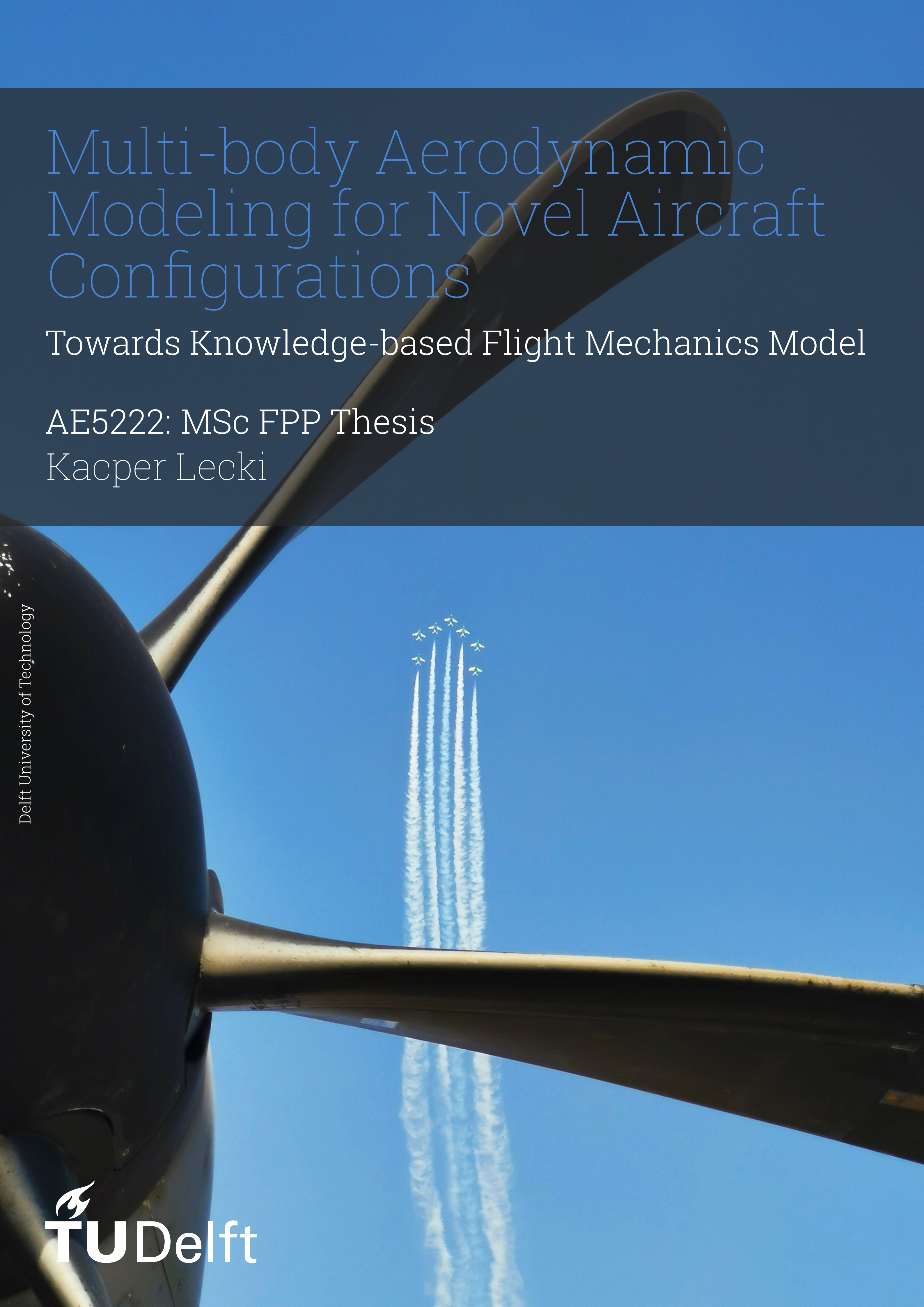


Multi-body Aerodynamic Modeling for Novel Aircraft Configurations

Towards Knowledge-based Flight Mechanics Model

AE5222: MSc FPP Thesis

Kacper Lecki

Delft University of Technology

Multi-body Aerodynamic Modeling for Novel Aircraft Configurations

Towards Knowledge-based Flight Mechanics Model

by

Kacper Lecki

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday January 17, 2025 at 3:00 PM.

Student Name	Student Number
Kacper Lecki	4675754

Project Duration:	March, 2024 - January, 2025	
Faculty:	Faculty of Aerospace Engineering, Delft	
Thesis committee:	Dr. C. Varriale,	TU Delft, Supervisor
	Dr.ir. G. La Rocca	TU Delft, Chair
	Dr. A. Bombelli	TU Delft, Examiner

Cover: Formation flight at Dubai Airshow, UAE, Captured by the author
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Preface

With this report and accompanying defense, I am concluding the degree of Master of Science in Aerospace Engineering. The path to completing this qualification has been shaped by challenging circumstances well beyond the academic realm. I deeply appreciate everyone who supported me on this journey.

First of all, I would like to thank my supervisor Dr. C. Varriale for his patience, guidance, flexibility, and understanding when the research kept changing course. I would also like to thank Dr.ir. G. La Rocca for introducing me to the world of knowledge-based engineering and providing valuable feedback at the crucial milestones of this project. Furthermore, I extend my gratitude to Dr. A. Bombelli for being a part of my thesis committee and supporting me in fulfilling the requirements for this degree.

I would like to acknowledge everyone who enabled me to study at TU Delft. I am deeply grateful to my friends, who generously opened their homes to me, providing both a place to stay and invaluable support during a challenging time. I am profoundly appreciative of my teammates for their patience, flexibility, and collaborative spirit in tackling group assignments over a distance. I would also like to thank everyone who made my study time enjoyable and fulfilling. To my family, I extend my heartfelt gratitude for their constant encouragement and selflessness. Their readiness to shoulder my duties and responsibilities during this time was invaluable to my ability to succeed.

Kacper Lecki
Delft, January 2025

Summary

Urban congestion and the environmental impact of traditional oil-based transport continue to increase. Hence, there is a pressing need for alternative transportation systems. Being able to design an aerial vehicle with previously developed components would make it easier to introduce novel aircraft configurations to fill the new air mobility market. Traditionally, an aircraft's flying qualities are evaluated by analyzing the whole geometry as a single (rigid) body. This work aims to explore a paradigm in which the isolated components are analyzed separately and combined together in a multi-body system representative of the whole geometry.

Existing methods and tools are explored at the intersection of multi-domain systems, aerodynamic modeling, and flight mechanics. This work reviews early efforts to link geometric modeling with physical simulations, including frameworks such as CEASIOM and the Design Engineering Engine (DEE). These were introduced using the methodology of Knowledge-Based Engineering (KBE) aiming to streamline the development of new aircraft. This approach integrates multi-fidelity analysis tools to reduce time, cost, and expertise requirements in aircraft design. Innovations such as high-level primitives/CAD templates and multiphysics information models were proven useful in these tools. These frameworks provide parametric modeling environments and integrate with aerodynamic analysis modules, but often lack comprehensive automation or prescribing interactions between components. Aerodynamic analysis tools such as AVL, Tornado, and XFLR5 are evaluated for their ability to estimate stability derivatives and simulate flight mechanics with reasonable accuracy. The review concludes that methods based on vortex filaments offer a good balance between fidelity and computational cost. Flight mechanics frameworks such as PHALANX and Modelica-based libraries, which integrate multi-domain models, are also explored. The analysis highlights the need for a flexible, automated, and robust KBE framework that combines these diverse approaches and supports many moving components with prescribed interactions between them.

This need is aligned with the aim of the FLAPERON (FLight mechanics Analysis and PERformance OptimizatioN) framework. By combining object-oriented architecture with multidisciplinary analysis capabilities, FLAPERON aims to provide a scalable and adaptable solution for the design and evaluation of various aircraft configurations supporting multi-body analysis. Core elements include a robust method for creating and placing components with hierarchical reference frames and classes to represent aerodynamic, inertial, and elastic properties.

This project employs the FLAPERON toolbox to assess the impact of vortex-filament-based aerodynamic interactions on the simulated aircraft's response when considering aerodynamic data only at the component level. To achieve this goal, wings and the model of their downwash are added to the FLAPERON toolbox working towards the high-level primitives. Physical networks are employed to simulate the aircraft's response in environments configured to assess key stability characteristics in symmetrical flight. Aerodynamic data is prescribed at a component level for a configuration that includes a main wing, a horizontal stabilizer, and a vertical stabilizer. Vortex filaments are modeled for the main wing and the horizontal stabilizer. Increasingly complex scenarios are simulated, starting with no interactions and advancing to systems with coupled interactions.

Including the downwash from the main wing on the horizontal stabilizer requires the increase of the stabilizer's incidence angle to keep the aircraft trimmed. The lift-curve slope is lower compared to the case with no interactions prescribed as extracted from the state-space representation of the multi-body system. This is as expected as the increase in the angle of attack of the main wing creates a larger downwash at the horizontal stabilizer. The simulated time-to-half-amplitude of the short-period motion of the aircraft increases by less than 1%. The lift-curve slope (as well as $C_{l_{\dot{\alpha}}}$, $C_{m_{\alpha}}$, and $C_{m_{\dot{\alpha}}}$) seems not to be impacted when considering the change of the effective angle of attack of the main wing due to the vortex filament of the horizontal stabilizer. The consideration of the velocity induced by the horizontal stabilizer on the main wing decreased the time-to-half-amplitude by over 18%.

This shows that multi-body analysis of flying qualities is possible based on the data prescribed at the component level. The introduced interactions change the simulated response of the aircraft in accordance with predictions. This suggests that aircraft design with already prepared components can be significantly streamlined compared to current methods. One should note, however, that the prescribed simple lifting-line-based interactions in this project result in force and moment coefficients that differ significantly from the ones obtained when analyzing the whole aircraft geometry as a single body in DATCOM. The discrepancies ought to be studied more to advance the modeling of systems consisting of multiple interacting components in the context of aircraft design.

Furthermore, the FLAPERON's adaptability was confirmed as simulations incorporated environmental factors, such as the International Standard Atmosphere (ISA), aerodynamic interactions, and multiple moving bodies. The recommendations for automation of the analysis within the toolbox are provided based on the identification of the repetitive steps in the simulation setup. Overall, the results validate FLAPERON's potential as a flexible tool for early-stage aircraft design. The research sets a foundation for advancing the integration of KBE into flight mechanics modeling, paving the way for faster, more cost-effective development of innovative aircraft configurations.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
AVL	Athena Vortex Lattice
CAD	Computer-Aided Design
CFD	Computational Fluid Dynamics
DEE	Design Engineering Engine
DES	Detached Eddy Simulation
DoF	Degrees of Freedom
DNS	Direct Numerical Simulation
FMM	Flight Mechanics Modeling
FOI	Foreign Object Ingestion
HLCt	High-level CAD template
HLP	High-level primitive
ISA	International Standard Atmosphere
KBE	Knowledge-based Engineering
LES	Large Eddy Simulation
MBSE	Model-based Systems Engineering
MDA	Multidisciplinary Analysis
MDO	Multidisciplinary Optimization
MMG	Multi-Model Generator
NS	Navier-Stokes
(U)RANS	(Unsteady) Reynolds-averaged Navier-Stokes
SFT	Subscale flight test
UAV	Unmanned Aerial Vehicle
VLM	Vortex Lattice Model

1

Introduction

Roads are becoming increasingly congested, and cities are reaching their limits in terms of transportation capabilities across the world. The vehicle-kilometers traveled (VKT) have been consistently increasing and research implies that increased provision of roads or public transport is not able to relieve congestion [1]. This suggests that a new transportation system such as air mobility may be needed to decrease the significant cost and climate impact of commuting.

Furthermore, there is soaring concern regarding the oil-based economy in terms of climate impact. In particular, aviation is forecasted to continue growing faster than the global economy which suggests that its climate impact will increase in absolute and relative terms [2]. This results in huge social pressure to move towards more climate-friendly air transportation.

To answer this need, the aerospace community is rallying around the mission to create sustainable flying machines. The resulting new air mobility market is predicted to grow to $74 to $641 billion US space by 2030 [3]. The market for air travel will have an unprecedented variety. Likely many new players will enter the market with new ideas and unconventional designs that are more purpose-specific than the aircraft of our era.

