle of attack and central airspeed in the slipstream area is therefore given by[52]

$$\alpha_{ss} = \tan^{-1}\left(\frac{V_{zss}}{V_{xss}}\right) \qquad V_{a_{ss}} = \sqrt{{V_{xss}}^2 + {V_{xss}}^2} \tag{4.1.22}$$

Note that sideslip is not included here. Including the effect of the slipstream, the longitudinal aerodynamic forces and moments become[51,52,54]

$$\begin{aligned} L &= \sum_{j=1}^{m} \frac{1}{2}{V_{a_{ss\,j}}}^2 C_L\left(\alpha_{ss_j}\right) S_{ss\,j} + \frac{1}{2}{V_a}^2 C_L(\alpha)(S - S_{ss}) \\ D &= \sum_{m=1}^{m} \frac{1}{2}{V_{a_{ss_j}}}^2 C_D\left(\alpha_{ss_j}\right) S_{ss_j} + \frac{1}{2}{V_a}^2 C_D(\alpha)(S - S_{ss}) \\ m &= \sum_{m=1}^{m} \frac{1}{2}{V_{a_{ss_j}}}^2 C_m\left(\alpha_{ss_j}\right) S_{ss_j}c + \frac{1}{2}{V_a}^2 C_m(\alpha)(S - S_{ss})c \end{aligned} \tag{4.1.23}$$

where m is the number of rotors. Equation 4.2.? is applied to the results of Equation 4.2.?.

Passenger-carrying UAM tiltrotors with large nacelles may experience a change in CG and moment of inertia during transition[50]. This change is given by[50]

$$\Delta x = \frac{m_{NAC} R_H \sin(\theta_r)}{m} \tag{4.1.24}$$

$$\Delta z = \frac{m_{NAC} R_H (1 - \cos(\theta_r))}{m}$$

where m and m_{NAC} are the mass of the nacelle, and total mass of the aircraft, respectively. R_H is the rotor's height above the wing. θ_r is the rotor deflection. Where data is unavailable, this may be excluded by setting the rotor mass to zero. Representing the change of inertia requires estimation of the inertia coefficient[50] which is unlikely to available at the early concept stage, hence this is excluded.

4.1.3 Model Assumptions and Limitations

Not currently included are the detailed aerodynamic interactions between the rotors as modelled in [46], or obtained from CFD. The stall behaviour is simplified, however, it assumed that the model with remain largely in the linear region, with the stall behaviour instead acting as a bound on the optimisation problem. The current model is limited to tiltrotor or tilt-duct configurations due to limitations in the availability of generalised aerodynamic modelling derivatives. The model is limited to eVTOL designs employing rotors or ducts, however, this is only due to limitations in the availability of aerodynamic modelling methods. If derivatives are obtained, or equations for the change in lift and drag with wing tilt are available, this is easily integrated by modifying the lift calculation in 'function'. Complex rotor effects such as the Vortex Ring State (VRS) are not accounted for.

4.3. Trim and Actuation

The UAM transition stage is both highly non-linear and presents an over-actuated system due to the combination of rotorcraft and fixed-wing controls[28,29]. To provide insight into controllability, transition performance, and passenger comfort for a variety of mission profiles, a widely applicable control design methodology is employed here. The aim is a high-level of automation, such that high-level control design knowledge is not required to use the tool.

The control vector is taken as

$$U = [\delta_{c1}, \dots, \delta_{cn}, F_{Tx}, F_{Ty}, F_{Tz}, \tau_{Tx}, \tau_{Ty,} \tau_{Tz}]^T \tag{4.3.1}$$

where δ_{cn} is the deflection of the nth control surface, F_{Tx}, F_{Ty}, F_{Tz} is the force due to thrust in [x, y, z], and $\tau_{Tx}, \tau_{Ty,} \tau_{Tz}$ are the torques due to thrust about [x, y, z]. F_T and τ_T are taken instead of ω_m as control in yaw, roll and pitch is dependent on the thrust differential, rather than ω_T. Linearisation with respect to individual rotors cannot capture this coupling. Hence, the function 'torqueForce2U' is used to select $[\omega_1 \dots \omega_m]$ by solving simultaneous equations 4.1.10 and 4.1.11 in post. Saturation limits are applied. The demanded thrust is restricted to $0.6mg < F_T < 1.4mg$ by default. Note that F_{Ty} is included for completeness but is optionally removed as a control input in most cases, where rotors only rotate forward.

4.3.2. Trim Optimisation

An optimiser is created to generate trim points throughout the mission. These equilibrium points provide the basis for linearisation of the system and control (LQR) tuning, as well as providing point performance parameters for a given mission. The lateral velocity, and all rotational velocities are set to zero.

Each point on the mission profile can be defined by the airspeed (V_a), course angle (χ), and flight path angle (γ), for which the optimal trim control inputs can be calculated.

The objective function is defined as

$$J = J_{trim} + J_{traj} \tag{4.3.2}$$

$$J_{traj} = (\gamma - \gamma_d)^2 + (\chi - \chi_d)^2 + (V_a - V_{a_d})^2 \tag{4.3.3}$$

$$J_{trim} = \dot{u}^2 + \dot{v}^2 + \dot{w}^2 + \dot{p}^2 + \dot{q}^2 + \dot{r}^2 \tag{4.3.4}$$

where J_{traj} represents error in following the desired trajectory, and J_{trim} is the condition for trim[ref]. γ_d, χ_d and V_{a_d} define the demanded trajectory. V_a, χ and γ are calculated as[ref]

$$V_a = \sqrt{u^2 + v^2 + w^2} \tag{4.3.5}$$

$$\chi = \beta + \psi \tag{4.3.6}$$

$$\gamma = \alpha - \theta \tag{4.3.7}$$

χ_d and γ_d are approximated at analysis points along a mission as $\chi_d = \tan^{-1}\left({}^{\Delta z}/_{\Delta x}\right)$ and $\chi_d = \tan^{-1}\left({}^{\Delta y}/_{\Delta x}\right)$. The cost function is squared to greater penalise larger error, provide effective computation of gradients, and improve stability and efficiency[55].

To assess the error landscape, a sweep of key control variables was performed about the states where a simple gradient-based algorithm failed to converge. Two examples are shown in **Figure 18**.

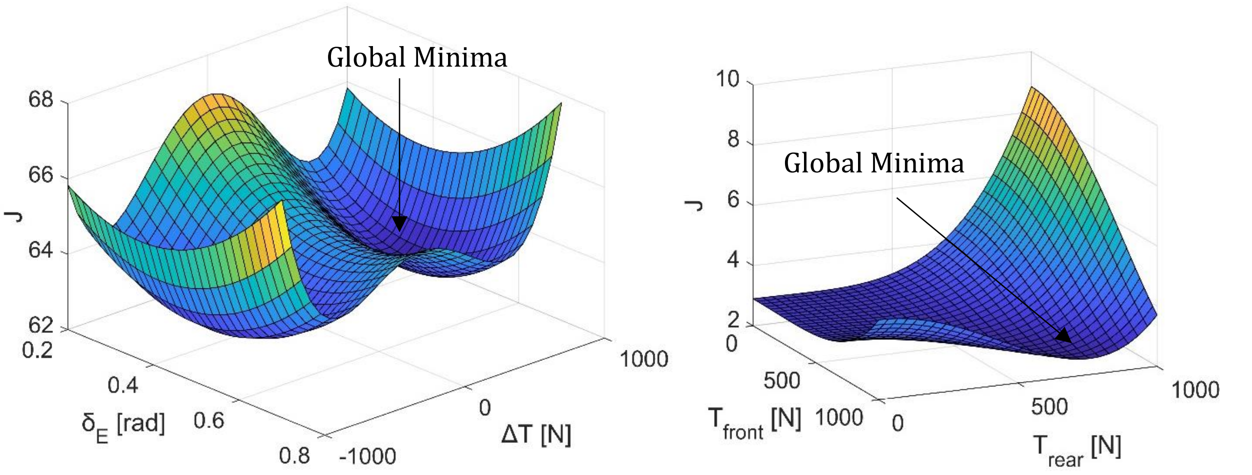


Figure 18: *Parameter space for trim optimisation. Multiple local minima and plateaus can be seen.*

There are multiple local minima closely located due to the trade-off between rotor differential and control surface deflection for moment control. Large plateaus are present in the control optimisation for several states. Note that trim is a multivariable problem, and **Figure 18** represents only the complexity attributed to one set of states. This presents a difficulty in optimisation, namely that a local, gradient-based solver is not sufficient. 3 approaches were considered: using purely global optimisation, using theory to make an initial guess, or using a gradient-free global method to find an initial guess, then using a gradient-based method to converge. Due to the complexity of the aerodynamics, a sufficient guess for all states could not be made from theory. Global optimisation algorithms are generally slower to converge. Hence, the particle swarm method (PSO)[56] was used to provide the initial guess.

PSO is a meta-heuristic method[57,58]; there a stochastic element, and convergence on a global minimum is not guaranteed[57,58]. This has been compensated by allowing the 'guess' function to make multiple attempts with PSO, as well as manual tuning. The methods employed in [[59]] were applied to tune the solver. A large upper inertia range [0 2.3] was required to improve the global search ability[60], much higher than the typical range of [0 1.4][61]. Further research may be refining the method and investigating more effective algorithms, beyond that available in the MATLAB toolbox.

The trim conditions were confirmed by a simple model in Simulink, where there was negligible deviation in any states over 10 s. Note that the aircraft is inherently unstable in VTOL and early stages of transition (Section 4.3.3.). To prevent arbitrary control surface deflection at low speeds - where they have little influence – an additional check is included. If a 10 % change in control surface deflection does not cause a change in J of $> 10^{-12}$, it is set to zero. Similarly, a small additional term is included in J to favour control surface deflections for ω control. **Figure 19** illustrates the process of solving a single trim point.

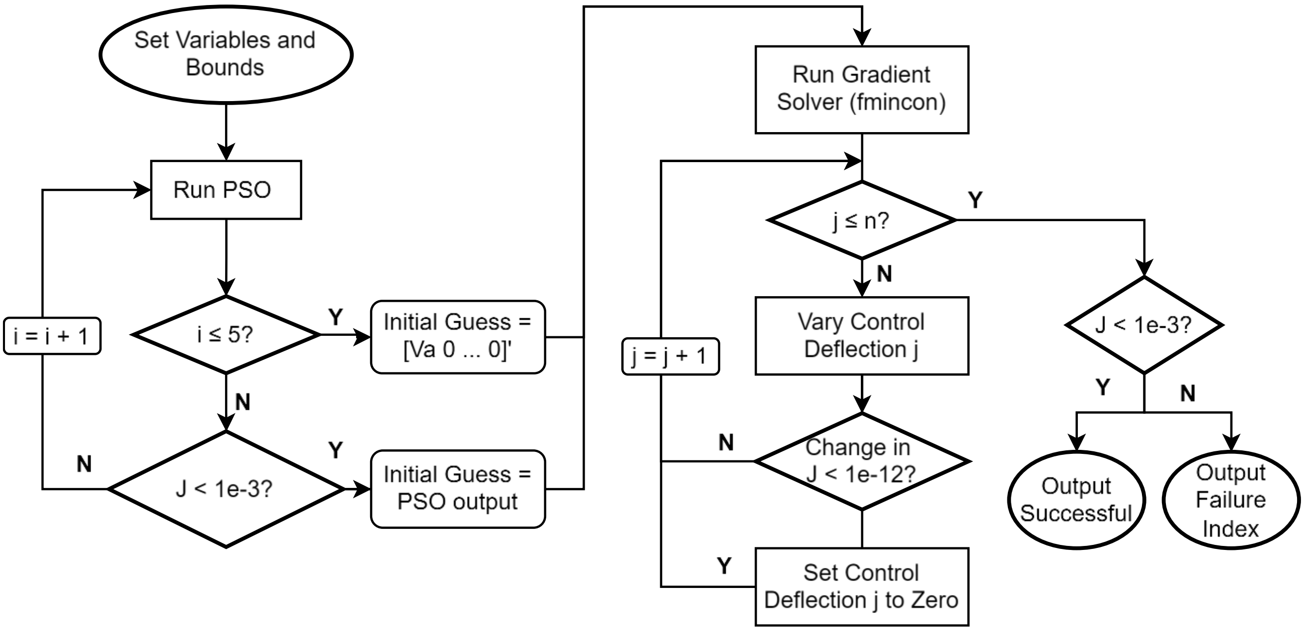


Figure 19: *Hybrid trim solver iteration process.*

Extensive programs and methods have already created to estimate power requirements from thrust requirements and trim conditions. The trim analysis performed here may be used to inform detailed rotor analysis iteration in outside software, but a simple ADT method is employed here as an early estimate. Another alternative path may be to directly map these directly to supplier data using the advance ratio, as seen in [29], and [22]. The power coefficient required for a single rotor is calculated as[13,62,63]