To develop new configurations, a couple of barriers have been identified:

- Regulation – it is difficult for new players to obtain approvals for untested aircraft, and it is difficult to test aircraft without approvals.
- Cost – it is expensive to develop, build, and test aircraft and it's becoming increasingly challenging to secure funding in the space without a working proof of concept.
- Pilot availability – it is difficult to find test pilots willing to fly unproven concepts.
- Hardware challenges – it may be difficult to manufacture new designs as they may require new methods or technologies.
- Air traffic management – the growing number of flying vehicles increases the chance of accidents.
- Infrastructure – current infrastructure in cities was developed with road transport in mind and thus would require adjustments to fit the new transport system.

To tackle the first two barriers, a change in design methodology was proposed. While the big airframe manufacturers tend to use Computer-Aided Design (CAD) throughout the whole design process, it is more appropriate to use parametric geometry modelers to reduce the required cost, time, and expertise in development [4]. To further decrease the time and effort required and allocate more resources towards the creative aspects of aircraft design, a Knowledge-based Engineering (KBE) approach has been suggested [5]. Thanks to building the designs from predefined primitives, such frameworks can be adapted to be included in Multidisciplinary Analysis and Optimization (MDA/MDO). Furthermore, it is possible to couple such frameworks with multi-fidelity analysis modules such as Athena Vortex Lattice (AVL) or automated CFD analysis. This allows for rapid and accurate estimation of key performance indicators at different design stages.

The challenge that remains is the necessity to re-run the aerodynamic analysis every time there is a change in the geometry or relative positions of the components. There are situations in which one may want to use a library of available parts. These include cases when one wants to use third-party components or one already developed a component for another purpose. In such situations, it seems wasteful to re-run aerodynamic analysis for the whole configurations. Especially if high-fidelity aerodynamic data is available from CFD or wind tunnel experiments for the component.

Wald, Fay, and Gleich [6] identify the pathways to achieve better fuel efficiency as the popularization of composites, new propulsion technologies, aerodynamic optimization of winglets, space reorganization, and change in construction. In fact, throughout history, aircraft were retrofitted with winglets to improve aerodynamic performance [7]. In these situations, designing winglets could be computationally cheaper if they were considered as an addition to the existing system with already immense amounts of aerodynamic and flight performance data available.

Additionally, there is a trend in the increase in engine diameter and reduced ground clearance that raises concerns about foreign object ingestion (FOI) [8]. An aircraft manufacturer may consider switching from a low-wing to a high-wing configuration to accommodate a larger engine and reduce the costs associated with FOI. Since they already have production and assembly of their original wing design deployed, it could be easier and more cost-effective to only consider different placements of this wing for the new configuration. It would be ideal if the aerodynamic data of this wing could be re-used rather than relying on running new analyses for each new placement considered. This is especially important for placement optimization as aerodynamic analysis must be processed at each iteration.

One should not ignore the growing popularity of Unmanned Aerial Vehicles (UAVs). They were shown to have a wide range of potential applications across a multitude of industries [9]. UAVs are commonly built from off-the-shelf components with purpose-built airframes connecting them [10]. Hence, the properties of individual components are rarely modifiable but their placement can be easily changed.

This presents a need for modeling aircraft as a composite of its constituent parts with the (aerodynamic) models attached to the components rather than being considered only at the aircraft level. The goal of this work is to fill this gap by laying a foundation for an aircraft-level analysis based on its components in the KBE framework working towards an automatic and configuration-agnostic flight mechanics model.

In this chapter, the background information for the main pillars of this project is provided. In section 1.1, the theoretical review for flight mechanics and stability is given discussing the interaction between forces and inertia acting on an aircraft. In section 1.2, relevant aerodynamic models are discussed. In section 1.3, the review of approaches to multi-domain systems is provided. In chapter 2, a literature study is reported discussing the state-of-the-art methods in the relevant disciplines. Chapter 3 presents the methodology of this work including details of the model used. In chapter 4 the results of the research are presented. The work is concluded and discussed in the broader context of the industry in chapter 5.

1.1. Flight Mechanics

To properly understand flight mechanics modeling, it is essential to grasp the relevant contributing factors. Flight mechanics is a term that encompasses the statics, kinematics, and dynamics of aircraft. These are defined by the interaction between forces acting on the aircraft and its inertia (mass and moments of inertia) and describe fully the motion of aircraft with 6 degrees of freedom (DoF): translational motion in 3 directions and rotational motion related to pitch, roll, and yaw of the vehicle. There are more complex models allowing for the motion of aircraft's components with respect to each other, for example through bending. These models are much more convoluted and are considered out of the scope of this work.

1.1.1. Forces

Traditionally, if the flying vehicle is considered a point mass, the forces have been split into four main components: weight (W), lift (L), drag (D), and thrust (T) [11]. More generally, this could be categorized into aerodynamic, propulsive, and inertial forces. Physically speaking, lift and drag (aerodynamic forces) are related to pressure and shear forces that result from the body moving in a fluid as discussed in section 1.2. Thrust is related to the aircraft's propulsion system and is not discussed in detail in this work as its modeling is out of scope for this project. Oftentimes, the thrust is assumed independent from the aerodynamic forces to make the flight mechanics modeling easier. This assumption holds for most standard transonic transport aircraft as the turbofan engines are usually displaced from the wings. Yet, for the most accurate estimation of aircraft dynamics of novel configurations, wing-propeller interactions should be included in the calculation of the forces [12]. Note that this creates a requirement for a configuration-agnostic flight mechanics model to allow the user to define the interaction between various components.

1.1.2. Inertia

Inertia quantifies the resistance of the body to change its velocity. In the case of translational motion, the relevant parameter is the total mass of the object. In the case of rotating motion, not only the total mass but also its distribution is combined to form moments and products of inertia. Note that in general inertia is not constant across the flight due to load redistribution such as fuel motion and consumption. Nonetheless, fuel consumption during most of the maneuvers is small compared to the total mass of the vehicle and hence can be assumed to be quasi-steady [11]. This assumption is rendered invalid in the case of in-flight refueling and morphing or flapping vehicles [13]. Modeling such vehicles and maneuvers is out of scope for this project and, hence, the research will focus on rigid-body aircraft.

1.1.3. Equations of Motion

Aircraft motion can be described by applying Newton's second law to the vehicle. The sample procedure along with transformations between body, wind, and stability reference frames is described in chapter 4 of Yechout, Morris, Bossert, *et al.* [11]. The equations take the form:

$$\begin{aligned} m(\dot{U} + QW - RV) &= F_{G_x} + F_{A_x} + F_{T_x} \\ m(\dot{\mathbf{V}} + RU - PW) &= F_{G_y} + F_{A_y} + F_{T_y} \\ m(\dot{W} + PV - QU) &= F_{G_z} + F_{A_z} + F_{T_z} \end{aligned} \tag{1.1}$$

$$\begin{aligned} \dot{P}I_{xx} + QR\left(I_{zz} - I_{yy}\right) - (\dot{R} + PQ)I_{xz} &= L_A + L_T \\ \dot{Q}I_{yy} - PR\left(I_{zz} - I_{xx}\right) + \left(P^2 - R^2\right) I_{xz} &= M_A + M_T \\ \dot{R}I_{zz} + PQ\left(I_{yy} - I_{xx}\right) + (QR - \dot{P})I_{xz} &= N_A + N_T \end{aligned} \tag{1.2}$$

where U, V, W are velocities in the x, y and z body axes. Subscripts G, A, T refer to gravitational, aerodynamic, and thrust forces respectively. One of this model's conventional assumptions is that the forces can be decomposed and treated separately. P, Q, R are roll, pitch, and yaw rates expressed in the body axes. Related to these are moments in the respective axes (L, M, N).

1.1.4. Stability

When the net external force and moment on the aircraft is zero, the vehicle stays in a steady flight. If attitude and flight conditions allow for a steady flight, the aircraft is said to be trimmed. Stability analysis is related to the vehicle's response if it is perturbed from this state.

The basic evaluation of the stability can be done by considering stability derivatives, for example, C_{m_α}, C_{l_β}, C_{n_β}. These quantities represent a change in moments due to a change in local air velocity relative to the body axes. Note that this is a result of first-order Taylor expansion and thus disregards the impact of higher-order time derivatives. In other words, it follows a quasi-steady analysis. For the aircraft to maintain positive static stability, it must have a restoring moment/force if it is perturbed from the trimmed condition. This will ensure that the system can get back to its equilibrium position after the perturbation.

Traditionally, the equations of motions are linearized to perform analysis of dynamic stability [11]. This is done with the help of perturbation theory and considering a small change to the trim condition. This way the equations of motion can be expressed in terms of stability parameters and the perturbation velocities.

$$
\begin{aligned}
\dot{u} &= -g\theta\cos\Theta_1 + X_u u + X_{T_u} u + X_\alpha \alpha + X_{\delta_e}\hat{\delta}_e \\
\dot{w} - U_1 q &= -g\theta\sin\Theta_1 + Z_u u + Z_\alpha \alpha + Z_{\dot{\alpha}}\dot{\alpha} + Z_q q + Z_{\delta_e}\hat{\delta}_e \\
\dot{q} &= M_u u + M_{T_u} u + M_\alpha \alpha + M_{T_\alpha} \alpha + M_{\dot{\alpha}}\dot{\alpha} + M_q q + M_{\delta_e}\hat{\delta}_e
\end{aligned}
\tag{1.3}
$$

$$
\begin{aligned}
\dot{v} + U_1 r &= g\phi\cos\Theta_1 + Y_\beta \beta + Y_p p + Y_r r + Y_{\delta_a}\hat{\delta}_a + Y_{\delta_r}\hat{\delta}_r \\
\dot{p} - \frac{I_{xz}}{I_{xx}}\dot{r} &= L_\beta \beta + L_p p + L_r r + L_{\delta_a}\hat{\delta}_a + L_{\delta_r}\hat{\delta}_r \\
\dot{r} - \frac{I_{xz}}{I_{zz}}\dot{p} &= N_\beta \beta + N_{T_\beta}\beta + N_r r + N_{\delta_a}\hat{\delta}_a + N_{\delta_r}\hat{\delta}_r
\end{aligned}
\tag{1.4}
$$

where X, Y, Z, M, L, N are stability parameters as defined in tables 6.3 and 6.4 in Yechout, Morris, Bossert, *et al.* [11]. Subscript 1 denotes steady-state value. These equations can be recast using the Laplace transform to analyze the dynamic response of the aircraft. Based on this analysis, the fundamental dynamic modes of the aircraft are established: phugoid, short-period oscillation, aperiodic roll, spiral, and dutch roll. Damping and the period of these motions are important for the determination of the flying qualities of the aircraft. Commonly they also form a benchmark for aircraft simulation in the time domain as they can be easily analyzed in an open-loop simulator.

1.2. Aerodynamics

The analysis of flight mechanics requires knowledge of the stability parameters that depend on (aerodynamic) forces and inertia of the aircraft. Thus, the estimation of the relevant quantities needs a proper understanding of aerodynamics.

Aerodynamics is governed by Navier-Stokes equations (NS) that represent the conservation of momentum, mass and energy in the flow. These equations are a set of nonlinear partial differential equations that can be expressed in Eulerian (fixed point in the flow) or Lagrangian (particle moving with the flow) formalism. The Eulerian representation is shown in equation 1.5 in accordance with Anderson [14].

$$
\begin{aligned}
\frac{\partial(\rho u)}{\partial t} + \nabla\cdot(\rho u \mathbf{V}) &= -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x \\
\frac{\partial(\rho v)}{\partial t} + \nabla\cdot(\rho v \mathbf{V}) &= -\frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + \rho f_y \\
\frac{\partial(\rho w)}{\partial t} + \nabla\cdot(\rho w \mathbf{V}) &= -\frac{\partial p}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} + \rho f_z
\end{aligned}
\tag{1.5}
$$

Solving these along with the conservation of mass and conservation of energy for a particular flight condition and particular geometry gives insight into the forces that act on the aircraft. Peerlings [15] summarizes the equations needed to find fluid properties in viscous rotational unsteady compressible flows as

$$
\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = \mathbf{J}
\tag{1.6}
$$

where

$$
\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho\left(e + \frac{v^2}{2}\right) \end{pmatrix}
$$

$$\mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho u v - \tau_{xy} \\ \rho u w - \tau_{xz} \\ \rho\left(e + \frac{V^2}{2}\right) u + u p - k\frac{\partial T}{\partial x} - u\tau_{xx} - v\tau_{xy} - w\tau_{xz} \end{pmatrix}$$

$$\mathbf{G} = \begin{pmatrix} \rho v \\ \rho u v - \tau_{yx} \\ \rho v^2 + p - \tau_{yy} \\ \rho v w - \tau_{yz} \\ \rho\left(e + \frac{v^2}{2}\right) v + v p - k\frac{\partial T}{\partial y} - u\tau_{yx} - v\tau_{yy} - w\tau_{yz} \end{pmatrix}$$

$$\mathbf{H} = \begin{pmatrix} \rho w \\ \rho u w - \tau_{zx} \\ \rho v w - \tau_{zy} \\ \rho w^2 + p - \tau_{zz} \\ \rho\left(e + \frac{v^2}{2}\right) w + w p - k\frac{\partial T}{\partial z} - u\tau_{zx} - v\tau_{zy} - w\tau_{zz} \end{pmatrix}$$

$$\mathbf{J} = \begin{pmatrix} 0 \\ \rho f_x \\ \rho f_y \\ \rho f_z \\ \rho\left(u f_x + v f_y + w f_z\right) + \rho\dot{q} \end{pmatrix}$$

Unfortunately, these equations are nonlinear for most real situations and, thus, difficult to solve. Various simplifications can be applied to reduce the complexity of the calculation but only a handful result in exact solutions. The most common assumptions involve steady, inviscid, and/or incompressible flow.

In general, the only way to obtain flow properties from these equations is to solve them numerically. The solution with the fewest assumptions despite potential numerical errors can be obtained using Direct Numerical Simulation (DNS) [16]. This method aims to resolve the turbulence at smallest and largest scales, and does not use any approximations. Unsurprisingly, this method is prohibitively expensive computationally. Argyropoulos and Markatos [17] estimate that the commercial application of DNS will be possible with a computational power ≈ 1750 times of the current most powerful supercomputer. Hence, some simplifications are necessary to use aerodynamic modeling for flight mechanics.

To reduce the computational requirements, turbulence can be modeled on some scales. Traditionally, this has been achieved by projecting the flow domain on the grid and modeling turbulence on subgrid scales. This is accomplished by Large Eddy Simulations (LES) and Detached Eddy Simulations (DES) that model the largest scales directly [18]. A more commercially common method, (Unsteady) Reynolds-averaged Navier-Stokes equations (URANS/RANS) is based on modeling turbulence at all spatial scales [19]. These methods are still too computationally expensive to be readily deployed in the KBE framework so further simplifications are required.

At high Reynolds numbers, it can be noticed that in part of the flow domain viscous forces are negligible. For traditional aircraft configurations, only the flow in the proximity of the aircraft is dominated by viscous forces. Hence, for the majority of the flow, inviscid Euler equations can be used. This becomes invalid for low Reynolds number flows and flows at high angles of attack.

If the flow is further assumed to be irrotational ($\nabla \times \mathbf{V} = 0$), the velocity in the flow can be described by a gradient of a scalar function [14].

$$\nabla\phi = \mathbf{V}$$

Assuming steady isentropic flow and no body forces, the full potential equation can be obtained [14].

$$\begin{aligned}&\left[1-\frac{1}{a^2}\left(\frac{\partial\phi}{\partial x}\right)^2\right]\frac{\partial^2\phi}{\partial x^2}+\left[1-\frac{1}{a^2}\left(\frac{\partial\phi}{\partial y}\right)^2\right]\frac{\partial^2\phi}{\partial y^2}+\left[1-\frac{1}{a^2}\left(\frac{\partial\phi}{\partial z}\right)^2\right]\frac{\partial^2\phi}{\partial z^2}\\&-\frac{2}{a^2}\left(\frac{\partial\phi}{\partial x}\right)\left(\frac{\partial\phi}{\partial y}\right)\frac{\partial^2\phi}{\partial x\partial y}-\frac{2}{a^2}\left(\frac{\partial\phi}{\partial x}\right)\left(\frac{\partial\phi}{\partial z}\right)\frac{\partial^2\phi}{\partial x\partial z}-\frac{2}{a^2}\left(\frac{\partial\phi}{\partial y}\right)\left(\frac{\partial\phi}{\partial z}\right)\frac{\partial^2\phi}{\partial y\partial z}=0\end{aligned} \tag{1.7}$$

where

$$a^2=a_\infty^2-\frac{\gamma-1}{2}\left[\left(\frac{\partial\phi}{\partial x}\right)^2+\left(\frac{\partial\phi}{\partial y}\right)^2+\left(\frac{\partial\phi}{\partial z}\right)^2\right]$$

as presented by Peerlings [15] citing Anderson [14]. This is a single nonlinear differential equation combining the conservation of mass, momentum, and energy. It can be linearized by considering small fluctuations and reduced to a form known as Prandtl-Glauert equation [20].

Finally, if compressibility effects are neglected, the full potential equation is reduced to a linear Laplace equation. This is regarded as valid for flows $M_\infty < 0.3$. The equation takes the following form[14].

$$\nabla^2\mathbf{V}=0 \tag{1.8}$$

Since this equation is linear and has exact solutions, the most common way to solve arbitrary geometry is to approximate it as a sum of the exact solutions for simpler flows. The elementary solutions include [14]:

- uniform flow
- source flow
- doublet flow
- vortex flow

One should note that the most relevant for evaluating forces and hence the most important for flight mechanics is vortex flow. This is the only flow that results in a force. This can be explained by the Kutta-Joukowski theorem noting that the vortex flow results in non-zero circulation in the flow.

$$L'=\rho_\infty V_\infty\Gamma \tag{1.9}$$

It is worth noting that for lifting geometries, the inflow conditions and the vortex strength (lift) are dependent on each other and thus require iterations to obtain the flowfield. There are methods for circumventing this requirement and they will be discussed in section 2.3. In this project, the aerodynamic modeling of the bodies whose main purpose is not generation of aerodynamic forces such as fuselage is neglected. In reality, there are aerodynamic interactions between all bodies in the flowfield but they are considered out of scope for this project.

Considering that flight mechanics involves inherently unsteady problems, it is also worth discussing the time-dependence of aerodynamics. The basics of the unsteady analysis were developed in Wagner [21], Theodorsen [22], and Karman and Sears [23]. These works despite their age still provide valuable insight into unsteady aerodynamics [24]. According to these works, the aerodynamics of the airfoil can be modeled by the vortex flow centered at the quarter chord and vorticity shed from the trailing edge (and subsequently convected) that is equal in magnitude and opposite to the change of circulation on the airfoil. This is a result of the second Helmholtz theorem. This means that there's a delay between the change in the angle of attack and the expected value of lift.

Furthermore, for the full aircraft configuration, the shed vorticity may become important for the components downstream of the wing and control surfaces in particular. While the shed vortex is moving only away from the airfoil it is likely to have a nonmonotonic impact on the lift generated by the horizontal stabilizer. Evaluation of this effect is out of the scope of this work but could be an interesting venture for another project (see Greenwell [25] for a review of the related aerodynamic modeling).

1.3. Multi-domain systems modeling

As one can see, flight mechanics modeling is an interdisciplinary field. As such, it involves multiple domains that are interacting with each other. Hence, it is worthwhile to discuss the background of the multi-domain analysis in the context of this work.

In previous sections, several sets of equations each for a separate domain were presented. Describing a multidomain system as a set of equations is the most direct representation of any system. This allows for applying traditional mathematical analysis. The biggest disadvantage of this approach is the difficulty in gaining insight into the governing physical processes [26]. This methodology is present in most academic textbooks discussing flight mechanics including Yechout, Morris, Bossert, *et al.* [11].

Paynter [27] introduced the concept of bond graphs. They are an example of a graphical representation of (multidomain) physical systems. It reduces the required abstraction when analyzing the system and allows for physical insight into the energy flow in the system. Note that the energy/power is domain independent and hence it allows for integrating systems of multiple domains into a single compound system. Within the scope of this methodology, the bonds (links) between elements carry information about *flow* (rate of generalized displacement) and *effort* (energy per unit of displacement) variables. The product of the two is the instantaneous power flowing through this bond. For example, in a mechanical system, the flow property is velocity and the effort is force. The physical components can be translated into bond-graph elements. The bond graph methodology distinguishes the following elements based on their impact on the physical systems:

- Sources (S)
- Hyperance (H)
- Compliance (C)
- Resistance (R)
- Inertance (I or L)
- Abrahance (A)
- Magnance (M)

Additionally, some elements allow for changing the effort/flow variables in the bond without power loss (the power remains constant). Physically, these correspond to components such as ideal transformers or gearboxes. Lastly, there are junctions that allow for connecting multiple components such that all flows sum up to 0 (0-junctions) or such that all efforts sum up to 0 (1-junctions). This methodology adds physical meaning to the analysis by considering energy flows between ports of real and arbitrary components. Such representation allows interpretable analysis and simulation of complex engineering systems. Karimian and Jahanbin [28] used the bond graphs paradigm to study a flapping bird robot. The system was divided into components which were subsequently modeled and connected using bond graphs showing the capabilities of this framework.

Furthermore, the bond graphs can be translated into a state-space representation of the system. This allows solving multi-order differential equations as a system of first-order differential equations. The general equation is provided by

$$\dot{x}(t) = Ax(t) + Bu(t)$$

where x is a vector of state variables (unknowns in the system) and u is a vector representing inputs to the system. Matrices A and B are state and input matrices respectively containing values based on the system. In the general formulation, they can be time-variant. Commonly, however, they are considered constant. State-space representation can be used to analyze and simulate the system's response.

Physical networks reduce the required abstraction even more than bond graphs. The elements in these networks represent physical components that exchange energy through ports. The number of ports corresponds to the number of ways the component can exchange energy with the rest of the system. Hence, the properties of the particular component and its interactions with the rest of the system depend on the idealization of the component's model. Similarly to bond graphs, the ports and connections between the elements represent the energy flow with the combination of *flow* and *effort* variables. The disadvantage of the physical network approach is the necessity of creating models of the components.

This is understandable for the lower abstraction of this approach and can be tackled by using a library of prepared components. On the upside, the physical networks allow us to gain insights into the workings of a singular component within a complex system.

The approach used by Karimian and Jahanbin [28] is also an example of Model-based Systems Engineering (MBSE). This paradigm focuses on the use of domain models as the main means of information exchange [29]. This is also listed as one of the requirements for multidomain simulation by Ferretti, Magnani, and Rocco [30]. It is also worth noting that the main difference from multidisciplinary analysis is creating models based on physical domains rather than expertise disciplines.

MBSE can be applied through object-oriented modeling [26]. In this framework, elemental components are modeled as classes in a library. Each of them can be instantiated and assembled into more complex systems by engineers. A number of software packages including libraries of components have been developed for the analysis of such systems (for example Dymola, OpenModelica, Simscape) [26]. Considering the availability, maturity and popularity of these tools in academia and industry, they can be considered well-suited for developing flight mechanics models.

Lastly, one should note that in MBSE frameworks geometry is not generally modeled. There can be translations from CAD models to MBSE to allow calculations of the inertia of components but these are often unidirectional [30]. For the design process, it is beneficial to couple multidomain analysis with CAD models. In the context of this project, such a connection would allow automatic evaluation of the sensitivity of flying qualities with parameters related to the geometry of the aircraft. For example, a change in the moment generated by the tail with a change in the length of the fuselage could be analyzed. The technology that allows for coupling between the various domains and the (parametric) model of the geometry is known as Knowledge-based Engineering (KBE). La Rocca and Van Tooren [31] suggest using high-level primitives (HLPs) for the conceptual design of the aircraft. Bérard, Rizzi, and Isikveren [32] focus on using high-level CAD templates (HLCts) in the CAD environment to facilitate earlier use of CFD in the aircraft design process. These works lay a foundation for coupling between CAD models and multi-domain aircraft analysis and as suggested before could provide valuable insight into flight mechanics.

2
State of the Art

While the background information on flight mechanics, aerodynamics, and multidomain system analysis is provided in chapter 1, the exact way of combining these in a single KBE framework has not yet been established. This chapter focuses on state-of-the-art research at the intersection of these areas. The broader context of this work is identified in section 2.1. In section 2.2, frameworks used to set up KBE systems are reviewed. Methodologies used to implement aerodynamic and flight mechanics models are discussed in sections 2.3 and 2.4 respectively.

2.1. Literature Overview

Firstly, the search terms were defined based on the main themes identified. These are presented in table 2.1.