$$C_p = C_{p,i} + C_{p,0} + C_{p_{net}} \tag{4.3.8}$$

where $C_{p,i}$, $C_{p,0}$ and $C_{p_{net}}$ are the induced power coefficient, induced profile power coefficient, and net power coefficient. $C_{p_{net}}$ represents the contribution of the unsteady motions of the aircraft[13]. Parasitic power due to drag and the power required for manoeuvring is directly accounted for in the trim analysis, and therefore represented within $C_{p_{net}}$. $C_{p,i}$ is given by[62,63]

$$C_{p,i} = \kappa C_T \lambda_i \tag{4.3.9}$$

where κ is the correction factor, set as 1.15[13,62], C_T is the thrust coefficient, and λ_i is the inflow coefficient. λ_i varies with the phase of flight. The induced velocity ratio during axial climb/descent is dependent on the velocity, such that[27,62,63]

$$\lambda_h = \frac{V_h}{\Omega R} = \sqrt{\frac{C_T}{2}} \tag{4.3.10}$$

$$\begin{cases} \dfrac{\lambda_i}{\lambda_h} = -\dfrac{V_c}{2V_h} + \sqrt{\left(\dfrac{V_c}{2V_h}\right)^2 + 1} & \dfrac{V_c}{V_h} \geq 0 \\ \dfrac{\lambda_i}{\lambda_h} = \kappa + k_1\left(\dfrac{V_c}{V_h}\right)^2 + k_2\left(\dfrac{V_c}{V_h}\right)^2 + k_3\left(\dfrac{V_c}{V_h}\right)^3 + k_4\left(\dfrac{V_c}{V_h}\right)^4 & 0 \leq \dfrac{V_c}{V_h} \leq 0 \\ \dfrac{\lambda_i}{\lambda_h} = -\dfrac{V_c}{2V_h} - \sqrt{\left(\dfrac{V_c}{2V_h}\right)^2 - 1} & \dfrac{V_c}{V_h} \geq 0 \end{cases} \tag{4.3.11}$$

where V_c is the climb/descent velocity, V_h is the induced velocity at hover, and λ_h is the hover inflow ratio. Ω is the rotational velocity of the rotor, and R is the radius. k_1, k_1, k_3, and k_4 are empirical constants of -1.125, -1.372, -1.718, and -0.655, respectively. In forward flight, λ_i is given by[62,63]

$$\frac{\lambda_i}{\lambda_h} = \frac{1}{\sqrt{(\overline{\mu_x})^2 + \left(\bar{\mu}_z + \bar{\lambda}_I\right)^2}} \tag{4.3.12}$$

where $\bar{\mu}_z$ and $\bar{\mu}_x$ are the advance ratio in x and y nondimensionalised with respect to λ_h. $\bar{\lambda}_I$ is the nondimensionalised inflow coefficient. Equation 4.3.12 is solved using MATLAB's Symbolic Math Toolbox. The profile power is given by[13,62,63]

$$C_{p,0} = \frac{\sigma C_{d0}}{8}\left(1 + K{\mu_x}^2\right) \tag{4.3.13}$$

where C_{d0}, μ_x and σ are the blade aerofoil drag coefficient, advance ratio, and solidity, respectively. σ is given by[27] $\sigma = {}^{NcR}/_{\pi R^2}$ where N, c and R are the number of blades, chord length, and blade radius. K

is a semi-empirical constant. A typical range for K is between[62] 4.5 to 5; here it is taken as 4.6, as estimated in [13]. $C_{p_{net}}$ is given by[13]

$$C_{net} = C_T \frac{V_a}{\Omega R} \tag{4.3.14}$$

Figures **22** to **28** show the analysis results for UAM 1 for the mission shown in **Figure 20** (Mission 1), representing a simple take-off, cruise, and landing procedure. The aircraft is assumed to linearly decelerate from 10 m/s to 0 m/s (purely vertical) in take-off, accelerate to a forward velocity of 20 m/s in transition, and vice versa for landing. 20 m/s is maintained at cruise. The transition trajectory is parabolic, as in [13]. The aim is to cover the range of possible velocity points; however, a slow descent velocity may be unrealistic due to the effects of the vortex ring state (VRS).

The deflection of all 4 rotors is fixed within a tight bound of 0.99 to 1.01 times the rotor tilt shown in **Figure 21**, where this is linearly interpolated to enforce θ_r = 90° when approaching cruise. This is currently a placeholder. However, depending on the design intent, these bounds and the cost function may be redefined, as is seen in Section 6, such as optimising θ_r to minimise power requirements. As with all other modules, the cost functions are anonymous, and hence user defined functions may be input.

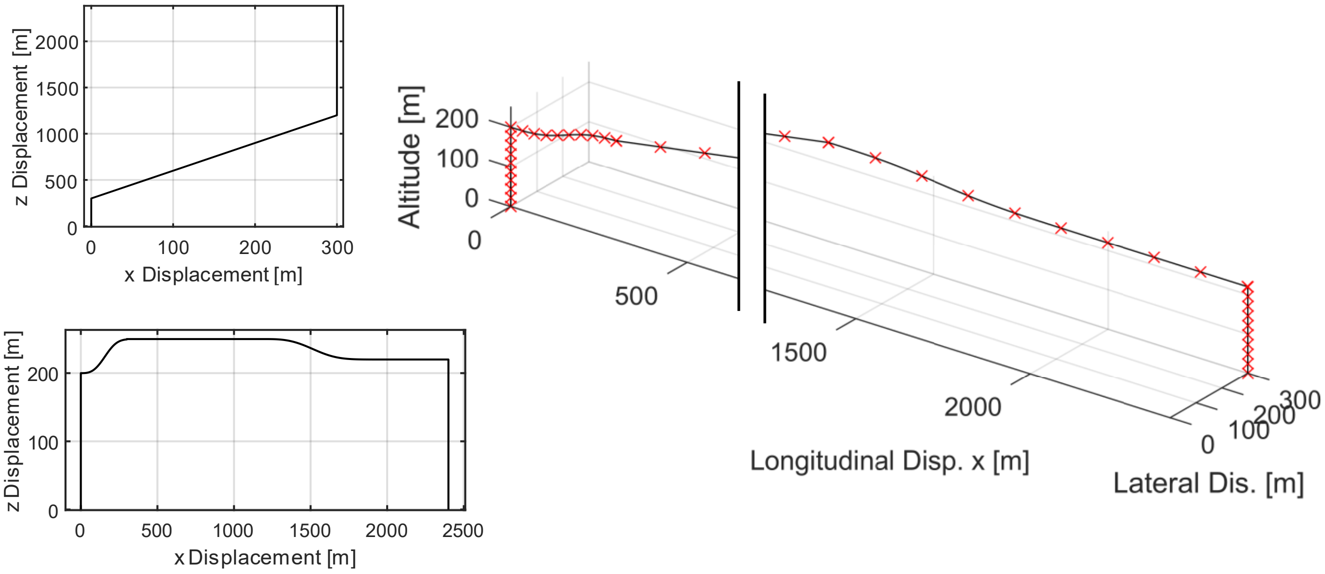


Figure 20: Mission 1 profile.

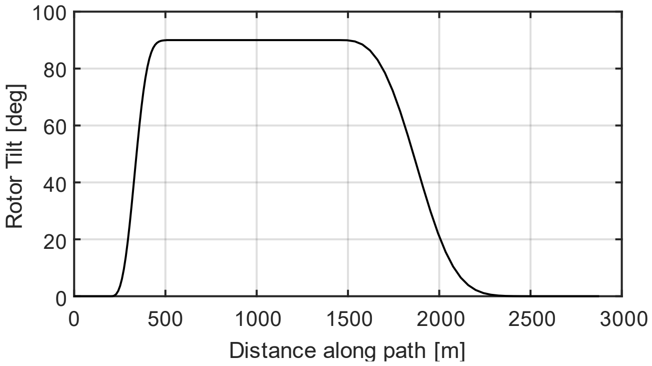


Figure 21: *Mission 1 rotor deflection through the mission profile.*

Figure 22 shows the balance between aerodynamic forces and rotor thrust in each axis.

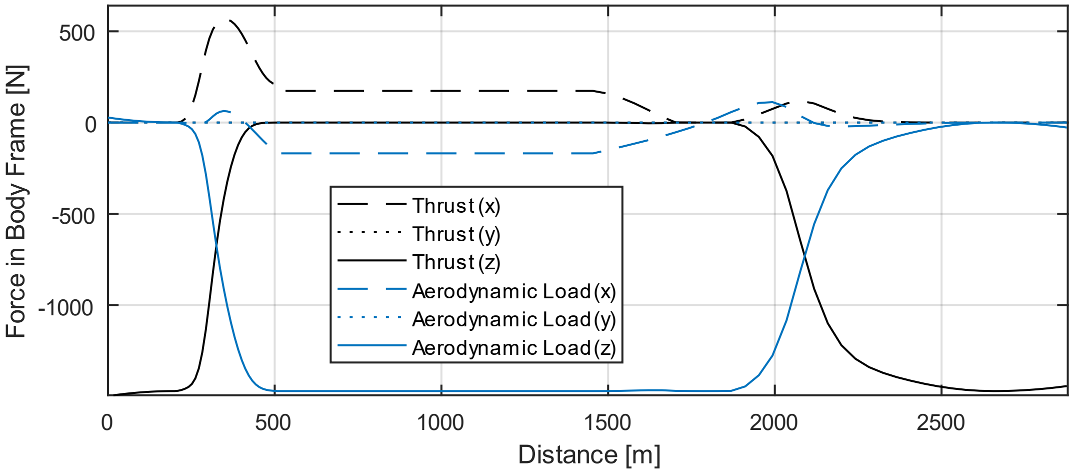


Figure 22: *Required forces in the body frame to trim UAM 1 for Mission 1.*

It is clear that there is a trade-off between lift and vertical thrust in z as the rotors tilt forward. There is an imbalance during the transition climb/descent due to the shifting components of F_g. There is a positive aerodynamic force in the body frame x coordinate during climb. This may initially seem counterintuitive; however, the body frame is taken relative to the chord line, producing a positive component of lift along the reference axis. **Figure 23** shows the aerodynamic coefficients in the wind frame for reference. The lateral (y) loads are negligible as no sideslip is present. Note that this does not account for the requirements for accelerating between trim points but is representative of overall power requirements.

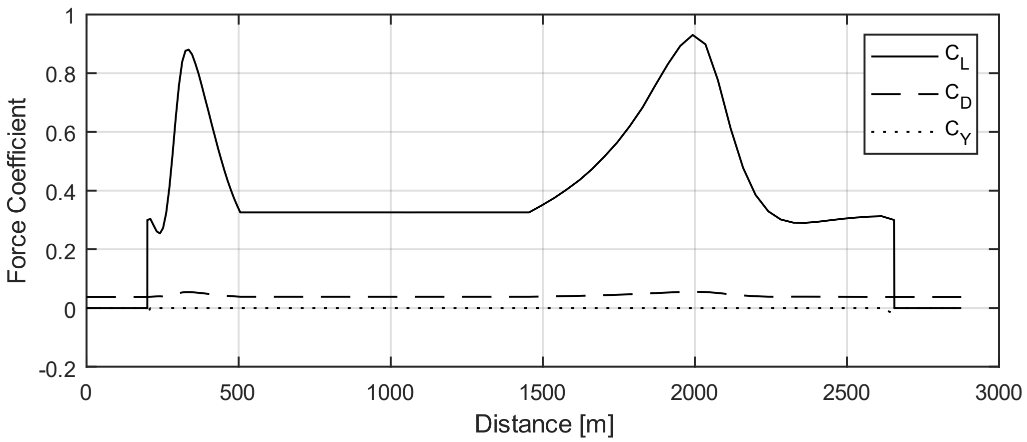


***Figure 23:** Aerodynamic force coefficients required to trim UAM 1 for Mission 1.*

Figures **24** and **25** show the control inputs required at each stage of the mission. **Figure 24** shows the variation of α. The VTOL stages are cut off as $\alpha \rightarrow 90°$ (effective). There is a differential between the front and rear rotors to balance the CG, which gradually decreases as the rotors tilt forward and the aerodynamic forces dominate. The thrust reduces during early descent as a positive component of gravitational force arises, balancing drag. The angle of attack increases during transition as expected to produce additional lift, with the elevator deflection compensating for trim. The deflection is within bounds (±25°). The deflection is cut off once the airspeed is low enough that the elevators' impact on trim is negligible.