Aircraft	Flight Mechanics Modeling	Aerodynamics	Multidisciplinary Analysis
aircraft	stability	aerodynamics	MDO
fixed-wing	dynamics	fluid dynamics	MDAO
airplane	simulation	CFD	object-orient*
aeroplane	mechanics	airflow	KBE
		drag	optimi*
		lift	multidiciplin*
			knowledge-based

Table 2.1: Search terms based on the main themes

To understand the context of the project, the multidisciplinary database Scopus was searched with the use of these terms. It resulted in 1309 articles and reviews which gives a good overview of the work currently done on the topics. The number of articles per year has a generally upward trend peaking at 109 articles per year in 2022 and 2023.

The relevant disciplines were identified based on the reported subject area. A pie chart of the relevant fields is presented in Figure 2.1. As one can see, this study despite being focused on flight mechanics, may be relevant to many other fields. This is to be expected as the analysis in multiple sciences becomes more and more interdisciplinary. Unsurprisingly, engineering seems to be the most important subject for this research. Furthermore, it seems to overlap also with mathematics and physics. This is likely due to the fact of inclusion of physics-based models and mathematical analysis tools. Computer science is the 4th most commonly reported subject which is not surprising considering object-oriented modeling is often based on object-oriented programming.

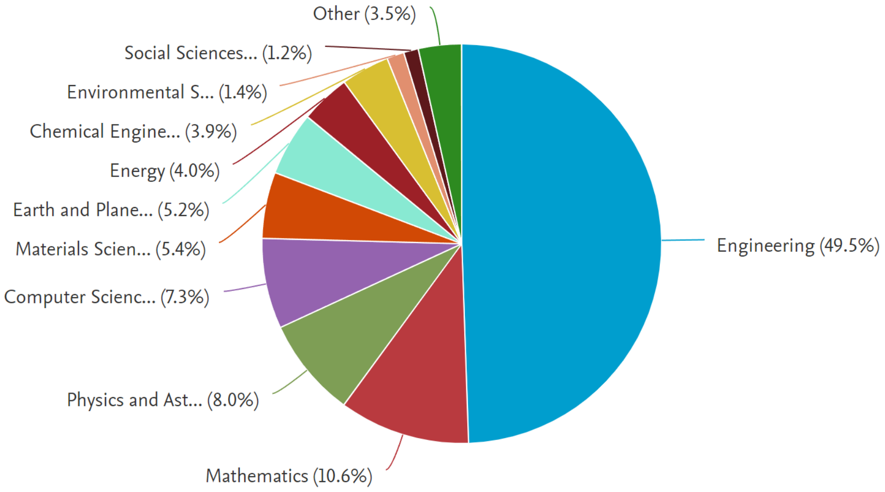


Figure 2.1: Articles found by discipline

2.2. Knowledge-based Engineering

A lot has been done to connect geometry with models of physical systems. As early as 2000, component objects combining CAD models with simulation models were introduced [33]. Other researchers focused on the connection between the disciplines by extracting information from CAD and using it as a base for the creation of models for simulation in Modelica [34]. Biahmou, Fröhlich, and Stjepandic [35] suggested *neutral data format* as a basis for the communication between CAD and analysis modules.

Bérard, Rizzi, and Isikveren [32] noticed the need for creation of CFD-compatible geometries from CAD design. The authors automate this process by the use of parametric geometry coupled with high-level CAD templates [32]. Such a process allows automated meshing of the aircraft geometry, however lacks connectivity to other analysis modules or simulations.

The foundations for the connection between geometry and the disciplinary abstractions were laid by Morris, Arendsen, LaRocca, *et al.* [36] in the MOB project. In this approach, KBE application is used to create aircraft models and feed various analysis tools with their disciplinary abstractions. The implemented Computational Design Engine (CDE) was able to employ modules provided by industry partners to perform an optimization on the Blended Wing Body aircraft. This project showcases the ability to connect geometry with various models/simulations for aircraft design.

La Rocca and Van Tooren [31] suggested using a single master model to capture the relevant knowledge and handle all the interactions between sub-domains. This led to the conception of the design engineering engine (DEE) building on knowledge from the MOB project. It is enabled by the multi-model generator (MMG) which is a KBE application providing designers with a parametric modeling environment and supporting various analysis tools with dedicated aircraft model abstractions [37]. It is based on high-level primitives (HLPs) such as fuselage, wing, engine, and connection element. Additionally, it allows for multi-fidelity capability modules. The results suggest the usefulness of DEE for aircraft and automotive conceptual design [38], [39]. The Initiator which can be part of DEE to generate initial design was also proven useful. The accuracy of the weight estimation in the tool was satisfactory [40]. Additionally, it can produce a number of feasible designs based on top-level requirements enabling direct comparison between various designs [41].

CEASIOM was another aircraft design engines in which the geometry is directly linked to analysis. It was part of the SimSAC project aiming to decrease the time and cost required with the design of new aircraft. CEASIOM was successfully employed to design transonic passenger aircraft with canard configuration [42]. The tool is centered around a geometry definition that ensures consistency across various modules. Its unified model allowed for consistent integration of various toolboxes including Tornado (described more in section 2.3). It allowed the modeling of aerodynamics and flight mechanics. The static stability aerodynamics were validated in another study but it is noted that there are discrepancies in dynamic stability derivatives [43]. From the KBE and MDO standpoint, the biggest disadvantage of CEASIOM is the unidirectionality of the framework. The geometry is fixed prior to the analysis so there's no feedback from the analysis modules that could impact the geometry directly. It's worth noting the framework is now being implemented in an open-source python-based project CEASIOMpy based on an open-standard format: Common Parametric Aircraft Configuration Schema (CPACS) [44].

What was missing in the previously developed design engines is the ability to automatically create element-based representation of the system. The behavior of the system is explored but mostly at the system level (or multiple discipline systems) rather than the component level. The automatic generation of models was proven in the automotive industry [45], [46]. It's also important for the aircraft as an accurate model offers a great insight into the behavior of the system and enables finding a consistent control system (without a requirement for tuning gains) [47].

Tian and Voskuijl [26] introduce a concept of the multiphysics information model. This model is a type of a master model that also includes simulation expertise. This allows for the automatic generation of simulation models involving multiple domains and establishing control system [26].

2.3. Aerodynamic Modeling

The lifting line theory is very useful for inviscid analysis. It is based on representing a lifting body with a vortex filament. This line is connecting aerodynamic centers (usually quarter-chord) of the wing sections. According to Helmholtz's theorems, the line extends to infinity downstream forming a horseshoe shape. To model the aerodynamics more accurately, more such lines extending to various spanwise locations can be superimposed. In particular, these can be modeled as infinity of such lines resulting in a vortex strength distribution and creating what's known as a vortex sheet. Such distribution can be used to calculate forces acting on the wing including lift and induced drag (Trefftz plane analysis was found to be particularly accurate compared to pressure integration) [14].

Another method that uses similar modeling is the Vortex Lattice Method (VLM). Instead of horseshoe vortices, the surface is split into quadrilateral filaments of constant strength. Additionally, horseshoe filaments are added to model the trailing wake. Peerlings [15] argue that this method is computationally simpler than the lifting-line model. Note that both of these methods model wings as planar (with zero thickness). Traditionally, the boundary condition enforced on both models is no normal flow through control points placed at 3/4 chord of the local section/panel. The local velocity at the control point is calculated as the sum of free-stream velocity and velocities induced by the vortex filaments (with the help of Biot-Savart law). This results in $O(n^2)$ time-complexity due to n control points being influenced by n filaments.

At this point, it's worth noting that there exist higher-fidelity methods that model the entirety of the geometry (including thickness). These, however, are more computationally expensive. Since the aim of this work is to lay a foundation in the field, these methods are out of the scope of this project. Nonetheless, the development of the framework should take into account that the computational power available to users has been consistently increasing and users may want to incorporate higher-fidelity tools in the future. A review of both methods and solvers was concluded by Peerlings [15].

One of the most popular solvers using VLM is AVL. It is particularly interesting in the context of this project as it is capable of calculating stability derivatives. The publications involving the solver are centered around smaller air vehicles (scale models, UAVs, and MAVs) [15]. AVL was successfully used as a base for the flight simulation of a 12′ airplane [48]. Lykins, Riley, Garcia, *et al.* [49] found that AVL can be successfully used to investigate dynamic modes. In particular, the natural frequency of Dutch Roll was predicted with only 5% error [49]. Furthermore, AVL proved to be comparable in the estimation of stability derivatives with ESDU and DATCOM semi-empirical methods for traditional configurations [50]. Lifting-line-based methods (in the form of AVL and XFLR5) lead to results comparable to those of higher-order CFD codes [51], [52].

Another popular vortex lattice method solver is Tornado. It's written in MATLAB and hence it is expected to be easily integrable within the Simscape environment which makes it an interesting tool to discuss in this project. The biggest difference in implementation compared to AVL is the introduction of sling vortices for the wake [15]. This allows for the trailing vortex system to be aligned with the free-stream rather than the body axis [53]. Da Ronch, Badcock, McFarlane, *et al.* [54] suggest that the Tornado can be used to study the stability characteristics based on pole plots. Yet, the authors notice the high value of coupling the method with higher order analysis [54]. The inaccuracies seem to be larger for C_{m_α} compared to C_{l_α} and C_{d_α} [55]. Lee and Visser [56] find the tool useful within expected uncertainty. Peerlings [15] citing Yerly [53] and Da Ronch, Badcock, McFarlane, *et al.* [54] note that including 2D viscous data improves the accuracy significantly at higher angles of attack. Overall, the research suggests that the vortex lattice methods can be successfully deployed for basic flight mechanics modeling but require consciousness of its limitations. Importantly for this work, Pérez Segura, Mook, and Preidikman [57] implemented an (unsteady) vortex lattice model in the object-oriented framework proving the concept for a rotor, quadcopter, flying-wing, joined-wing aircraft, and a flapping-wing insect. This suggests that such methods can be used to effectively model a variety of applications making them suitable for use in a robust KBE framework.

Another interesting set of tools in the context of this project are methods that combine 3D analysis of the geometry with 2D analysis of wing sections (2.5D models). Such models can be combined with experimental or computational analysis. The foundation of modern 2.5D analysis was laid in Van Dam, Kam, and Paris [58]. Such methods can be coupled with both experimental and computational data to include viscous effects [58]–[60]. Some tools also expand on VLM methods by including free-wake (a wake that can be convected rather than frozen in space) resulting in more realistic boundary conditions [61].

XFLR5 mentioned previously assumes independence of 2D viscous and 3D inviscid analysis. Combining the two methods results in more accurate results [51]. The code allows for the choice between a lifting-line model, VLM, and 3D-panel code. It was found that VLM is at least as accurate as higher-order 3D panel code. Yet, both methods underestimate drag [62]. The code calculates the lift coefficient with VLM (or another method of choice) and uses it to estimate the viscous drag for this coefficient [62]. Such modeling restricts the validity of the analysis to low angles of attack [51]. The tool has been successfully used for designing UAVs and MAVs [63]. More importantly for this research project, Kurukularachchi, Munasinghe, and De Silva [64] used the tool for the determination of stability derivatives.

Q3D (standing for Quasi-3D Aerodynamic Solver) combines AVL with 2D methods such as XFOIL, VGK, or MSES based on the Mach regime of the flow. Similarly to XFLR5, the solver first estimates the lift coefficient using AVL and then follows a 2D analysis procedure [65]. The analysis performed is more appropriate for transonic flows. This solver was successfully deployed in MDO applications [66], [67]. Considering the high accuracy and low computational cost of 2.5D methods, it's reasonable to assume they are a good choice for KBE flight mechanics model.

Furthermore, such methods are modular and easily modifiable making them suitable for KBE. Multi-body modeling can be used to reduce computational time significantly. Instead of modeling the entirety of geometry as one body, it can be split into multiple geometries (traditionally wing and tail) reducing the time-complexity of the calculation with negligible change in fidelity [68]. The methods can be extended to allow unsteady modeling [69], [70]. The control-point-placement can be improved for better estimation of aerodynamic forces [71]. The interactions between components can be prescribed using lifting line theory or semi-empirical methods. Using this methodology the effective angle of attack for horizontal stabilizer is calculated based on the lift generated by the main wing. This allows also for employing high fidelity models of isolated components (CFD or wind-tunnel data) to model aircraft as the whole system. The exact effects of various interactions in this approach were not quantified, especially in an aircraft-configuration-agnostic context. Additionally, in these works the interactions were only unidirectional. The main wing created a downwash on the horizontal stabilizer but the induced flow at the main wing due to the presence of the horizontal stabilizer was not considered.