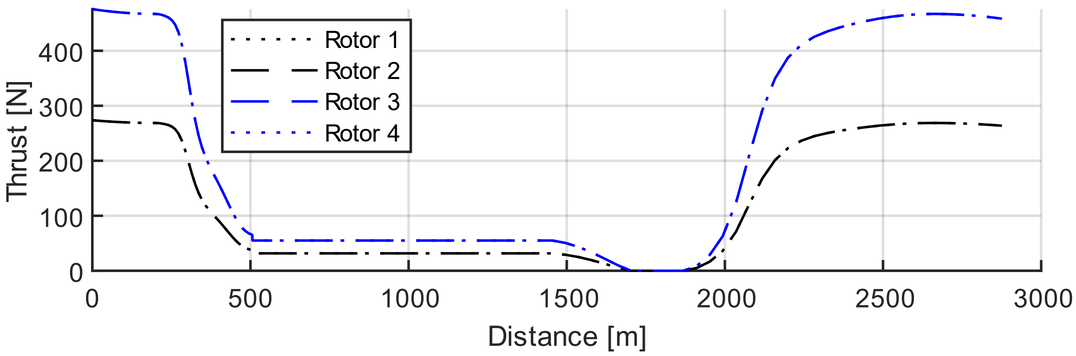


***Figure 24:** Rotor thrust required to trim UAM 1 for Mission 1.*

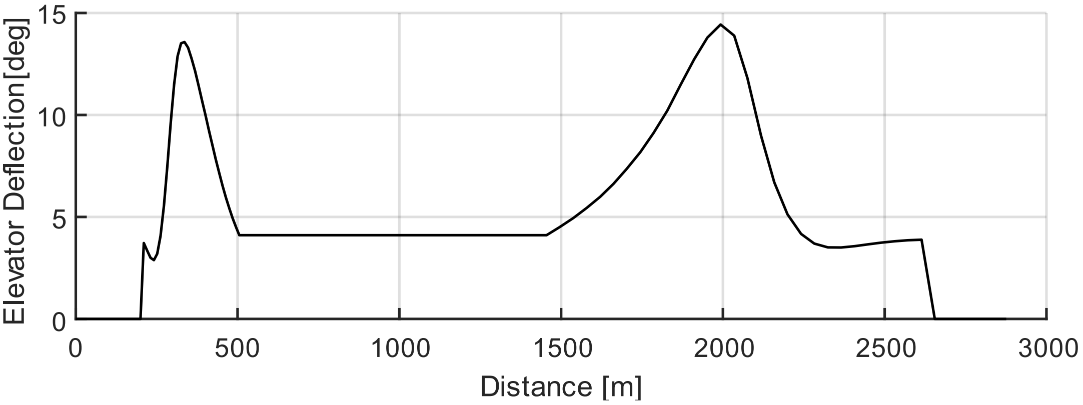


Figure 25: *Elevator deflection required to trim UAM 1 for Mission 1.*

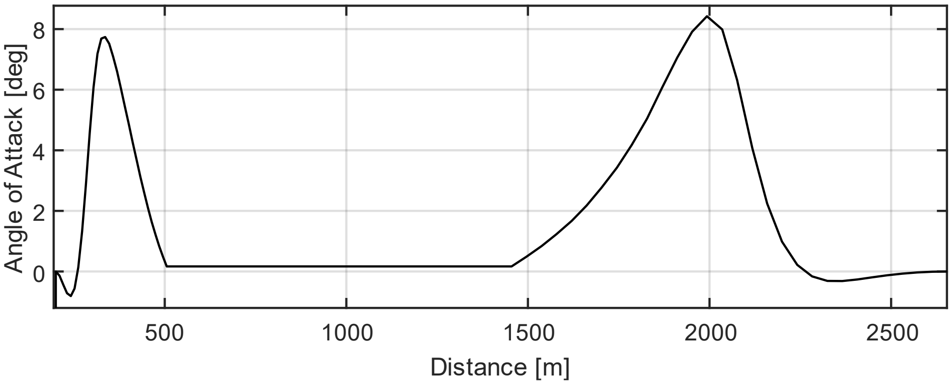


Figure 26: *Angle of attack required to trim UAM 1 for Mission 1.*

Figure 27 shows the velocity in the Earth frame, demonstrating that the solution accurately represents the mission profile. u_e is decreased at cruise as the aircraft is at an angle to the fixed earth reference axis, seen by the corresponding v_e component. w_e corresponds with accelerated VTOL, and transition climb.

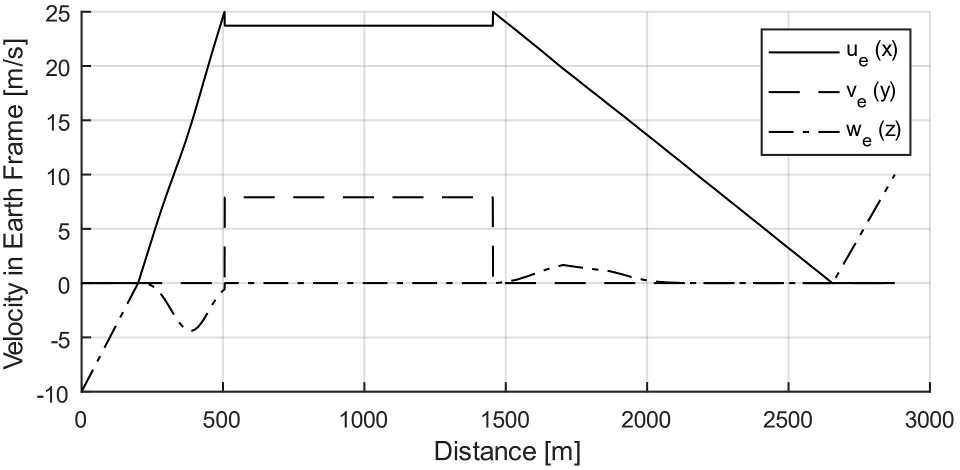


Figure 27: *Velocity in the Earth Frame axis corresponding to Mission 1.*

Figure 28 shows the power requirements through transition and cruise.

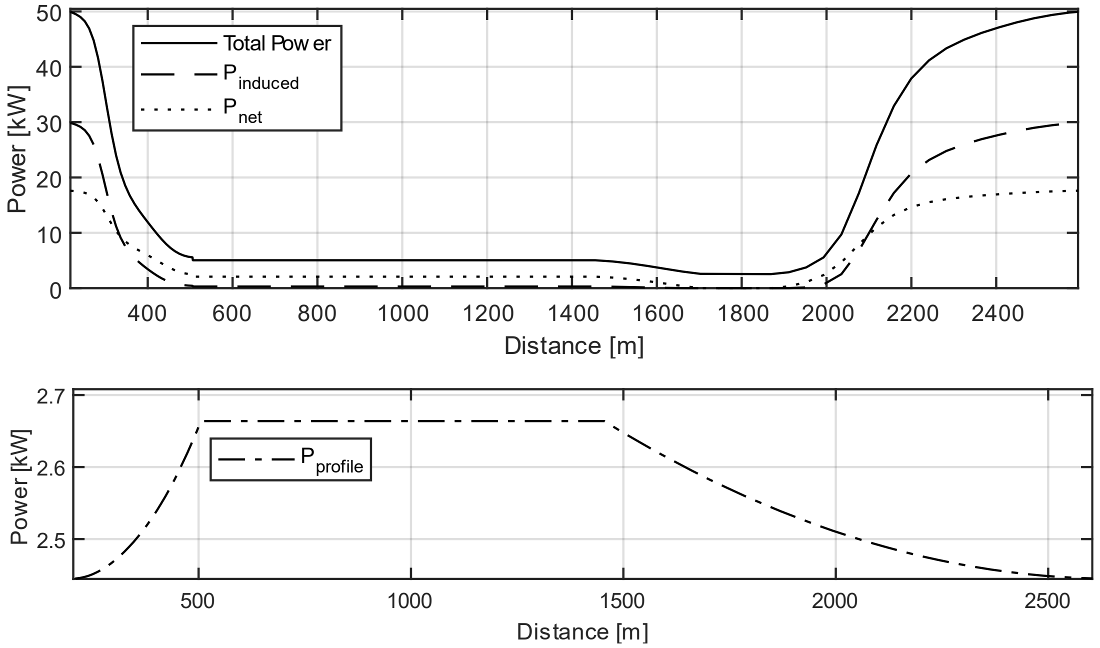


Figure 28: *Transition power requirements to trim of UAM 1 for Mission 1.*

P_{net} dominates during cruise to overcome drag. $P_{induced}$ is reduced due to the increased mass flow rate through the rotor[27]. The profile power directly corresponds to the square of forward velocity, as expected. The behaviour qualitatively aligns with the analysis in [13], albeit with no initial peak as climb acceleration is not included here. The total power required for cruise is consistent with preliminary work in [40].

4.3.2. Reference Scheduling Controller Design

This section builds upon the work in [29], instead applying the reference scheduling concept to a tiltrotor. The control is extended to a whole mission profile, rather than velocity through transition only. The controller employed is reference scheduling controller, a type of gain scheduling that moves between trim states. The individual gains are tuned using Linear Quadratic Regulator (LQR) control, requiring the system to be linearised about the trim points.

The advantage of this method is that LQR can be applied to MIMO systems with little manual tuning, has guaranteed stability[64] (in the absence of modelling error). LQR is also a form of optimal control that balances controller effort and state error. This is useful in power critical cases, where the role of the rotors in control may be minimised.

Figure 29 illustrates this concept[29]. To accelerate between trim points, the controller moves between indices containing the trim points and associated gain. The lateral and longitudinal states are decoupled in **Figure 29**, but not this work. In [29], stability checks are performed between each index, however, this was not found to be necessary in this work. The instability likely arises from the additional complexity of a tilt-wing design.

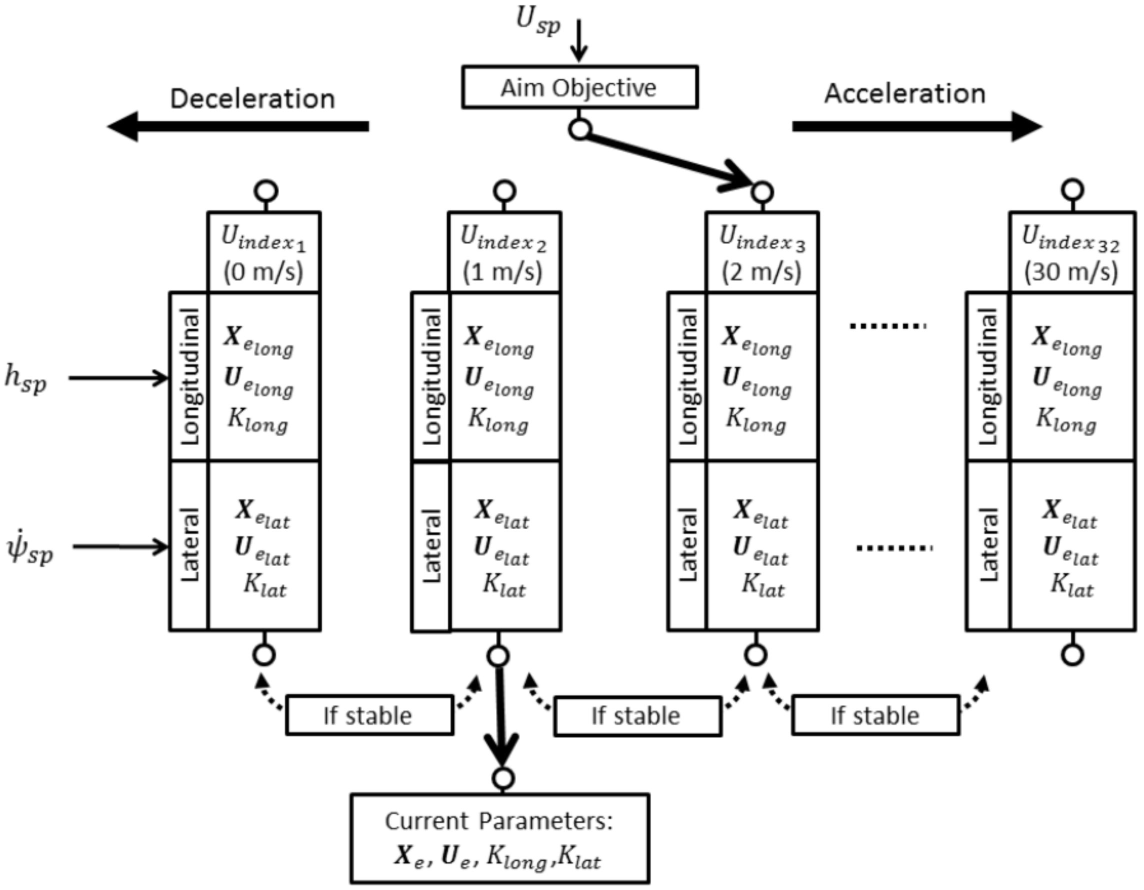

U_{sp}
Aim Objective
Deceleration
Acceleration
U_{index_1} (0 m/s)
U_{index_2} (1 m/s)
U_{index_3} (2 m/s)
$U_{index_{32}}$ (30 m/s)
Longitudinal
$\boldsymbol{X}_{e_{long}}$ $\boldsymbol{U}_{e_{long}}$ K_{long}
Lateral
$\boldsymbol{X}_{e_{lat}}$ $\boldsymbol{U}_{e_{lat}}$ K_{lat}
h_{sp}
$\dot{\psi}_{sp}$
If stable
If stable
If stable
Current Parameters:
$\boldsymbol{X}_e, \boldsymbol{U}_e, K_{long}, K_{lat}$

Figure 29: *Reference scheduler parameter selection. Source: [29].*

For a predefined mission profile, it is assumed that the aircraft will stay within an acceptable corridor. However, this same concept may be applied to a real-time controller by using state-dependent feedback, where the model is linearised at regular timesteps and gain is recalculated based on the current state, as in [65]. Vector field correction is used here to maintain trajectory, which may also be applied in a real-time controller. Critical to real-time application is that not every state will be measurable, and a Kalman filter or observer will be required to compensate.

In this work, the reference scheduling concept will be extended to cover the full mission profile, including all 3 key modes. While, highly non-linear transition manoeuvres require a large number of sample points for stability, as demonstrated by [29], this is generally less necessary for cruise and VTOL. This is a requirement that should be assessed qualitatively and set by the user. For the purpose of visual clarity, in this work several (30) points are sampled for each stage.

4.3.3. Numerical Linearization

The controller gains were calculated using Linear Quadratic Regulator (LQR) control for linearised dynamics about the trim points. The goal is to minimise the deviation from the desired trim state ($\breve{x}$) by applying a change in control input ($\widetilde{U}$), such that

$$\tilde{x} = \breve{x} - x \tag{4.3.15}$$

$$\widetilde{U} = \breve{U} - U \tag{4.3.16}$$

The aircraft dynamics can be characterised by a set of n scalar non-linear functions such that[29,65,66]

$$\dot{x} = f(U, x) = 0 \tag{4.3.17}$$

at each trim point. To linearise about the trim points in state-space form, the system must first be written in implicit linear state-variable form as[29,65,66]

$$\begin{cases} \dot{\tilde{x}} = A\tilde{x} + B\widetilde{U} \\ y = C\tilde{x} + D\widetilde{U} \end{cases} \tag{4.3.18}$$

where A and B are Jacobian matrices calculated as[29,65,66]

$$\nabla_x f_i = \begin{bmatrix} \frac{\partial f_i}{\partial x_1} & \frac{\partial f_i}{\partial x_2} & \cdots & \frac{\partial f_i}{\partial x_n} \end{bmatrix} \tag{4.3.19}$$

$$A = \begin{bmatrix} \nabla_x f_1 \\ \vdots \\ \nabla_x f_n \end{bmatrix} \qquad B = \begin{bmatrix} \nabla_U f_1 \\ \vdots \\ \nabla_U f_n \end{bmatrix} \tag{4.3.20}$$

$C = I$, and $D = 0$. $\partial f_i / \partial x_j$ can be approximated by using Taylor series expansions about the equilibrium point to give equations 4.3.11 and 4.3.12, and neglecting terms of order δx^2 and higher, or δx^4, respectively[66].

$$\left.\frac{\partial f_i}{\partial x_j}\right|_{x_j = \breve{x}_j} = \frac{\dot{x}_1 - \dot{x}_{-1}}{2(\delta x)} + O(\delta x^2) \tag{4.3.21}$$

$$\left.\frac{\partial f_i}{\partial x_j}\right|_{x_j = \breve{x}_j} = \frac{8(\dot{x}_1 - \dot{x}_{-1}) - (\dot{x}_2 - \dot{x}_{-2})}{12(\delta x)} + O(\delta x^4) \tag{4.3.22}$$

where

$$\begin{aligned} \dot{x}_1 &= f(\breve{x}_j + \delta x) \\ \dot{x}_{-1} &= f(\breve{x}_j - \delta x) \\ \dot{x}_2 &= f(\breve{x}_j + 2\delta x) \\ \dot{x}_{-2} &= f(\breve{x}_j - 2\delta x) \end{aligned} \tag{4.3.23}$$

When selecting δx, it is key to ensure convergence. Too large of a perturbation will cause truncation error, whilst too small of a step will cause a round-off error due to MATLAB's finite precision.[66] This is verified iteratively within the function by calculating the fractional error, and comparing it with a convergence tolerance, as in [[66]].

To verify accurate linearisation, the step response of the linear system was compared to that of the non-linear model in Simulink at the trim condition. The results of deviation in δ_E (from equilibrium) of amplitude 0.0005, 0.01 and 0.05 rad are shown in **Figure 30**.

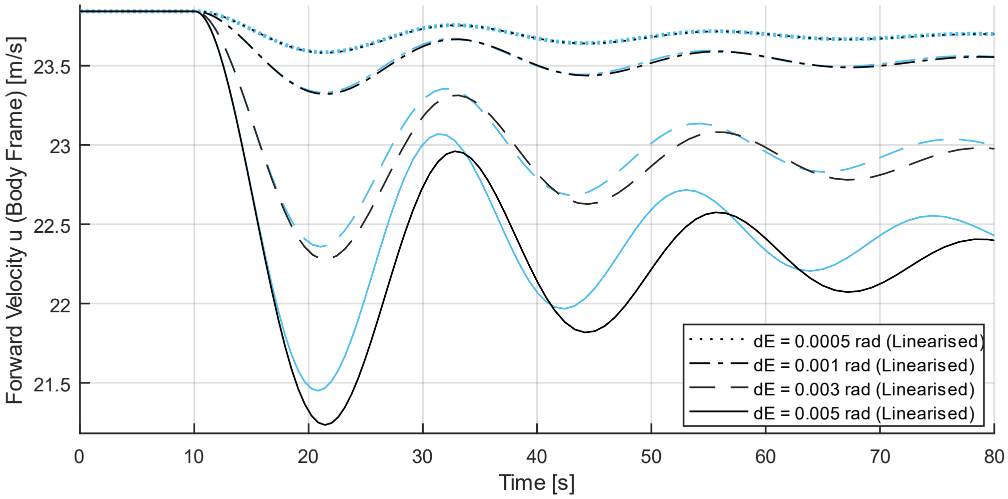


Figure 30: *Comparison of linearised and non-linear aircraft dynamic systems for a step input in elevator deflection (dE). The corresponding non-linear responses are shown in blue/grey. The step is applied at 10s.*

The linearised system is accurate for $\Delta\delta_E = 0.0005$, but does not follow well for larger deviations. This is as expected as the numerical method used is based on small perturbation theory, and the aircraft analysed is highly manoeuvrable[40] with the elevator having high control authority ($C_{L_{\delta E}}$> 0.2, $C_{m_{\delta E}}$ > 0.2). u changes by > 10% for a deflection of 0.005. This disparity was also present in the steady state at 200 s but is not shown in **Figure 30** for visual clarity.

Simulink offers a linearisation tool, however this can be slow for large numbers of I/O and lacks flexibility; a new model with resized matrices is required for differently sized I/O, and therefore cannot be easily automated within the program. However, the tool provided additional verification in development stages. An additional check included comparison with hand calculations of the state-space during VTOL and cruise.

The small perturbation assumption is acceptable as the mission profile is predefined, with the reference scheduler making small steps between equilibrium points. The impact of linearisation error will therefore be dependent on step size. The controller will actively compensate to correct the error, provided it is stable under the non-linear condition.

4.3.3. Stability Analysis

With all 3 mission stages linearised, the stability through the mission profile may be analysed. The eigenvalues through the transition mode of UAM 1 are shown in **Figure 31**.

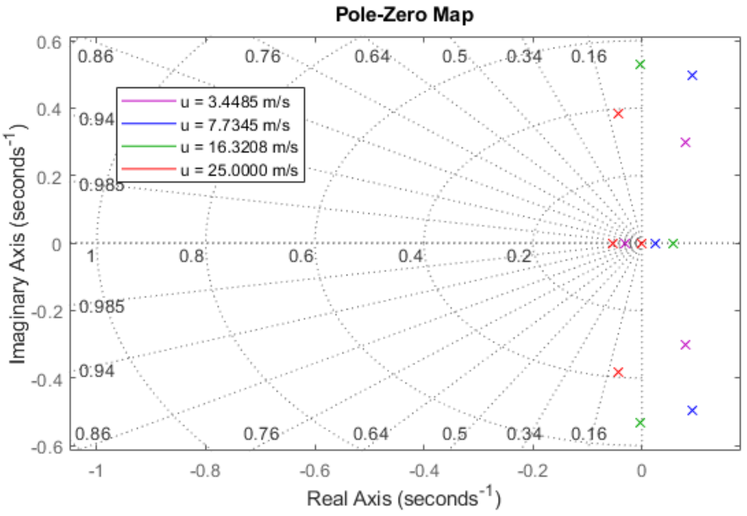


Figure 31: *Stability analysis of linearised transition mode.*

The cruise model is inherently stable, confirmed by analysis of the aerodynamic and stability derivatives. Quadcopters, on the other hand, are inherently unstable without control. By analysing the pole zero map of the transition mode we can see that these unstable modes are still present up until right before cruise.

4.3.4. Linear Quadratic Regulator (LQR) Design

LQR is applied to determine the gain, K, to minimize the cost function[65,67]

$$J = \int_0^{\infty} (\tilde{x}^T Q \tilde{x} + \tilde{u}^T R \tilde{u})\, dt \tag{4.3.24}$$

where Q and R are weighting matrices, representing the cost of state error and control effort[ref], respectively. R has been defaulted to 1 for all inputs but may be adjusted based on application - e.g. setting a large thrust weighting for a power critical case.

Q and R may be selected manually or optimised using the function '*solveQR.m*' to minimise the error in Equation 4.2.13, as in [67]. Instead of a genetic algorithm (as in [67]), PSO is again used for improved speed[60], using a modified version of the model *('Model_TuningQR')* with fast restart such that it compiles only once, and higher step tolerance. H is a simple weighting function of [0.5 0.5 0.5 4 4 4 4 4 4] to prevent u from dominating the error, better enforcing $p, q, r = 0$.

$$J = \int_0^{\infty} H\tilde{x}^2 \ dt \tag{4.3.25}$$

Equation 4.3.13 is a generalised example to minimise overall error; It may be modified to target application-specific aims, such as a term to directly prioritise speed of response or eliminate oscillation.

This is easily done as the function is passed into the functions anonymously. J is evaluated at multiple sample points in each mode.

The results for each mode of UAM1 are given in Table X. The transition stage is split into two modes to account for how the authority of the control surfaces increases with speed. Values of 3000 represent the upper bound of the search. The limiting factor of Q is the saturation limits. In this case, Q reflects the importance of the individual states in each mode.

***Table 3** – LQR Weighting Matrices for UAM 1. The Q diagonal represents [u v w p q r ϕ θ ψ].*

Mode	Q	R
VTOL	diag([680, 403, 1000, 1000, 737, 291, 177, 21, 206])	diag([1, 1, … 1])
Transition (Stage 1)	diag([931, 294, 408, 773, 1000, 974, 868, 432, 741])	diag([1, 1, … 1])
Transition (Stage 2)	diag([489, 1, 106, 574, 78, 430, 151, 1000, 811])	diag([1, 1, … 1])
Cruise	diag([715, 567, 375, 492, 34, 778, 377, 924, 700])	diag([1, 1, … 1])

K is determined from the solution (P) to the algebraic Riccati Equation[65,67]

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \tag{4.3.26}$$

$$K = R^{-1}B^T P \tag{4.3.27}$$

for each trim point using an inbuilt MATLAB function. The input is modified as $\widetilde{U} = -K\tilde{x}$, where K_i takes the form

$$K_i = \begin{bmatrix} k_{u\delta c_1} & k_{v\delta c_1} & k_{w\delta c_1} & k_{p\delta c_1} & k_{q\delta c_1} & k_{r\delta c_1} & k_{\varphi\delta c_1} & k_{\theta\delta c_1} & k_{\psi\delta c_1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ k_{u\delta c_n} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ k_{u_{FTx}} & k_{v_{FTx}} & k_{w_{FTx}} & k_{p_{FTx}} & k_{q_{FTx}} & k_{r_{FTx}} & k_{\varphi_{FTx}} & k_{\theta_{FTx}} & k_{\psi_{FTx}} \\ k_{u_{FTy}} & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ k_{u_{FTx}} & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{u_{\tau_x}} & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{u_{\tau_y}} & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{u_{\tau_z}} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \tag{4.3.28}$$

where i is the aircraft trim reference point and n is the number of control surfaces. k_{a_b} is the gain in control term b in response to error in a.