2.4. Flight Mechanics Modeling

2.4.1. Mathematical description

A basic flight mechanics model was presented in section 1.1. It was based on a quasi-steady analysis developed in the early 20th century. Since then, there have been many improvements in the methodology particularly in trying to account for unsteady effects. In particular, efforts have been made to include acceleration and higher-order terms. Greenwell [25] reviews the unsteady models and notes that according to mathematical analyses there should be no higher order terms involved. The frequency-domain analysis has been criticized for its lack of applicability. This stems from the fact that the analysis of forces would require knowledge of the frequency and amplitude of the motion at each moment. This is not straightforward to model and lacks physical meaning. A significant effort has been put into modeling nonlinearities by introducing (nonlinear) dependency of static derivatives on angle of attack (and subsequently sideslip angle, direction of motion and pitch rate). Greenwell [25] concludes that Goman's nonlinear differential equation model is the most promising thanks to relative simplicity, interpretability of parameters, and modeling of a wide range of flow features.

The mentioned model employs state-space representation to model unsteady aerodynamics [72]. Goman and Khrabrov [72] notice the similarity between this model and the indicial response. The work introduces the flow state (such as separation point) as the dynamical variables of the system. This captures the prehistory of the state and is able to replicate the results of a wing in forced oscillation. The research suggests that with this representation the stability analysis and dynamic simulation can be done with eigenvalue analysis or time-integration [72].

2.4.2. Estimation of Stability Derivatives

Schmidt and Newman [73] developed a methodology for the estimation of dynamic stability derivatives with the help of RANS and URANS. The results are in accordance with the experimental results for low angles of attack [73]. A similar methodology is also applied to more complex configurations suggesting that steady aerodynamics can be used up to 10° [74]–[76]. Ronch, Vallespin, Ghoreyshi, *et al.* [77] found that the results of such analysis are highly dependent on geometry. In general, they are reliable at low angles of attack but are not able to model large-amplitude motions accurately [77]. Mader and Martins [78] used RANS and Euler-based methods to not only estimate the stability derivatives but also employ the adjoint method to perform stability-constrained optimization. Lykins, Riley, Garcia, *et al.* [49] and Rose, Yaralian, Wagster, *et al.* [48] successfully used AVL for the estimation of stability and control derivatives of two scaled models validating the results with flight tests. Furthermore, the research also presents the methodology of estimating parameters from a flight test [48], [49]. Recent research suggests that VLM-based methods provide good estimates for stability derivatives [79]. Semi-empirical methods (DATCOM) were also employed successfully to simulate the flight of Cessna 172 [80]. Overall, the estimation of stability and control derivatives was found to support flight tests and simulation model development [81].

2.4.3. Toolboxes

Various attempts have been made to develop toolboxes capable of modeling, simulating, and analyzing flight dynamics. Looye [82] describes Modelica-based flight dynamics library. It allows for multi-disciplinary modeling for use in aircraft and control system design. It is capable of automatic generation of simulation, inverse model, trimming, and linearization codes. The tool is validated on a couple of test cases including the development of an automatic landing system and flexible aircraft flight simulator. The biggest shortcoming of this software is the lack of captured knowledge thus requiring a number of user-specified definitions of physical processes [82].

Voskuijl, Klerk, and Ginneken [47] investigate the flight mechanics model of the PrandtlPlane. In particular interest of this work is the flight mechanics toolbox. It enables the integration of submodels from various disciplines and automated analysis. It is argued that the toolbox can be deployed in MDO framework. In this work, the aerodynamics is modeled with the help of VSAERO at various conditions. Stability and control derivatives are obtained in the same way. Overall, the flight mechanics toolbox was proven to generate flight mechanics models automatically making it suitable for the design environment [47]. Heimans [83] conducted a sensitivity analysis for engine integration building on the experience in this project with use of FlightStream and VSAERO.

The toolbox from Voskuijl, Klerk, and Ginneken [47] was further developed into Simscape-based Performance, Handling Qualities and Loads Analysis Toolbox (PHALANX) and employed to establish a configuration-agnostic methodology to size control surfaces [84] and model configurations with coupled aero-propulsive performance [85], [86]. There are a couple of limitations of this toolbox. It focuses on an isolated aircraft with manually prescribed interactions between the aircraft and a fixed environment without any possibility of changing parameters and models of atmosphere and gravity. In PHALANX, the airframe is treated as a single rigid body which makes it impossible to model aerodynamic lags in case of maneuvering or gusts. Additionally, such treatment makes it impossible for the user to define aerodynamic interactions between various components. This is the biggest disadvantage in the scope of this project. PHALANX does not allow automating modeling of the whole aircraft configuration based on the data prescribed at the component level which may be beneficial as outlined in chapter 1.

3 Methodology

As briefly introduced in chapter 1, the scope of this project is to lay a foundation for aircraft analysis using KBE technology based on data prescribed at the component level working towards an automatically generated and aircraft-configuration-agnostic flight mechanics model. In particular, the literature study suggests that there is little research done on the intersection of KBE, aerodynamic modeling, and flight mechanics modeling. MMG combined with PHALANX seems to be the most closely related to this project. Nonetheless, there are significant shortcomings limiting the ability to support multi-body analysis. There were projects that included aerodynamic data at the component level, but the aerodynamic interactions were based on major assumptions about geometry and their effect was not quantified.

A new project called FLAPERON (FLight mechanics Analysis and PERformance OptimizatioN) was established to allow modeling of complex dynamic systems and, in particular, facilitate the analysis of aircraft flight characteristics. The main advantage of this toolbox compared to PHALANX within the scope of this project is the goal of FLAPERON to support for many, moving, and interacting components. This toolbox is developed using object-oriented programming in MATLAB environment and aims to allow conversion to Simscape simulations. The latter is a tool developed by MathWorks that allows the creation and simulation of physical networks as discussed in section 1.3. It includes a library of prepared components across multiple domains. This allows us to build on top of these components when creating flight mechanics simulations. Additionally, Simscape is compatible with Simulink which allows the addition of non-physical components such as control logic to the system.

In this work, I build on the experience of MMG and PHALANX and contribute to the FLAPERON toolbox by modeling components, sub-systems, systems, and interactions in an aircraft-configuration-agnostic manner. This approach allows for flexible, adaptable, and cross-disciplinary applications without being constrained by specific models, systems, or assumptions. This investigation lays a foundation for automatic aircraft-configuration-agnostic estimation of aircraft performance based on data prescribed at the component level.

In particular, the work aims to answer the following research question:

What is the impact of including lifting-line-based aerodynamic interactions on the simulated aircraft response when considering aerodynamic data only at the component level?

To address this research question, further development of the FLAPERON framework was necessary. Specifically, the framework needed to evolve to support full aircraft geometries, albeit in simplified form. Additionally, the simulation using physical networks had to be validated within an aircraft-configuration-agnostic context. Finally, it was essential to establish a foundation for connecting and translating between these two approaches.

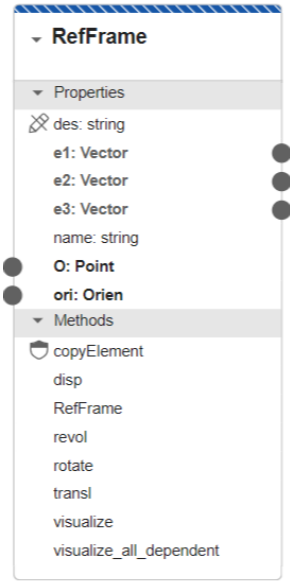
In this chapter, the details of the model are presented and the methodology allowing for answering the overarching research question is discussed. In section 3.1, an emphasis is placed on the code's architecture with the reasoning behind the related design choices. In section 3.2, the methodology for setting up the physical network is discussed. In section 3.3, the methods for ensuring that the software is behaving as expected are discussed including basic test cases. In section 3.4, the methods for assessing the accuracy of the 2.5D aerodynamic modeling will be discussed.

3.1. Model

3.1.1. Geometry

State of the art

As discussed in previous chapters, the inclusion of the geometrical model is one of the important parts of the KBE approach. Each of the objects needs to have a shape and be positioned and oriented in space. This requires the definition of reference frames. In FLAPERON, the reference frames are defined with a point (Origin of the new reference frame) and orientation. Note that the point and the orientation themselves make sense only if they are defined with respect to another reference frame. This is the reason why in FLAPERON there exists a master reference frame - `World`. There exists only one instance of `World` and all of the other reference frames are defined with respect to it. It is also possible to define the reference frame with respect to a non-`World` reference frame which results in a dependency tree that always has `World` on the top. The class diagram of the reference frame is presented in Figure 3.1.



RefFrame

Properties

des: string

e1: Vector

e2: Vector

e3: Vector

name: string

O: Point

ori: Orien

Methods

copyElement

disp

RefFrame

revol

rotate

transl

visualize

visualize_all_dependent

Figure 3.1: Class diagram for Reference Frame

Both the orientation and the origin need to be defined with respect to a reference frame and offer the capability of being expressed in another reference frame. Hence, the choice of defining an abstract superclass that would encapsulate this functionality. In the case of FLAPERON, this class is called `GeomObject` and all geometrical objects inherit from it. This presents an advantage of object-oriented architecture where code rarely needs to be duplicated even if functionalities are shared between multiple objects. The class is designed for flexibility in representing geometric objects within different reference frames, allowing for complex spatial relationships and transformations. The methods are focused on recalculating and expressing the object's properties (like position, orientation, etc.) relative to different reference frames, which is crucial for simulations or modeling tasks that involve dependent, moving, or rotating objects. Other geometric entities that have been implemented include `Vector`, `Tensor`, and `Wrench`. The class diagram for these objects is presented in Figure 3.2.

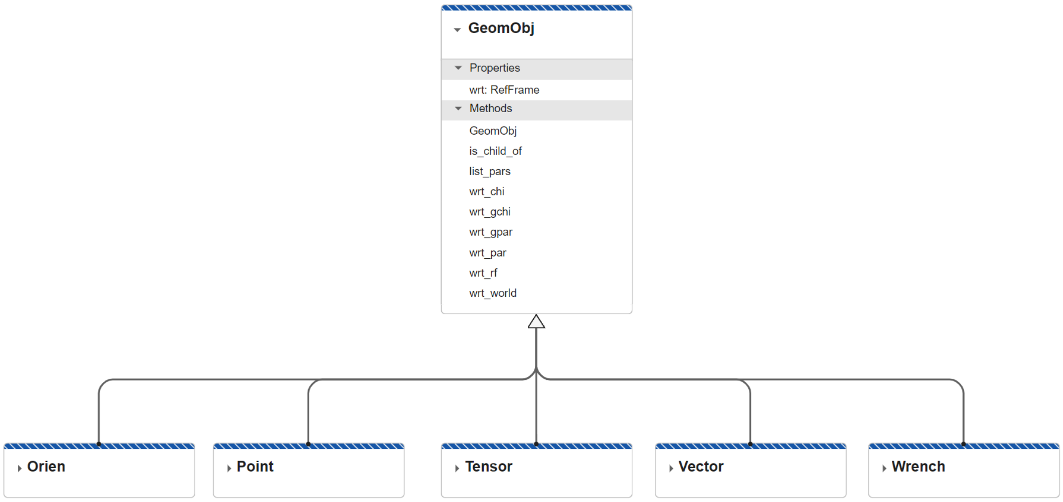


Figure 3.2: Class diagram for Geometric Objects

The `Vector` class aims to represent a vector in a 3D space in a given reference frame. It is characterized by magnitude and direction. While it does not have an application point, it can be considered to be applied at the origin of the reference frame it is defined in.

The `Tensor` class represents a second-order cartesian tensor in a given reference frame. It is expressed as a 3x3 matrix.

A `Wrench` class represents, in accordance with Screw Theory, forces and moments acting on a particular component. It is a 6 element array made of two vectors expressed in the same reference frame. The first vector describes the resultant force acting on the component. It is only responsible for changing the translational momentum of the component. On the contrary, the second vector represents a pure resultant couple/moment related to the change in the rotational momentum of the component.

Original Contribution

To ease the development of FLAPERON, an additional functionality of visualizing reference frames was added. This is a common capability of KBE languages that was not previously present in the toolbox. The newly implemented method allows for plotting on the most recent figure major axes of the reference frame with respect to the `World` frame by traversing the tree of reference frame dependencies. Additionally, an additional method was derived which visualizes all of the reference frames in the dependency tree. This enables easier verification and debugging while placing components. An example of such visualization is presented in Figure 3.4.