The state tracking performance for the trim points of Mission 1 is shown in **Figure 32.** 1-D lookup tables are used to interpolate between trim points. The control requirements are shown in **Figure 33**. It demonstrates that the true control requirement differs significantly from that required by trim, and are highly dependent on control performance. **Figure 32** shows significant steady state error in in both velocity and angular states. **Figure 33** shows that the overall thrust demand deviates minimally, hence the trim requirements alone provide an adequate assessment of power requirements.

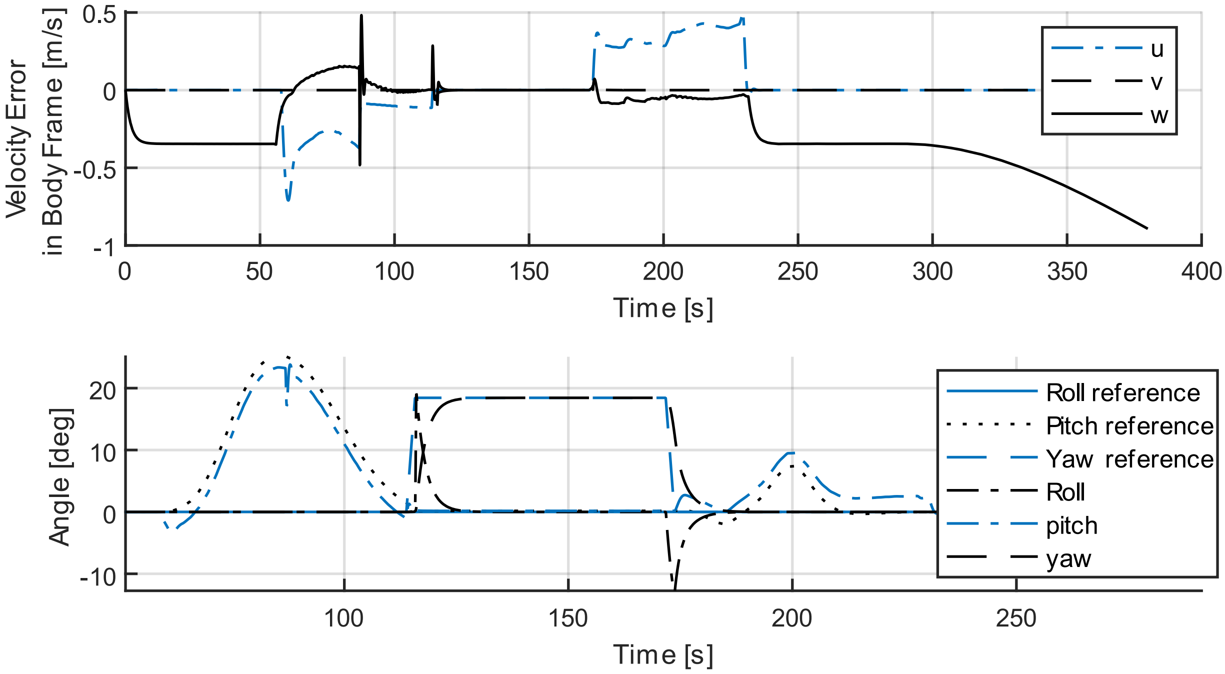


***Figure 32:** LQR controller demand state tracking performance.*

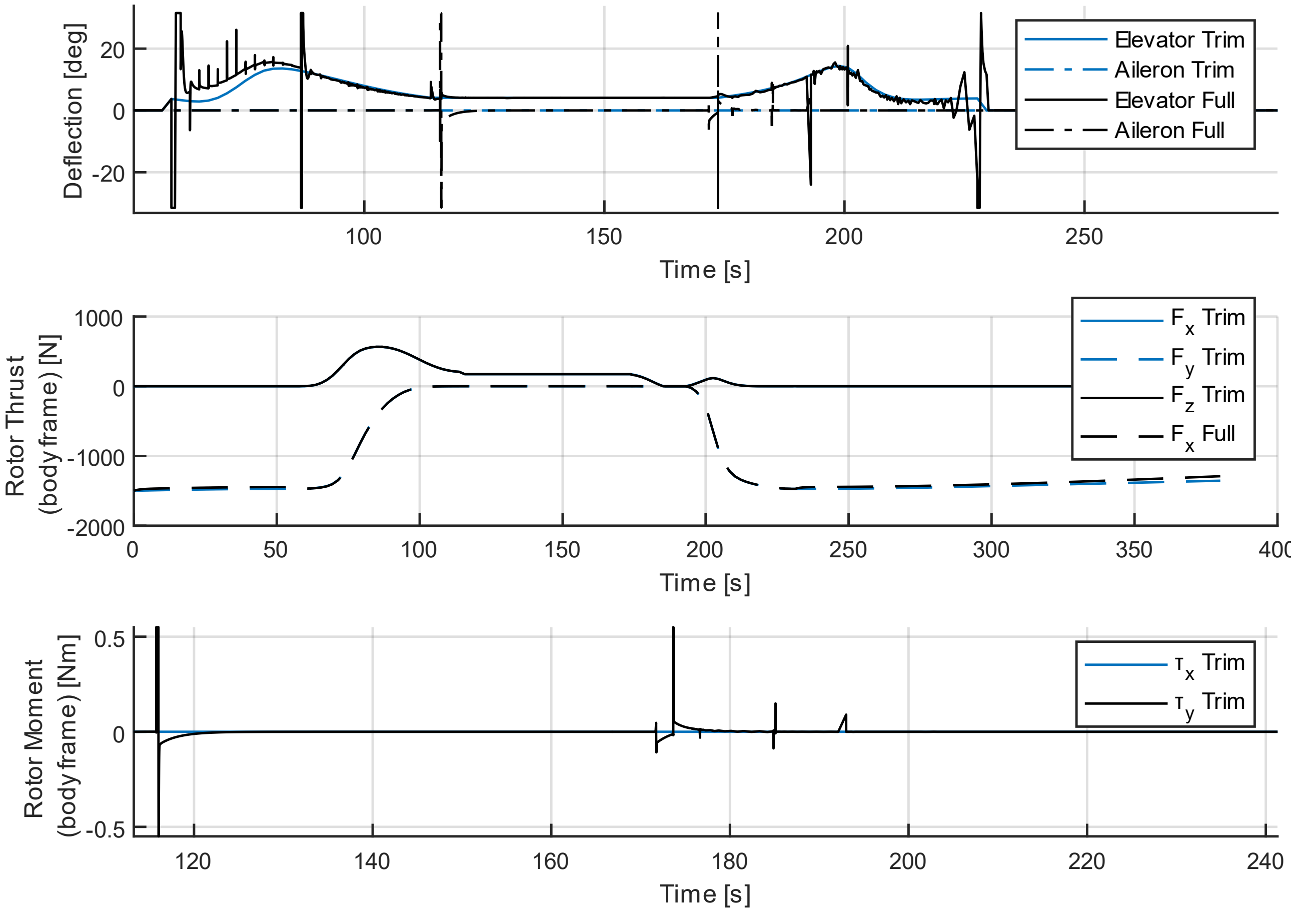


***Figure 33:** LQR controller control demands for Mission 1 compared against trim.*

4.3.5 LQI Control

The steady state error can be eliminated by introducing an integral term, K_I. The controller becomes Linear Quadratic Integral (LQI) control This is achieved by defining an augmented state-space including the term $\tilde{x}_i$ such that[67]

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{x}}_i \end{bmatrix} = \begin{bmatrix} A & O_{k\times 6} \\ -C_{yi} & O_{6\times 6} \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{x}_i \end{bmatrix} + \begin{bmatrix} B \\ O_{6\times k} \end{bmatrix} \widetilde{U} \tag{4.3.29}$$

$$y = \begin{bmatrix} C & O_{6\times 6} \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{x}_i \end{bmatrix} + D\widetilde{U}$$

where $\dot{\tilde{x}_\imath} = [u\ v\ w\ \emptyset\ \theta\ \varphi]^T$, and C_{yi} selects the outputs to be tracked[67]. k is the total number of control inputs ($k = n + 6$). Equations 4.3.14 to 4.3.15 can be applied to the augmented system to obtain the gain $K = [K_x\ K_i]$, where K_i is the integral gain. Back-calculation is used to prevent integrator wind-up[ref]. The back calculation coefficient (k_b) is selected by trial-and-error at this stage, but more sophisticated methods may be applied as in [68] for improved dynamic behaviour.

The system then takes the general form shown in **Figure 34**, adapted for linearisation about trim as in **Figure 35**. Note that the state output from the dynamics model is $x = \breve{x} + \tilde{x}$.

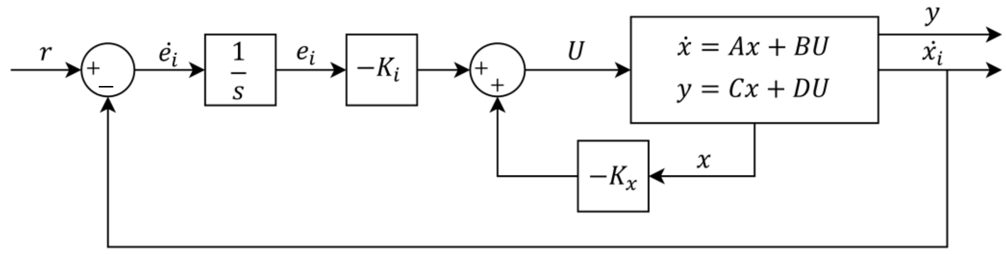


Figure 34: *LQI general form.*

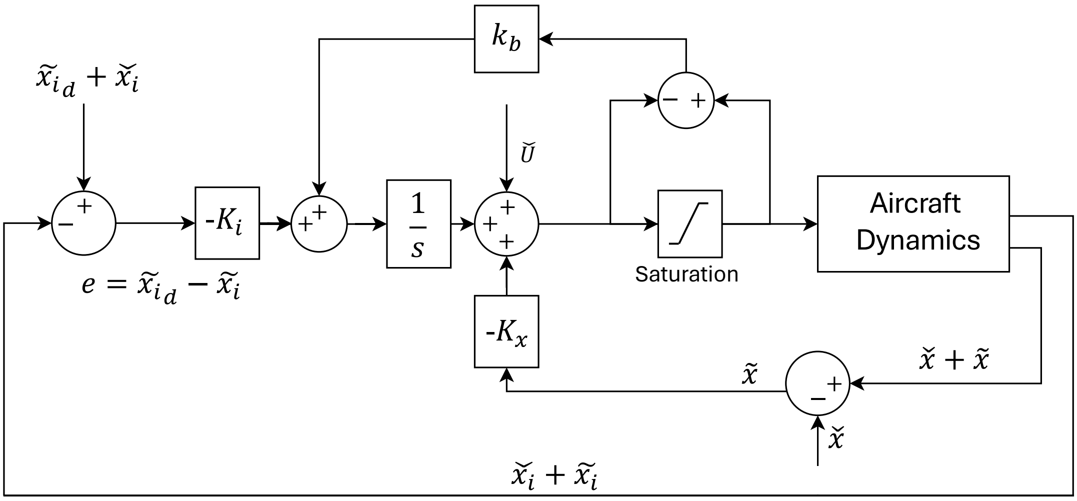


Figure 35: *LQI reference scheduler form with back calculation.*

The LQI Simulink model indexes time points stored in the mission structure to determine the reference parameters. The implementation in MATLAB is shown in **Figure 36**.

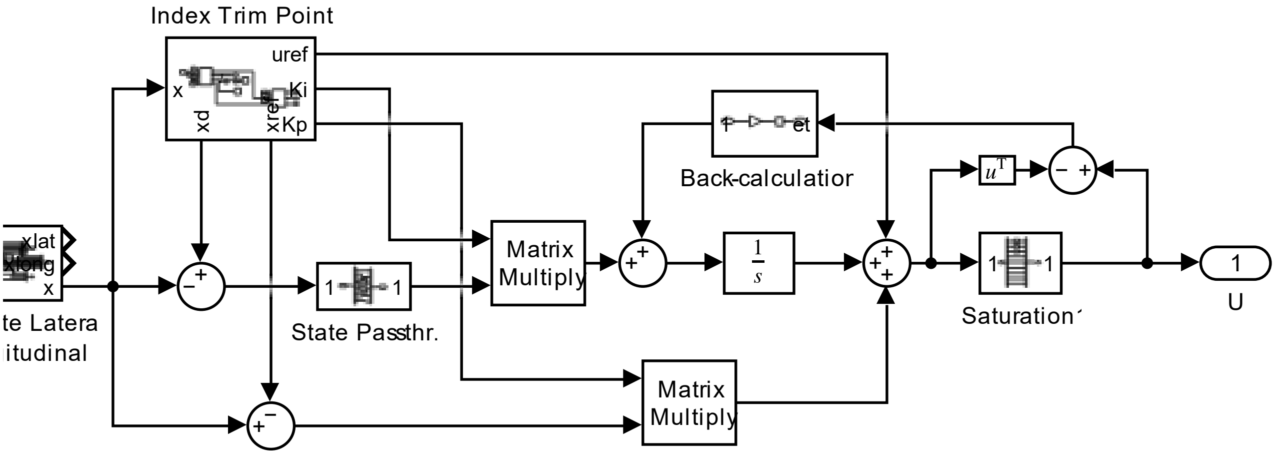


Figure 36: *LQI controller Simulink subsystem implementation.*

Q and R must be re-evaluated for LQI and are given in **Table 4**.

Table 4 *– LQI Weighting Matrices for UAM 1. The Q diagonal represents [u v w p q r ϕ θ ψ $\dot{\vec{x}}_\iota$].*

Mode	Q	R
VTOL	diag([757, 1000, 389, 182, 1, 1, 840, 928, 1000, 569, 1, 1, 1000, 1000, 1])	diag([1, 1, ... 1])
Transition (Stage 1)	diag([962, 1000, 840, 138, 930, 711, 1, 139, 121, 878, 1000, 1, 160, 410 746])	diag([1, 1, ... 1])
Transition (Stage 2)	diag([645, 1, 25, 982, 1, 926, 1, 687, 97, 1000, 1000, 248, 335, 978, 841])	diag([1, 1, ... 1])
Cruise	diag([1, 620, 820, 428, 1, 1000, 1000, 1000, 114, 1, 1000, 191, 775, 1000, 353])	diag([1, 1, ... 1])

Figure 37 compares the step response under LQI and LQR control. Generally integral action results in an improved speed of response but introduces overshoot and oscillatory behaviour. However, the LQI optimisation has converged on the opposite. This is due to these effects being heavily penalised by Equation 4.2.25, resulting in lower control effort and a slower but more stable response Stability is key during transition, however, Equation 4.2.25 may be adjusted should a more rigidly defined result (e.g. overshoot, speed of response, settling time limits) be desired.

The tracking error under LQI control is shown in **Figure 38**. It can be seen that the steady state error is eliminated. There is a significant spike in tracking error at the transition-to-VTOL boundaries. This is due to a significant change in dynamic behaviour at low speed, compared with earlier transition. This error could be minimised by selecting several sample points about that index for optimisation, however, not only would this introduce additional computational cost, but it may allow instability to arise later in transition. The Q and R matrices would be overtrained for a single region.

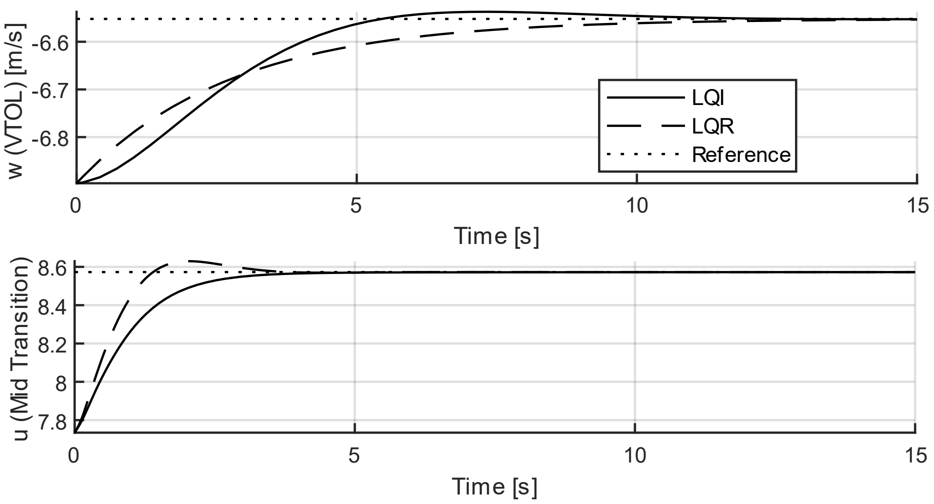


Figure 37: *Step response of LQR vs LQI control.*

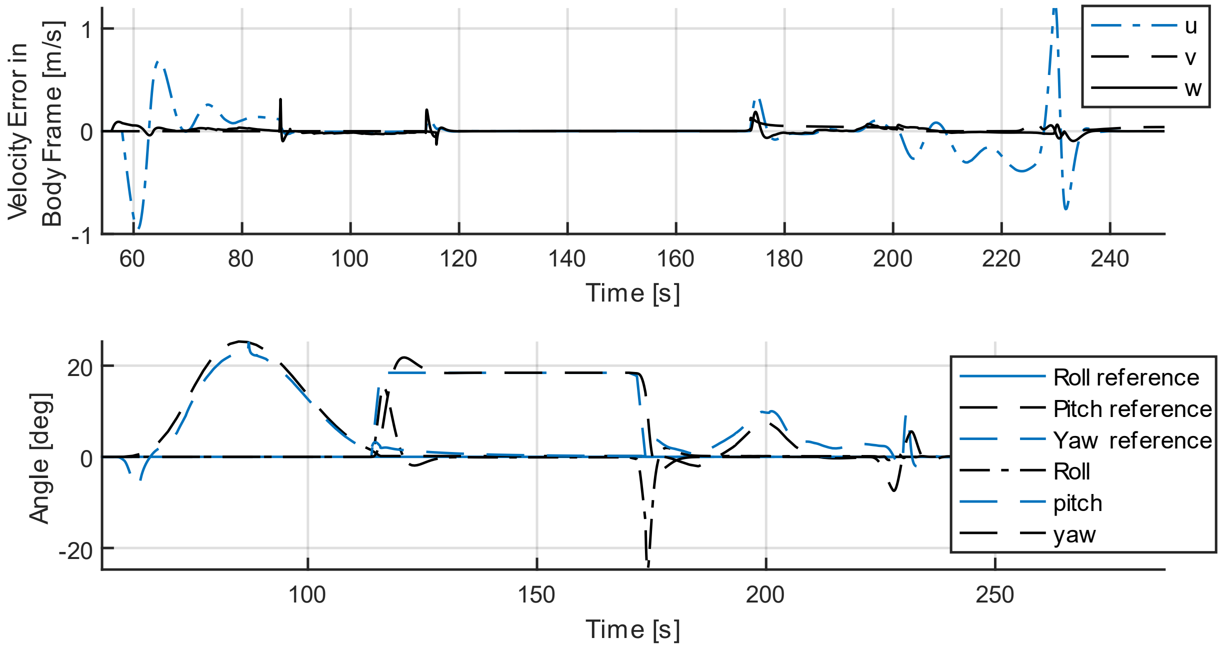


Figure 38: *LQI controller demand state tracking performance.*

Figure 39 compares the position following for Mission 1. The controller tracks the state vector over time, x, rather than the position along the mission itself. **Figure 39** illustrates the need for an additional navigation controller to prescribe a change in state. While the demand states track well, the response time taken to accelerate between trim points creates position error. This may be corrected by introducing a navigation controller to prescribe changes in state, as in [42], [65]. Attempts were made to implement this into the model, but significant stability issues arose in the model. This warrants further investigation.

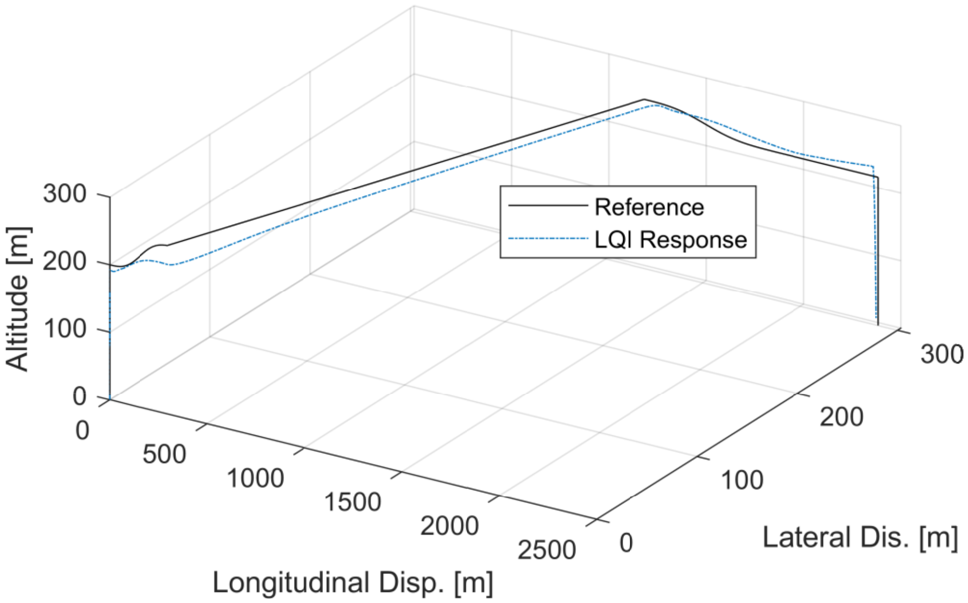


Figure 39: *LQI controller position tracking.*

5. Post Processing Performance Data

Unfortunately, due to a last-minute failure in the GitHub used to store the functions, final results comparing all UAMs could not be obtained. Instead, this section describes how the tools developed and evidenced throughout this report may be applied to a concept design.

5.1. Power Calculations

The power requirements are calculated as in Section 4.3.2. The total power requirements for UAMs 1 and 2 for Mission 1. The cruise velocity for UAM 2 is increased to 60 m/s to meet lift requirements due to the much greater mass (150 kg vs 2880 kg). Additionally, as one set of VTOL rotors has no pitching moment about the CG, the distribution of thrust between the 6 rotors becomes arbitrary, provided the other 4 are balanced. To compensate and produce a realistic result, a cost term should be introduced to minimise the change in control input between steps. To enforce wind-down of the VTOL rotors, a parabolic decreasing upper bound would be introduced.

The overall travel in $[x_E \; y_E \; z_E]$ should be kept the same as Mission 1 for fair comparison, with the velocity and transition profile being modified. UAM 2 is vulnerable to rotor-wing interference; hence, the results with and without are shown here. The expected result would be that UAM experiences higher power requirements due to loss of lift from the interference effects.

5.2. Transition Performance

Beyond power requirements, passenger comfort, and transition speed are key performance indicators . Passenger comfort may be characterised by both rotational acceleration and jerk. These would be shown for UAM 1 under LQI control.

5.3. Safety and Redundancy Analysis

To analyse the redundancy, it should be evaluated whether the aircraft can maintain trim and perform a landing manoeuvre in a rotor out case. With the original purpose of being unmanned, UAM 1 is not controllable in a rotor out case. UAM 2 has high redundancy and can behave as a quadcopter should the tilt rotors fail. The trim optimisation should be ran for UAM 2 to verify whether trim is achievable, and that data may then be taken forward to analyse the resulting jerk and acceleration experienced by passengers.

6. Application to Concept Iteration

This section involves application of the tool to concept iteration of UAM 1. Example MATLAB code for different use cases may be found in Appendix B.

6.1. Rotor Number and Position

3 different rotor configurations should be considered. These would include a hexacopter, a far outbound position (to reduce aerodynamic interference), and a close inbound position. The key advantages of increasing the number of rotors arises from redundancy, and power efficiency. The expected result would be that the far inbound position increases the power and trim requirements through the mission.

Say that UAM 2 is subject to space restrictions, or structural considerations that effect the rotor position. The tool may be used to assess the impact of rotor interference and centre of gravity effects on the trim requirements by varying the moment arm of the individual rotors (whilst maintaining balance).

6.2. Optimising Transition Profile

The trim solver may be adapted to optimise the transition profile, either by fixing the reduction in rotor thrust, or by setting Equation 4.3.2 to include a power term. The plot would then show the ideal path through transition for power. The control system could be designed for 3 different transition profiles, and the performance analysed

6.3. Aircraft Geometry

Concept design is vulnerable to modelling error. The effect of this error with respect to control surfaces may be analysed by varying the control authority (control derivative) by a range of factors. An example is shown below.