3.1.2. Abstract Components

State of the art

Considering the goal is to model a system composed of smaller parts, it is essential to implement components capable of interacting with each other. Components can be grouped into classes with common properties. In particular, some of the classes are abstract and are there to provide common properties for inheriting classes but are not aimed to be instantiated on their own. FLAPERON framework includes a main `Component` superclass which enforces each component to have a name and be expressed with respect to a reference frame. Furthermore, methods allow for copying components, defining auxiliary reference frames, plotting geometry, and prescribing translation into a Simscape environment. The last feature is important for prescribing interactions between components for the simulations and in particular, the most basic, rigid connection. These features are common for all components but they can be overridden if needed.

The groups of components can also be grouped according to the common properties. In FLAPERON, the components are divided into aerodynamic components (`AeroComponent`), elastic components (`ElasComponent`), and components with inertia (`InerComponent`). Out of these classes, the components have been conceptualized. In particular, the current development includes classes for airfoil, wing, blade, spinner, and propeller. Note that MATLAB supports inheritance from multiple superclasses which means the component can be at the same time inertial and aerodynamic component. The related class diagram is presented in Figure 3.3. The non-abstract classes are described in subsection 3.1.3.

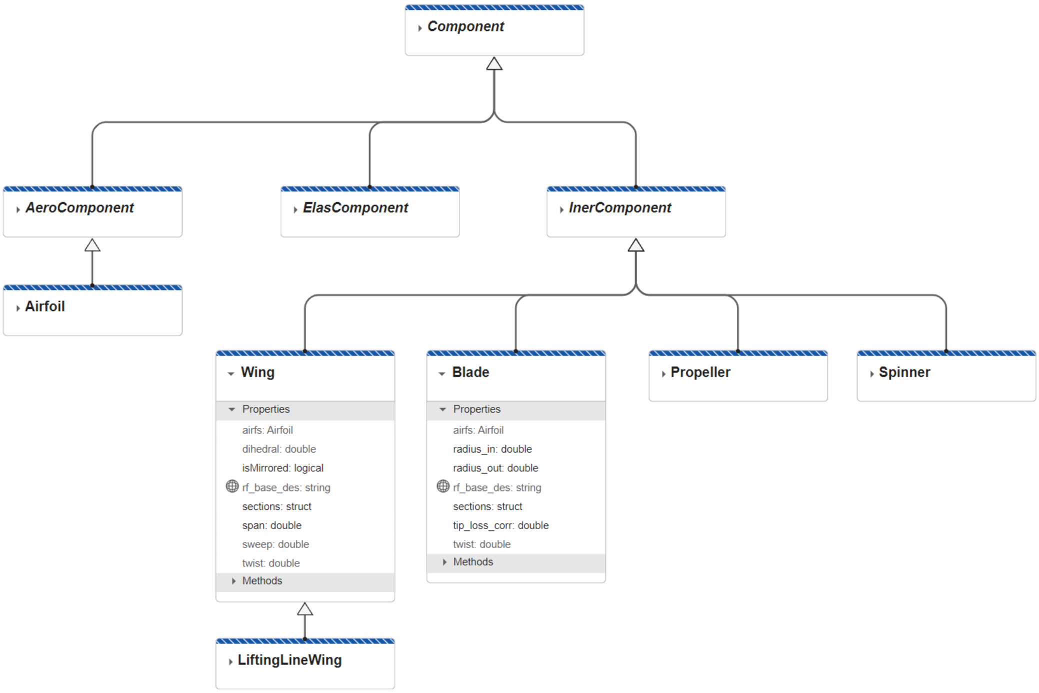


Figure 3.3: Class diagram for Components

`AeroComponent` class is an extension to the `Component` that includes aerodynamic properties. Thanks to this extension all inheriting classes are expected to contribute to the net force acting on the aircraft. This class does not include any information about elasticity, mass, or inertia. Hence, it can be combined as a part of the class with the appropriate properties which would also define how to interpret the aerodynamic forces.

`ElasComponent` is a class envisioned to model elastic properties in a multi-body system. Hence, this class contributes to the internal forces in the system. At this point, this class is not implemented and acts as a placeholder to guide the future development of FLAPERON.

The `InerComponent` class, derived from a `Component` base class, encapsulates the concept of a mechanical component with mass and inertia. It plays a crucial role in the dynamics of mechanical systems. It introduces properties such as mass (`mass`), center of gravity (`cg`), inertia tensor (`I`), and inertial reference frame (`iner`), which are fundamental to modeling the physical behavior of components in motion. This class is instrumental for simulations and analyses involving the dynamics of complex mechanical assemblies, leveraging principles such as the parallel axis theorem for accurate inertia calculations.

3.1.3. Components

State of the art

The `Airfoil` class encapsulates properties crucial for defining the airfoil's geometric and aerodynamic characteristics. It inherits from the `AeroComponent` class, indicating its role in aerodynamic computations. Dependent properties, computed on-demand, include chord, thickness, relative thickness, and quarter chord. Currently, the `Airfoil` class constitutes a significant building block for `Blade` and `Wing` classes described below.

Since FLAPERON is expected to be a flexible framework, it was envisioned to also be used for the analysis of propellers. This is out of scope of this project and hence will not be described in detail. It is worth noting, however, the level of complexity one could achieve by combining various components with different reference frames. The propeller is composed of a spinner (hub) and blades positioned radially (see Figure 3.4). It is itself a component with inertia which is composed of other inertial elements. Additionally, each of the blades is composed of airfoils that are connected to an aerodynamic model.

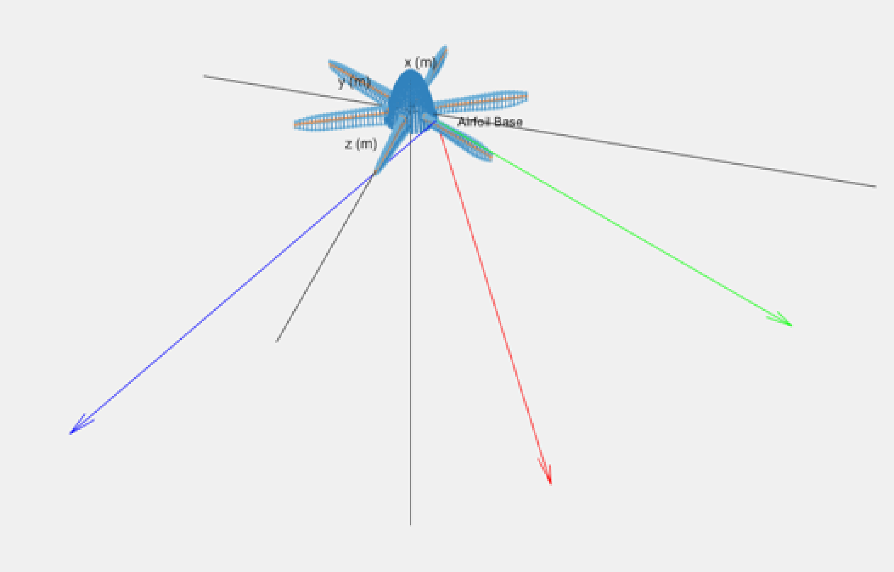


Figure 3.4: Propeller with visualized reference frame of one of the airfoils

Original contribution

The Wing class and related functionality within the FLAPERON framework was developed. The Wing class is strongly based on the Blade class considering the physical similarity between the objects. Both classes are modeled as components with inertia including airfoils as the main source of the aerodynamic data. Airfoil (2D) data cannot be directly translated to forces at the wing level. In the 3D context of a wing, each of the sections imparts induced velocity on the flowfield. This changes the inflow conditions and hence the aerodynamic forces exerted on the section. This phenomenon is described in more detail in subsection 3.1.4.

The main developments added to the Wing class are the change of the base reference frame to coincide with industry standards. Origin is defined at the leading edge of the root airfoil, with X-axis pointing towards the trailing edge and Z-axis points from the pressure side to the suction side of the wing. It may be beneficial for other applications to consider Y-axis in the spanwise direction instead. Twist, sweep, and dihedral definitions were added following the standard convention. It is worth noting that sweep and dihedral are applied by shearing the airfoils rather than rigid rotation of the wing. Despite being defined where the airfoils are specified, these parameters make sense only in the whole wing/blade context. These properties are combined in a sections structure. They are calculated based on simple trigonometric relations based on the relative positions of each section with respect to the most inboard airfoil as shown in Listing A.1. The set functionality has also been implemented and includes the inverse of the functions.

Additionally, the Wing incorporates a boolean property isMirrored that allows creation a symmetrical wing instead of a half-wing. The capability of creating a half-wing has been maintained to allow analysis of asymmetric aircraft. This is important to maintain to allow the crucial flexibility of the KBE approach. Additionally, this might allow analysis of aircraft in case of an unexpected configuration change (failure). The capability is maintained across the whole class but the two most prominent examples are setting of the airfoils and their relative positions. This is presented in Listing A.2. Setting airfoils is done by concatenating the passed list of airfoils with its reversed copy. The airfoils are placed traditionally for half-wing, and then the sign of the location is changed with respect to the symmetry plane for the other half. An alternative approach could involve creating concatenated lists with reversed copies (as for airfoils) for distance from the symmetry plane, twist, sweep, and dihedral. The current approach, despite being less human-readable, offers the computational benefit of lower memory requirements. Additionally, the code allows setting the axis of rotation for airfoil twist. Current code supports the rotation around leading edge but it can be tailored to use quarter chord (or any other chord fraction) instead with a simple change of parameter.

3.1.4. Interactions

Apart from modeling components, it is necessary to model interactions between them if one is to understand the behavior of the whole system. In a rigid-body aircraft, the two main interaction pathways are rigid connections/transformations and aerodynamic interactions. Rigid transformations are fully modeled by Simscape library treated in detail in section 3.2. Aerodynamic interactions can be perceived as perturbations to the steady-state flow field which changes the effective inflow conditions. In the case of a flying aircraft, the unperturbed flow field would be a uniform flow with a velocity opposite to the aircraft's velocity.

Various methods can be implemented to model the flowfield and hence perturbed velocity. In this work, the interactions are governed by vortex filaments as argued in sections 1.2 and 2.3.

Fundamental theorems

Vortex filament modeling requires consideration of fundamental theorems developed by Helmholtz and Kelvin [87]:

- Kelvin's Theorem: The circulation around a closed curve is invariant with time.
- Helmholtz's First Theorem: The strength of a vortex is constant along the vortex filament.
- Helmholtz's Second Theorem: A vortex filament cannot end in a fluid.
- Helmholtz's Third Theorem: In the absence of external forces, an irrotational fluid element remains irrotational.

Additionally, for the calculation of lift at a given section, Kutta-Joukowski theorem is relevant. This theorem states that the lift is directly proportional to the density, velocity of the free-stream and the circulation along the path enclosing the aerodynamic body. In particular, the latter can be represented by the integral of vortex strength across a filament.

Assumptions

It is also worth noting the assumptions behind this aerodynamic model.

- The viscous effect far from the airfoil can be neglected allowing the interactions to be modeled by vortex filaments. This disregards viscous effects which could lead to the overestimation of calculated induced velocities close to the singularity. This can be developed further to include a viscous vortex core to increase the accuracy of the solution [88].
- The flow is quasi-steady. The time-history is neglected and the lift buildup is assumed to occur immediately (without aerodynamic lag as described in section 2.3).

Implementation

Two approaches for modeling the interactions have been envisioned for this work. Firstly, the interactions can be modeled directly in simulation. The details of this modeling are discussed in section 3.2. Alternatively, the aerodynamic data can be generated prior to the simulation of the model at various attitudes and the simulation can interpolate between these states to extract forces/moments acting on the aircraft as described below. In principle, any aerodynamic solver can be used. In this work, a foundation for a vortex lattice method utilizing FLAPERON objects was laid. This was done to implement a highly versatile and modifiable solver directly compatible with the toolbox.

The velocity induced by a vortex filament is calculated according to the Biot-Savart law. Following this approach, each infinitesimal part of the filament $d\vec{l}$ induces velocity $d\vec{v}$.

$$d\vec{v} = \frac{1}{4\pi}\Gamma\frac{d\vec{l}\times\vec{r}}{|\vec{r}|^3}$$

where $\vec{r}$ is a vector from the filament to the point where one attempts to calculate velocity. Therefore, by (numerical) integration over the length of a filament, it is possible to obtain the induced velocity due to the presence of the filament. If a filament is composed of straight segments it is possible to combine analytical results [88] which is much cheaper computationally. Furthermore, following the linearized potential theory, one can sum up the contributions of various filaments.