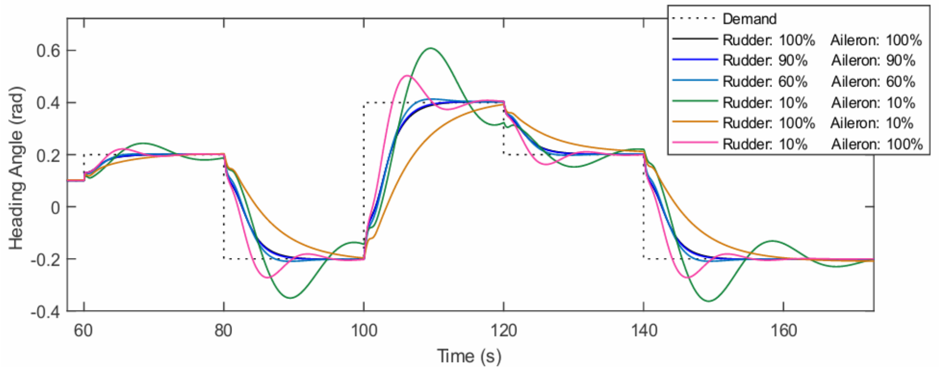


Figure 40: *Sensitivity analysis. Source: [40].*

7. Conclusions

A concept design tool to generate control inputs over a mission has been developed. An object-oriented approach has been taken to develop a unified set of component models for any number of rotors or lifting surfaces. The model is largely independent of empirical or high-level data, with semi-empirical methods being employed to represent key non-linear effects to be considered in eVTOL designs. However, the general model is not exhaustive, and is only representative of tilt duct or tilt rotor configurations. The aerodynamics module should be modified such that a tiltwing configuration may represented, with the limiting factor being obtaining accurate linear aerodynamic coefficients.

These were then implemented in a 6DoF model to simulate a mission profile to get the full states over time based on the given mission and obtain point performance parameters. Particle swarm optimisation has been employed to perform trim analysis through a mission profile, including the over-actuated transition stage. Investigation into the parameter space was performed, and the algorithm was tuned to purpose. However, in several places manual tuning of the cost function has been required to obtain a reasonable result, such as minimising the deviation between trim steps. Further investigation into how to automatically generate an effective cost function, beyond simple trim and trajectory requirements, should be carried out.

Detailed aerodynamic interference and stall behaviour is not included in the model. Selective fidelity may be introduced in individual modules to assess the impact of these effects; however, this may ultimately undermine the overall objective of the tool's position in the proposed concept design workflow. However, the model independent nature of each module allows the flexibility to add these details by simply modifying the fundamental function within the aircraft dynamics block (and adding any required parameters to the structures). The goal of building the framework to supplement and integrate with established tools have been achieved.

The control design approach is independent of the model and may be applied to any configuration that can be represented using the tools described. While the controllers designed tracked velocity well, there was overall significant position error; vector fields should be investigated to correct this. The control system designed is intended to be used for purposes of initial simulation, rather than final control law design. The control system design assumes an instantaneous response from the rotors and control surfaces, which is unrealistic. Additional insight into the control behaviour may be gained by introducing transfer functions to represent the delay in response of the actuation systems.

The model validation has largely relied on results from other theoretical approaches or established software, due to the availability of data. Mission profile data in literature is dominated by flight tests, late-stage design in high fidelity software, or results inherently tied to the performance of a specific type of controller. Effort has been made to make quantitative comparisons where possible, but future research should involve comprehensive testing of the tool against full concept datasets. Application examples have been given, and the intended workflow considered, but further investigation should integrate the tool into a real iterative concept design process to identify weak points, and more detailed user requirements.

The model has been created as a collection of functions that interface with the Simulink model; further development should involve design of full user interface which can be used to control the workflow, as an alternative to the script approach used here. Some functions and modules require direct manipulation, or multiple versions of a similar function, for the full breadth of use cases, and development should involve moving all possible inputs to few top-level functions. This was not exhaustively completed here due to the extensive time cost.

References

1. Bridgelall R, Askarzadeh T, Tolliver DD. Introducing an efficiency index to evaluate eVTOL designs. *Technological Forecasting and Social Change.* 2023;191:122539.
2. Song K, Yeo H. Development of optimal scheduling strategy and approach control model of multicopter VTOL aircraft for urban air mobility (UAM) operation. *Transportation Research Part C: Emerging Technologies.* 2021;128:103181.
3. Poynton J. FINAL YEAR MENG PROJECT, PROJECT SCOPING AND PLANNING, Mission Profile Generation for Urban Air Mobility. In: University of Bath; 2023.
4. VX4 programme moves to the next phase [press release]. Vertical Aerospace Ltd.31 August 2023.
5. French Evolution: What is Volocopter Doing in Paris to Fly there First by 2024? 2023; https://www.volocopter.com/en/blog/french-evolution-first-to-fly-2024. Accessed 17/11/2023.
6. Introducing the first electric vertical take-off and landing jet. (N.D.); https://lilium.com/jet. Accessed 17/11/2023.
7. Gettinger D. Airbus Reveals NextGen eVTOL Concept. *Vertiflite* 2021; https://evtol.news/news/airbus-reveals-nextgen-evtol-concept. Accessed 17/11/2023.
8. Garrow LA, German BJ, Leonard CE. Urban air mobility: A comprehensive review and comparative analysis with autonomous and electric ground transportation for informing future research. *Transportation Research Part C: Emerging Technologies.* 2021;132:103377.
9. Cho S-H, Kim M. Assessment of the environmental impact and policy responses for urban air mobility: A case study of Seoul metropolitan area. *Journal of Cleaner Production.* 2022;360:132139.
10. Alessandro Bacchini, Cestino E. Electric VTOL Configurations Comparison. *Aerospace.* 2019;6(3):26.
11. Ugwueze O, Statheros T, Horri N, Bromfield MA, Simo J. An Efficient and Robust Sizing Method for eVTOL Aircraft Configurations in Conceptual Design. *Aerospace.* 2023;10(3).
12. Vegh JM, Scott R, Basset P-M, Perret R, Beals N, Singh R. A COMPARISON OF THREE CONCEPTUAL DESIGN APPROACHES APPLIED TO AN ELECTRIC DISTRIBUTED LIFT AIRCRAFT Paper presented at: The 49th Annual European Rotorcraft Forum2023; Bückeburg, Germany.
13. Lim Kin Yip S, Boon ZY, Wang J. Trajectory Modelling of Transition Regime Using Blade Element Method for Different eVTOL Aircraft Configurations. Paper presented at: 49th European Rotorcraft Forum 2023; Bückeburg, Germany.
14. Hartman D, Foster J, Hartman C. An Interface Specification for Urban Air Mobility Performance Models to Support Air Traffic Management Research. Paper presented at: AIAA SciTech Forum and Exposition 2023; National Harbor, MD.
15. Yoon J, Nguyen NV, Choi S-M, Lee J-W, Kim S, Byun Y-H. *Multidisciplinary General Aviation Aircraft Design Optimizations Incorporating Airworthiness Constraints.* 2010.
16. Hartman D, Foster J, Hartman C. Performance Modeling of Urban Air Mobility Vehicles to Support Air Traffic Management Research. Paper presented at: AIAA SciTech Forum and Exposition 2023; National Harbor, MD.
17. Sarkar M, Yan X, Girma A, Homaifar A. A Comprehensive eVTOL Performance Evaluation Framework in Urban Air Mobility. Intelligent Systems Conference (IntelliSys) 2022; Amsterdam.
18. Botero EM, Wendorff A, MacDonald T, et al. SUAVE: An Open-Source Environment for Conceptual Vehicle Design and Optimization. In: *54th AIAA Aerospace Sciences Meeting.* American Institute of Aeronautics and Astronautics; 2016.
19. Lee Dh, Kim C-j, Lee Sh. Development of Unified High-Fidelity Flight Dynamic Modeling Technique for Unmanned Compound Aircraft. *International Journal of Aerospace Engineering.* 2021;2021:5513337.
20. Lee B-S, Tullu A, Hwang H. Optimal Design and Design Parameter Sensitivity Analyses of an eVTOL PAV in the Conceptual Design Phase. *Applied Sciences.* 2020;10:5112.
21. Bilici M, Karalı M. Modeling and Control of a Fixed-Wing High-Speed UAV. *International Journal of Aviation Science and Technology.* 2022;vm03:35-44.

22. Pavan N. *Design of Tiltrotor VTOL and Development of Simulink Environment for Flight Simulations*: Department of Aerospace Engineering, Indian Institute of Space Science and Technology; 2020.

23. Bertin JJ, Cummings RM. *Aerodynamics for Engineers.* Pearson/Prentice Hall; 2009.

24. Lee H, Sengupta B, Araghizadeh MS, Myong RS. Review of vortex methods for rotor aerodynamics and wake dynamics. *Advances in Aerodynamics.* 2022;4(1):20.

25. Proulx-Cabana V, Nguyen MT, Prothin S, Michon G, Laurendeau E. A Hybrid Non-Linear Unsteady Vortex Lattice-Vortex Particle Method for Rotor Blades Aerodynamic Simulations. *Fluids.* 2022;7(2).

26. Caries V, Montsarrat C, Boudet J, Lippinois E. Vortex Lattice Method for Fan Tip-Flow Modeling. European Turbomachinery Conference; 2023-04-24, 2023; Budapest, Hungary.

27. Wilson M. ME40343: Advanced Helicopter Dynamics, ROTOR AERODYNAMICS In: University of Bath; 2022.

28. Sagan D, Krishnamurthy P, Lau M, Augustine J, German B. Trim of Over-Actuated VTOL Aircraft in Transition. In: *AIAA AVIATION 2021 FORUM.* American Institute of Aeronautics and Astronautics; 2021.

29. Daud Filho AC, Belo EM. A tilt-wing VTOL UAV configuration: Flight dynamics modelling and transition control simulation. *Aeronautical journal.* 2024;128(1319):152-177.

30. Okasha M, Jordan K, Islam M. Design and Experimental Comparison of PID, LQR and MPC Stabilizing Controllers for Parrot Mambo Mini-Drone. *Aerospace.* 2022;9(6):298.

31. Sadraey MH. *Aircraft Design: A Systems Engineering Approach.* Wiley; 2013.

32. Raymer D. *Aircraft Design: A Conceptual Approach, Sixth Edition.* 2018.

33. Editors at Mathworks. 6DOF (Euler Angles). 2023; https://uk.mathworks.com/help/aeroblks/6dofeulerangles.html. Accessed 18/11/2023.

34. Wang Z, Wang Y, Cai Z, Zhao J, Liu N, Zhao Y. Unified Accurate Attitude Control for Dual-Tiltrotor UAV with Cyclic Pitch Using Actuator Dynamics Compensated LADRC. *Sensors.* 2022;22(4).

35. Melin T. A Vortex Lattice MATLAB Implementation for Linear Aerodynamic Wing Applications. 2000.

36. Katz and J, Plotkin A. Low-Speed Aerodynamics, Second Edition. *Journal of fluids engineering.* 2004;126(2):293-294.

37. Septiyana A, Hidayat K, Rizaldi A, et al. *Analysis of aerodynamic characteristics using the vortex lattice method on twin tail boom unmanned aircraft.* Vol 22262020.

38. Joseph C. A Comprehensive Julia implementation of the Vortex Lattice Method. 2021; https://github.com/byuflowlab/VortexLattice.jl/releases. Accessed 22/04/2024.

39. Su W, Qu S, Zhu G, Swei SS-M, Hashimoto M, Zeng T. Modeling and control of a class of urban air mobility tiltrotor aircraft. *Aerospace Science and Technology.* 2022;124:107561.

40. Poynton J. GBDP Drones. Team 1: Pytheas. STABILITY AND CONTROL In: University of Bath; 2023.

41. Wang Y, Zhu H, Zhao Z, Zhang C, Lan Y. Modeling, System Measurements and Controller Investigation of a Small Battery-Powered Fixed-Wing UAV. *Machines.* 2021;9(12).

42. Beard RW. *Small unmanned aircraft : theory and practice.* Course Book. ed. Princeton, N.J.: Princeton, N.J. : Princeton University Press; 2012.

43. Gryte K, Hann R, Alam M, Roháč J, Johansen TA, Fossen TI. Aerodynamic modeling of the Skywalker X8 Fixed-Wing Unmanned Aerial Vehicle. Paper presented at: 2018 International Conference on Unmanned Aircraft Systems (ICUAS); 12-15 June 2018, 2018.

44. Minervini A, Godio S, Guglieri G, Dovis F, Bici A. Development and Validation of a LQR-Based Quadcopter Control Dynamics Simulation Model. *Journal of Aerospace Engineering.* 2021;34(6):04021095.

45. Ahmad F, Kumar P, Bhandari A, Patil PP. Simulation of the Quadcopter Dynamics with LQR based Control. *Materials Today: Proceedings.* 2020;24:326-332.

46. Appleton W. *Aeromechanics Modelling of Tiltrotor Aircraft* [Ph.D.]. England, The University of Manchester (United Kingdom); 2020.