The filaments are implemented as an extension to the `Wing` class. The number of vortex filaments within a wing is equal to the number of airfoils defined for this wing. Within the wing, each filament runs through a midpoint connecting quarter chords of successive airfoils. At the tip of the wing, the filaments run simply through the airfoil defining the edge of the wing. This is an implementation of a Vortex Step Method [58]. From the quarter chord points, vortex filaments follow straight lines in a chordwise direction towards the trailing edge and extend a user-defined number of local chords downstream (default 20). It is worth noting here that the code is modular and can be adjusted to any methodology of describing wake including free wake. According to Helmholtz theorems the vortex strength is constant along a single vortex filament and extends to infinity (or form closed loops). Despite this, considering the high computational cost of numerical integration and $d\vec{v}$ dropping with the square of the distance, the vortex filaments are modeled as finite. This choice can be validated with a convergence study. If the implementation is adjusted to use only straight segments, it would be possible to adjust the formulation to use analytical result for straight segments and include semi-infinite filaments increasing the validity of the method. The current implementation of vortex filaments is visualized in Figures 3.5 and 3.6.

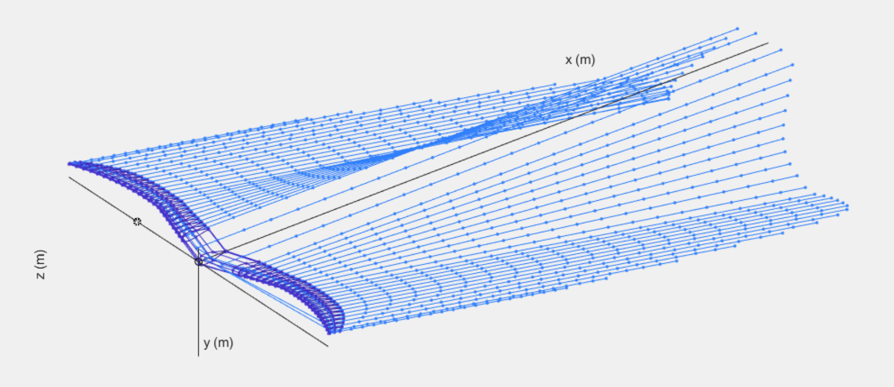


Figure 3.5: Wing with visualized vortex filaments

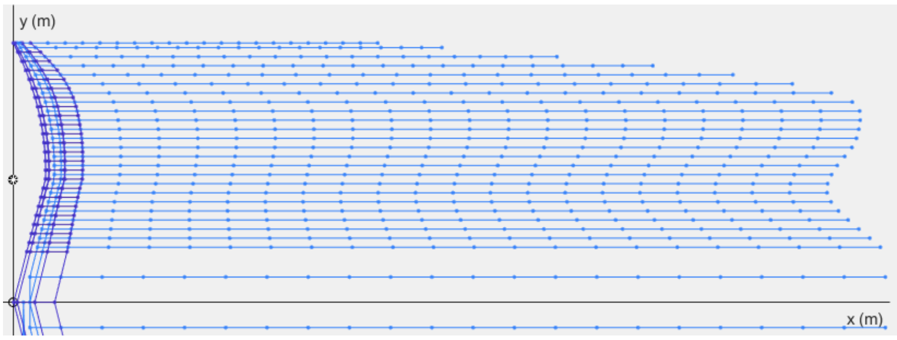


Figure 3.6: Top view of the wing with visualized vortex filaments

The analysis above assumed that the vortex strength of each filament is known a priori. This is not true in general. Yet, it can be calculated by solving a system of linear equations enforcing boundary conditions. In the case of lifting line method, the boundary conditions assume tangential flow at the midpoint between 3/4 chord of the successive airfoils. This is enforced in the model by setting the dot product between the normal vector and the induced velocity equal to the negative of the dot product between the normal vector and the unperturbed velocity.

$$\vec{N} \cdot \vec{v_{\text{ind}}} = -\vec{N} \cdot \vec{v_\infty}$$

The normal vector is a dependent property of the lifting line wing as presented in Listing A.3. It calculates the unit vector that is normal to the surface at the 3/4 chord point. It is based on the cross-product of the vectors connecting interchanging points at the airfoil's midpoint and at the airfoil's trailing edge.

The `lifting line wing` class also has a method for calculating the induced velocity by each filament at a given point per vortex strength as shown in Listing A.4. This is done by numerical integration of the induced velocity by a given vortex filament assuming the vortex strength of unity.

This is employed by the `lifting line solver` to construct a coefficient matrix. This matrix can be used to calculate the perturbed velocity given the vortex strengths.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{N1} & \cdots & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_N \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}$$

where a_ij is a coefficient representing induced velocity per vortex strength due to j-th vortex filament at i-th control point, Γ_j is vortex strength of j-th vortex filament, v_i is the component of the induced velocity parallel to the normal vector. The `lifting line solver` class takes as input a list of the `lifting line wings` and the aircraft's unperturbed velocity. It then uses the built-in methods to calculate the coefficient matrix connecting all of the vortex filaments and control points as presented in Listing A.5. The vortex strengths Γ are calculated by inverting the above matrix as implemented in Listing A.6. Knowing the vortex strength and its distribution on the wing, one can calculate relevant performance indicators for aircraft such as lift and induced drag. This is an example of a standard lifting line theory.

Possible extension

The class can be easily extended to include also a nonlinear vortex step method (also called Weissinger method). Thanks to this method, nonlinear aerodynamic airfoil data can be included. The data can be obtained through wind tunnel experiments or with a 2D viscous analysis such as XFOIL. Since this method is non-linear it depends on convergence. Regardless of the exact implementation, the lift is calculated based on the angle of attack which is used to calculate vortex strength which in turn impacts the angle of attack. This leads to a more generalized boundary condition:

$$|\vec{V}_\infty \times \vec{\Gamma}| = \frac{1}{2}|(\vec{V}_\infty + \vec{V}_{ind}) \times \vec{z}|^2 c C_L(\alpha)$$

where $\vec{V}_\infty$ is the free-stream velocity, $\vec{V}_{ind}$ is induced velocity, $\vec{z}$ is a unit vector normal to the airfoils plane, and c is the local chord. Matlab's *fmincon* function can be used to achieve the convergence. Traditionally, the velocities are calculated at the quarter-chord point. There have been other control points proposed in literature [88]. Yet, they require more complex calculations without a significant increase in the accuracy of the results.

The total drag can be calculated based on the sum of profile drag and induced drag. The profile drag can be calculated from the airfoil's drag coefficient and extending it to be valid for the whole section. In similar software, the Trefftz plane analysis has been proven very successful for calculating induced drag. Hence, this is the method of choice that could be implemented for the extension of this project.

Another algorithm that could be easily implemented using the framework would be an adjustment of the wake based on the local velocity. This would allow shifting from a frozen-wake approach to an evolving wake (similar to the algorithm implemented in VSPAero in which wake is adjusted based on the perturbed velocity) or even free wake. Additionally, one could decouple analyses of various lifting surfaces allowing in reduction of computational cost as discussed by Soikkeli, Matko, and Koopman [68].

Deployment in simulation

Implementing these algorithms in-house offers a number of benefits over using already existing solutions such as AVL. Firstly, the object-oriented approach and the modularity of the code allows easy implementation of new methods making use of tools available in FLAPERON. This makes it easy to capture new knowledge which is aimed to facilitate the Knowledge-based Engineering approach.

Alternatively, one could delegate the creation of the aerodynamic database to an external solver (such as DATCOM, AVL, or more complex CFD codes) by creating API to exchange the information about the geometry, simulation, and results between the tools.

The model for calculation of vortex strength distribution and, hence, lift distribution has been presented above. The lift distribution across different lifting surfaces can be easily translated to the forces and moments acting on the aircraft at a range of attitudes.

Thus, one can attempt to find a trimmed condition for the aircraft - a state in which there are no net forces and moments acting on the aircraft. This is the equilibrium state from which response to perturbations can be measured in order to estimate stability characteristics. This can be established by calculating the polar of C_m and finding the x-intercept of that polar. Since all the designs analyzed in this work are symmetric, there's no need to vary β and γ to find the trimmed state. The balance between weight and lift can be found by varying the speed or weight of the aircraft depending on which property is prescribed. Since modeling the propulsion system is out of the scope of this work, the thrust will be matched to the calculated drag. This methodology allows for fast and easy calculation of the trimmed state ahead of simulation. It can be extended for more complex configurations which may require yaw and roll to maintain a steady flight. Yet, as this work aims to lay a foundation for KBE Flight Mechanics Modeling this is considered out of scope of this project.

From the trimmed conditions one can evaluate the behavior if the system (aircraft) is perturbed from the equilibrium. This allows for a calculation of stability derivatives ahead of flight simulation runtime and using them for modeling flight. Note that estimation of forces, finding steady-state, and linearization of the model can also be delegated to the simulation as discussed in section 3.2.

3.2. Simulation

The simulation of the aircraft is set up within Simulink environment with the help of SimScape Multibody library. This library allows for the creation of simulations based on physical systems. The method is based on physical networks as described in section 1.3. This environment was chosen as it is well-proven, contains an extensive library of parts and is highly connected to MATLAB - the language in which FLAPERON is developed. Additionally, it supports script-based creation of the models, allowing physical network elements to be placed and connected automatically. As mentioned previously, the aircraft is modeled as a rigid body.

It is worth noting that the goal of the FLAPERON framework is for the flight mechanics model to be created automatically. For this work, blocks are connected manually with the recommendation for the automatic generation provided in Appendix D.

3.2.1. Environment

For the majority of the work, the density of the air is assumed constant with the value of 1.225 $\mathrm{kg/m^3}$. Thanks to the modularity of the physical network, it can be easily swapped for the ISA model if needed as presented in Figure 3.7. The gravitational acceleration is assumed constant for this project. In reality, the gravitational acceleration varies around the globe by about 0.7%. The reason for keeping these values constant is to reduce the number of variables in the model to allow for easier comparisons with other simulations. These variables are not expected to impact the answer to the research question as the changes in gravitational acceleration or ambient air density between different aircraft components are negligible. The ability to change these models, however, highlight the adaptability of the model.

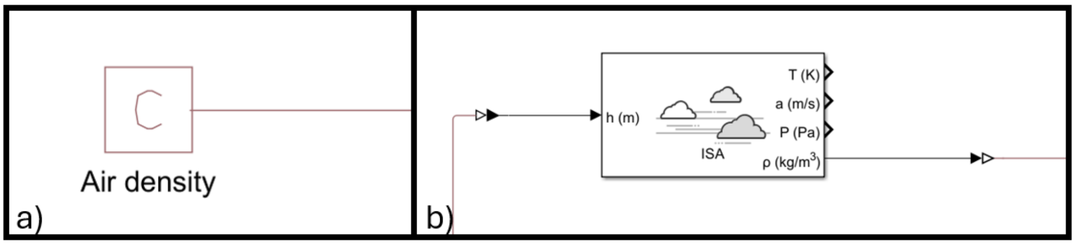


Figure 3.7: Air density model: a) constant b) ISA atmosphere

3.2.2. Inertial objects

Adding inertial objects to the simulation can be done in two ways. Firstly, mass and moments of inertia can be calculated within FLAPERON and implemented by adding a `General Variable Mass` to the simulation. This can be done by creating structural models or using semi-empirical relations to extract the inertia tensor for (components of) the aircraft. Alternatively, a simple model can be constructed from primitives available in Simscape Multibody library. Available body primitives are presented in Figure 3.8. In this work `Brick Solid` primitive is commonly used. This is because lifting surfaces as well as fuselage can be modeled as `Brick Solids` with appropriate dimensions.

The bodies can be connected using rigid transformations to form more complex configurations and Simscape can calculate the total mass and inertia matrix by itself. It is also worth noting that the connection instead of being rigid may be composed of deformable elements (or mix of the two) as seen in Figure 3.9.