47. Dahl JH. *Model-based control of UAV in icing conditions*. January 16th 2023: Department of Engineering Cybernetics, NTNU; 2023.

48. Benson T. Shape Effects on Drag. (N.D.); https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/shaped.html#:~:text=A%20flat%20plate%20has%20Cd,a%20Cd%20that%20varies%20from%20. Accessed 02/05/2024.
49. Dreier ME. Introduction to Helicopter and Tiltrotor Flight Simulation (2nd Edition). In: American Institute of Aeronautics and Astronautics (AIAA).
50. Sheng H, Zhang C, Xiang Y. Mathematical Modeling and Stability Analysis of Tiltrotor Aircraft. *Drones.* 2022;6(4).
51. Wang H, Sun W, Zhao C, Zhang S, Han J. Dynamic Modeling and Control for Tilt-Rotor UAV Based on 3D Flow Field Transient CFD. *Drones.* 2022;6(11).
52. Fan Y, Wang X, Hu Z, Zhang K. Nonlinear modeling and transition corridor calculation of a tiltrotor without cyclic pitch. In*.* Vol 355. Les Ulis: EDP Sciences; 2022.
53. Zhou P, Chen R, Yuan Y, Chi C. Aerodynamic Interference on Trim Characteristics of Quad-Tiltrotor Aircraft. *Aerospace.* 2022;9(5).
54. Wang X, Chen J, Zhang Q, Zhang J, Cong H. Research on Innovative Trim Method for Tiltrotor Aircraft Take-Off Based on Genetic Algorithm. *Journal of Sensors.* 2020;2020:8876867.
55. Why Do Cost Functions Use the Square Error? 2024; https://www.geeksforgeeks.org/why-do-cost-functions-use-the-square-error/. Accessed 25/04/2024.
56. Editors at Mathworks. Particle Swarm Optimization. 2024; https://uk.mathworks.com/help/gads/particleswarm.html. Accessed 20/04/2024.
57. Yang X-S. *Recent Advances in Swarm Intelligence and Evolutionary Computation.* 1st 2015. ed. Cham: Cham : Springer International Publishing : Imprint: Springer; 2015.
58. Yang X. Metaheuristic Optimization. *Scholarpedia.* 2011;6:11472.
59. Editors at Mathworks. Tune Particle Swarm Optimization Process. 2024; https://uk.mathworks.com/help/gads/tune-particleswarm-optimization-process.html. Accessed 10/03/2024.
60. Hassan R, Cohanim B, de Weck O, Venter G. A Comparison of Particle Swarm Optimization and the Genetic Algorithm. In: *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference.* American Institute of Aeronautics and Astronautics; 2005.
61. Lin X, Xu Z. Swarm size and inertia weight selection of Particle Swarm Optimizer in system identification. Paper presented at: 2015 4th International Conference on Computer Science and Network Technology (ICCSNT); 19-20 Dec. 2015, 2015.
62. Wilson M. ME40343: Advanced Helicopter Dynamics PART II: ADVANCED AERODYNAMICS & HELICOPTER PERFORMANCE. In: University of Bath; 2022.
63. Leishman JG. *Principles of Helicopter Dynamics.* 2nd ed: Cambridge University Press; 2006.
64. Kashyap M, Lessard L. Guaranteed Stability Margins for Decentralized Linear Quadratic Regulators. *IEEE Control Systems Letters.* 2023;PP:1-1.
65. Nielsen JB. *Robust Visual-inertial Navigation and Control of Fixed-wing and Multirotor Aircraft*. ProQuest Dissertations Publishing, Brigham Young University; 2019.
66. Stevens BL. *Aircraft control and simulation : dynamics, controls design, and autonomous systems.* 3rd ed. Hoboken, New Jersey: Hoboken, New Jersey : Wiley; 2016.
67. Cao-Tri D, Huynh T, Young-Bok K. LQI Control System Design with GA Approach for Flying-Type Firefighting Robot Using Waterpower and Weight-Shifting Mechanism. *Applied Sciences.* 2022;12(18):9334.
68. Markaroglu H, Guzelkaya M, Eksin I, Yesil E. *Tracking Time Adjustment In Back Calculation Anti-Windup Scheme.* 2006.

Appendix A – Sample UAM Aircraft

Table 5 details the full aerodynamic data obtained both in the module designed here, and the open-source software XFLR5. This serves as additional validation of the module. **Table 5** details the mass and inertial properties. Overall geometry may be seen in **Figure 14**.

Table 5: *Aerodynamic derivatives of concept UAM aircraft.*

Aerodynamic Derivatives	UAM 1 – XFLR5	UAM 1 – VLM Module	UAM 2 – XFLR5	UAM 2 – VLM Module
C_{L_α}	4.0266	3.9492	5.9873	6.0200
C_{D_α}	0.00479	0.0994	0.6821 (Viscous)	0.0957
C_{m_α}	-0.3821	-0.3679	-0.3681	-0.3683
C_{Y_β}	-0.2742	-0.1997	-0.1356	-0.1429
C_{l_β}	-0.1822	-0.1796	0.00212	0.00258
C_{n_β}	0.04835	0.0421	0.04846	0.04612
$C_{L_{\delta_E}}$	0.2020	0.1977	0.42351	0.4127
$C_{m_{\delta_E}}$	0.3178	0.2941	-1.2898	-1.1562
$C_{l_{\delta_A}}$	0.1289	0.1366	0.2092	0.2079
$C_{n_{\delta_A}}$	-0.0111	-0.0107	0.01059	0.01093
$C_{l_{\delta_R}}$	-0.0124	-0.0126	-	-
$C_{n_{\delta_R}}$	-0.03221	-0.03229	-	-
$C_{Y_{\delta_A}}$	0.01344	0.01532	-0.05944	-0.04678
$C_{Y_{\delta_R}}$	0.1112	0.0984	-	-
C_{Y_p}	-0.1213	-0.1211	-0.02302	0.02683
C_{Y_r}	0.2563	0.2494	0.09118	0.1062
C_{L_q}	4.7578	4.7784	8.7710	8.6743
C_{m_q}	-2.4584	-2.4803	-29.531	-32.769
C_{n_r}	-0.0394	-0.0387	-0.03658	-0.0379
C_{n_p}	0.01765	0.01634	-0.12820	-0.1382
C_{l_r}	0.1123	0.1134	0.28192	0.3138
C_{l_p}	-0.3292	-0.3291	-0.52820	-0.4932
C_{L_0}	0.3004	0.3125	0.5001	0.4958
C_{D_0}	0.0180	0.0167	0.0310	0.0298
C_{m_0}	-0.02	-0.02	-0.2469	-0.2543

Table 6*: Mass data of refererence UAM aircraft.*

UAM	Total Mass [kg]	I_{xx} [kgm^2]	I_{yy}[kgm^2]	I_{zz}[kgm^2]
1	150	107	93	196
2	2280	6400	4500	9400

Appendix B – Example Use Case

MATLAB Example Scripts

Example 1 demonstrates how an aircraft model may be defined, initially by creating geometry objects, then applying VLM.

EXAMPLE 1

```
%% Create Aircraft
VX4 = configDef;
VX4.CG = [1.71 0 0];
VX4.CoL = [2.04 0 0];

wing1 = wing(0.2, 1, 100, 1, 0.3, [3.1 0 0], [0 0 0], 2*pi, 3.5, 0.12, 1.625,"2412");
aileron = controlSurf(0.3, "TE", 2*pi, 1, 0.12);
aileron.deflection = 0*(pi/180); aileron.roll = 1;
wing1 = addCS(wing1, aileron, 2.2);

VX4 = addLift(VX4, wing1);

wing2 = wing(0.2, -1, 100, 1, 0.3, [3.1 0 0], [-0 0 0], 2*pi, 3.5, 0.12, 1.625,"2412");
aileron = controlSurf(0.3, "TE", 2*pi, 1, 0.12);
aileron.deflection = 0*(pi/180); aileron.roll = 1;
wing2 = addCS(wing2, aileron, 2.2);

VX4 = addLift(VX4, wing2);

tail1 = wing(0, -1, 100, 1, 0.9, [0.44 0 0.1], [-0 0 0], 2*pi, 1.5, 0.12, 0.5,"0012");
ruddervator = controlSurf(0.4, "TE", 2*pi, 0.8, 0.12);
ruddervator.deflection = 0*(pi/180);
tail1 = addCS(tail1, ruddervator, 0.5);

VX4 = addLift(VX4, tail1);

tail2 = wing(0, 1, 100, 1, 0.9, [0.44 0 0.1], [0 0 0], 2*pi, 1.5, 0.12, 0.5,"0012");
tail2 = addCS(tail2, ruddervator, 0.5);

VX4 = addLift(VX4, tail2);

rotor1 = rotor(0.4, 1, 1, 1, 2, [4 1.75 0.2], [0 0 0], 3000);
VX4 = addThrust(VX4, rotor1);

rotor2 = rotor(0.4, 1, 1, 1, 2, [4 -1.75 0.2], [0 0 0], 3000);
VX4 = addThrust(VX4, rotor2);

rotor3 = rotor(0.4, 1, 1, 1, 2, [0.25 -2 0.2], [0 0 0], 3000);
VX4 = addThrust(VX4, rotor3);

rotor4 = rotor(0.4, 1, 1, 1, 2, [0.25 2 0.2], [0 0 0], 3000);
VX4 = addThrust(VX4, rotor4);

rudder = controlSurf(0.4, "TE", 2*pi, 0.4, 0.12);
tail3 = wing(0, 1, 100, 1, 0.9, [3.6 1.75 0], [-pi/2 + 0.05 0 0], 2*pi, 0.9, 0.12,
0.4,"0012");
tail3 = addCS(tail3, rudder, 0.14);
VX4 = addLift(VX4, tail3);

tail4 = wing(0, 1, 100, 1, 0.9, [3.6 -1.75 0], [-pi/2 - 0.05 0 0], 2*pi, 0.9, 0.12,
0.4,"0012");
```

```
tail4 = addCS(tail4, rudder, 0.14);
VX4 = addLift(VX4, tail4);

plotConfiguration(VX4)


%% Set Airflow conditions
airflow = struct();
airflow.U = 1;
airflow.alpha = 1*pi/180;
airflow.beta = 0;
airflow.p = 0;
airflow.q = 0;
airflow.r = 0;

%% Run VLM
lift3 = [wing1 wing2 tail1 tail2 tail3 tail4];
Sref = 11.88;
cref = 1.836;
bref = 7;

referenceGeo = struct();
referenceGeo.Sref = Sref;
referenceGeo.cref = cref;
referenceGeo.bref = bref;

N_obj = [13 10; 13 10; 7 7; 7 7; 5 5; 5 5;]; % number of panels per component in lift3

alpha = ([0.1 0.21 0.3])*pi/180;
beta = [0 0.1 0.2]*pi/180;
pqr = [0.009 0.01 0.011]*pi/180;

stabilityDeriv = getStabilityDeriv(lift3, airflow, alpha, beta, referenceGeo, 0, N_obj,
[0 1 2 3; 0 1 2 3; 0 1 2 3;]*pi/180, pqr, VX4.CG, VX4.CG);
```

Example 2 gives an example script for determining an optimal K_i matrix (LQI gain).

EXAMPLE 1

```
%% Get Points for Reference
load('trimUAM1', 'trim')

X = reshape(trim.X,size(trim.X,1),[]); % reshape to linear indices (no stages)
U = reshape(trim.U,size(trim.U,1),[]);
Ut = [U(1:3,:); reshape(trim.Ut, size(trim.Ut,1),[])];

idx = [5, 119, 95, 68;
       135, 31, 59, 73;
       148, 38, 98, 80;
       25, 110, 54, 86 ]; % indices of analysis points (1 sample from each mode)

sysMat = lineariseTrimFull(VX4, VX4.refGeo, coefficients, 'trimUAM1', 1e-5);

Xes = X(:,1);  % set initial condition
```

```
%% Iterate
sysMat_LQI = augmentSys(sysMat);
[Q, R] = solveQR_LQI(sysMat_LQI, idx, X, Ut);

% Save Results in File
save('QRvalsLQI_new.mat', 'Q', 'R', 'idx')

%% Create Gain Matrix
Kpi = createGainMat(sysMat_LQI, 'QRvalsLQI_new.mat', 'trimUAM1');
```

Appendix C – Supplementary Equations

The rotational matrices $\boldsymbol{R}_b^e$ and $\mathbf{W}$ are given by[19]

$$\boldsymbol{R}_b^e = \begin{bmatrix} \cos(\theta)\cos(\psi) & \sin(\varphi)\sin(\theta) - \cos(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \cos(\theta)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{D.1}$$

$$\boldsymbol{W} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \tag{D.2}$$