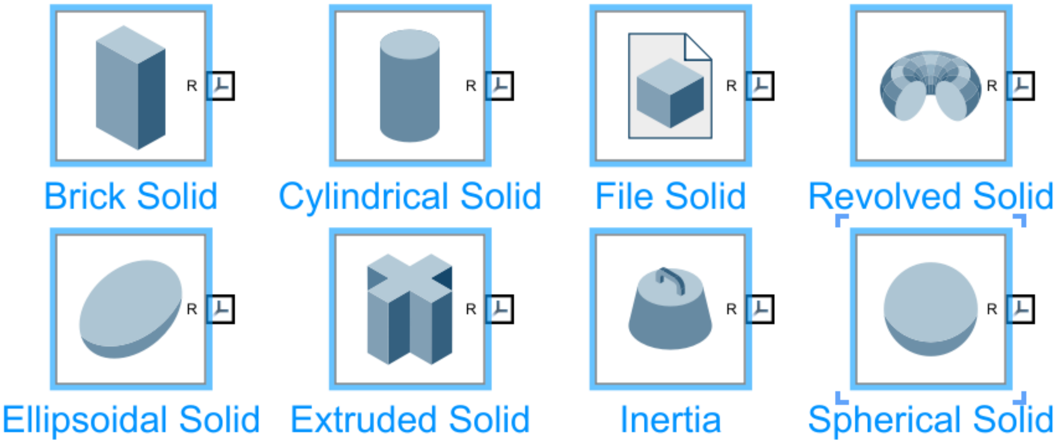


Figure 3.8: Solid models

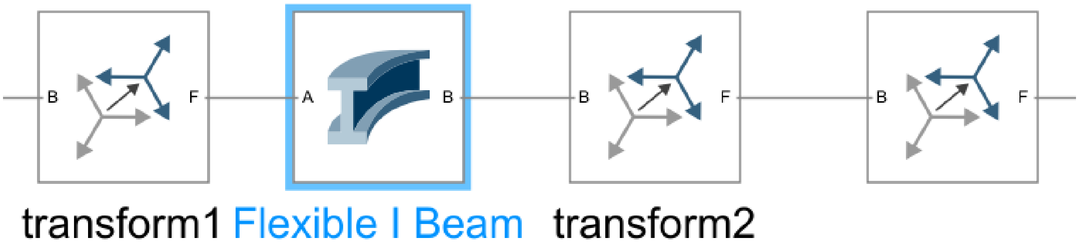


Figure 3.9: Integrated flexible I beam

3.2.3. Motion

In the case of modeling flight, it is important to discuss also the way the motion is modeled. In this work, the motion is measured with respect to the `World` reference frame. For this project air is assumed stationary with respect to the `World`. The two frames can easily be decoupled with prescribed motion between them but this falls out of scope for this project. The relative movement of the frames can be prescribed, measured, and constrained using joints. In particular, 6 degrees of freedom `Bushing Joint` was used to prescribe the aircraft's initial position and velocity as state targets. It is also used to provide information about aircraft motion (including altitude used for ISA atmosphere model). If the motion of a reference frame connected to the aircraft needs to be measured, `Transform Sensor` is employed. The `Transform Sensor` senses the spatial relationship between two connected reference frames (base and follower). The measurement can be taken in any of the two reference frames or the `World` reference frame. The sensor allows measuring translational and rotational positions, velocities, and accelerations. Measuring translational velocity is particularly useful for measuring local airspeed and angle of attack for various surfaces as described in subsection 3.2.4.

3.2.4. Forces

For any non-trivial dynamics in the simulation, the forces must be modeled. Traditionally, the forces and moments were non-dimensionalized in the form of aerodynamic coefficients and/or stability coefficients. These describe non-dimensionalized forces/moments and their derivatives with respect to other state variables. These are usually stored in the form of a pre-generated data table based on the angle of attack. Additionally, the lookup for such databases could also be based on other variables such as Reynolds number. This functionality have been implemented in the simulation with Simscape's `1-D Lookup Table`. This is presented in Figure 3.10 for lift and moment coefficients and their derivatives with respect to the rate of change of angle of attack.

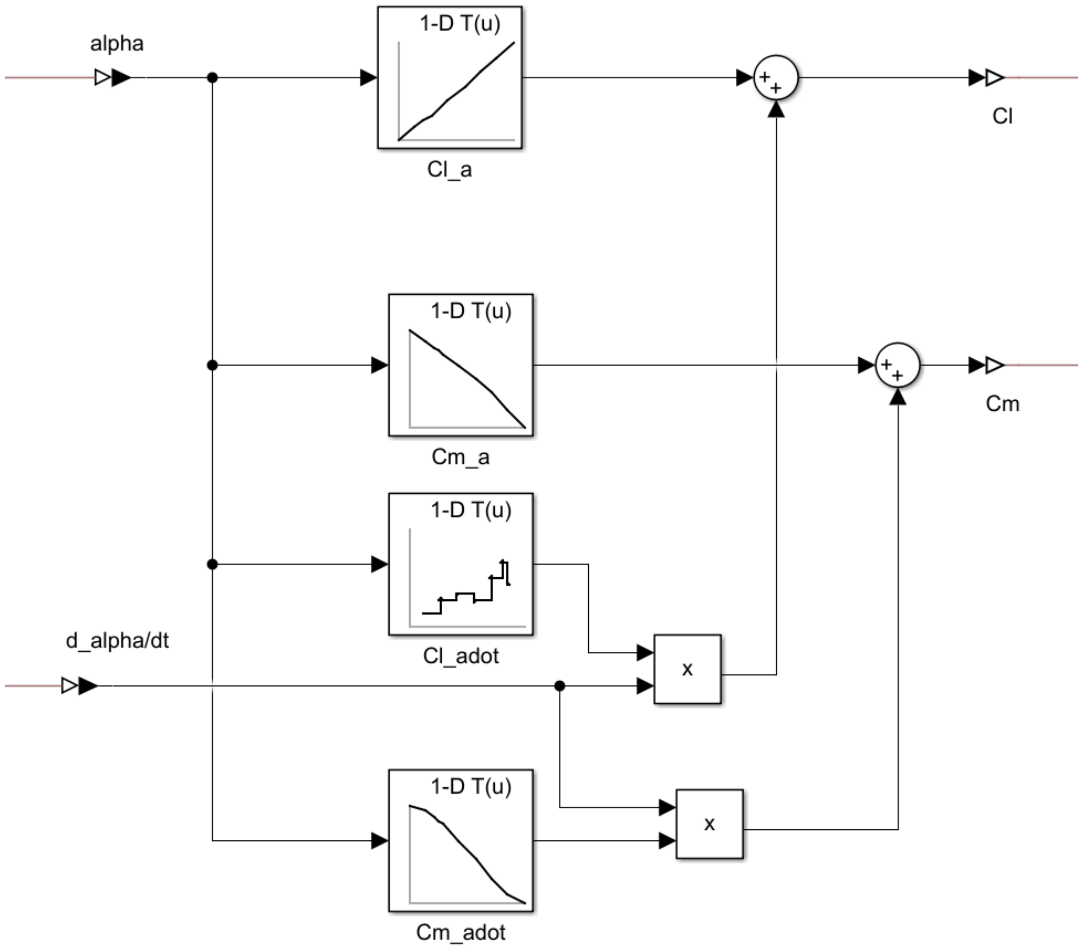


Figure 3.10: Lookup tables for aerodynamic coefficients

The traditional aerodynamic lookup tables for the whole aircraft, while being vastly proven in the industry and allowing for capturing knowledge about forces acting on the aircraft, lack interpretability and physical meaning. To get the full benefit of the physical networks and get better insight into the aircraft as a system, it is worth prescribing forces on a component level. This also allows for modeling physical phenomena directly rather than relying on external tools that could have questionable validity for novel configurations.

This can be done with lookup tables as previously on per component basis. However, since one of the benefits of the framework is modularity it has been decided to use a different approach. The lift coefficient is modeled as a linear function with angle of attack (constant C_{L_α}). The drag coefficient is calculated based on parabolic drag polar as shown in Figure 3.11.

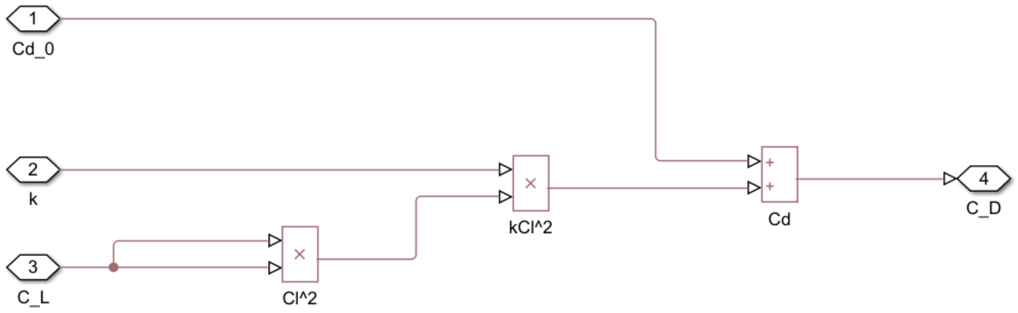


Figure 3.11: Implementation of parabolic drag

In both cases, it is necessary to convert the coefficients into forces acting at appropriate bodies. If the coefficients are prescribed at the aircraft level, the force and moments are applied at the center of gravity. For application on the component level, the application point of forces is attached to the main reference frame of the related component. In both cases, this is done by employing `External Force and Torque` block. This element is connected to a link with the information about the magnitude and direction of the force and to the reference frame at whose origin it is applied. This is also used for the implementation of the constant propulsion force. The latter is shown as an example in Figure 3.12.

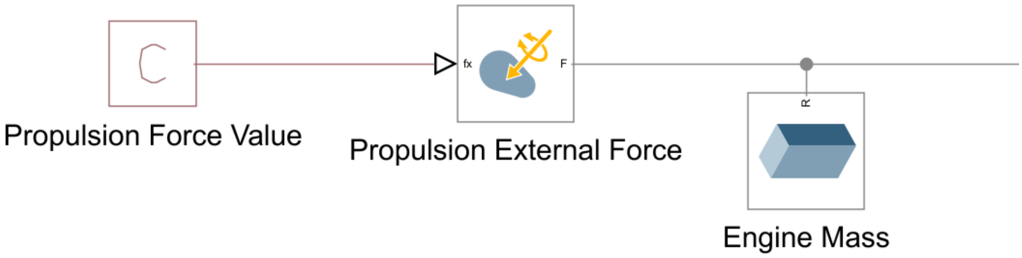


Figure 3.12: Example of implementation of force

The aerodynamic forces are proportional to the square of airspeed and their direction is also related to the airspeed. Drag is in the same direction as local airspeed while lift is perpendicular to the airspeed and spanwise direction (quarter-chord line). Hence, measuring airspeed is of the essence. It is sensed using `Transform Sensor` between air/world reference frame and the body frame and resolved in the reference frame of the body. The signal of the drag force is generated by multiplying the velocity vector with its norm, the density of air halved, the reference area, and the drag coefficient.

$$\vec{D} = \frac{1}{2}\rho S_{\text{ref}} C_D |\vec{v}|\vec{v}$$

Lift is calculated similarly with the swap in the coefficient and different direction.

$$\vec{L} = \frac{1}{2}\rho S_{\text{ref}} C_L |\vec{v}|\vec{v} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

where $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ represents the outboard spanwise direction. This necessitated the implementation of the cross-product within Simscape. This was achieved by employing `PS Selector` to isolate components of the vectors, `PS Product` to multiply the signals, `PS Subtract` to calculate components of the cross product, and finally `PS Concatenate` to assemble the final vector. The implementation of the block is presented in Figure 3.13.

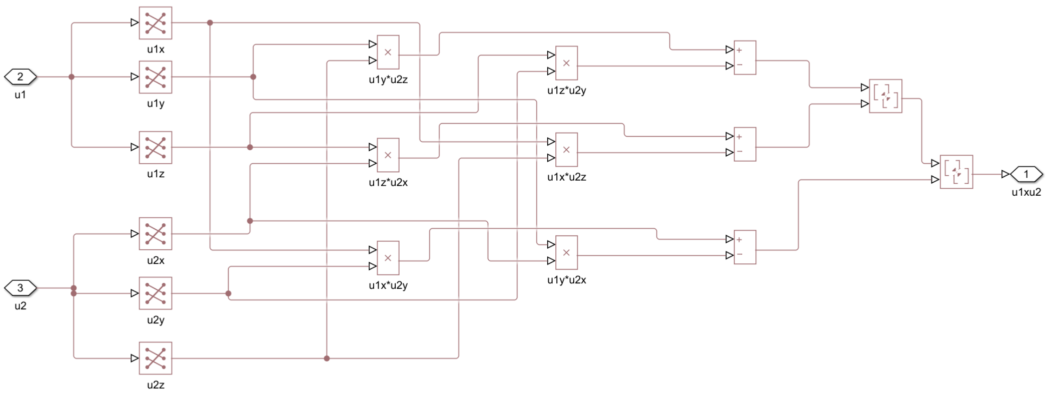


Figure 3.13: Implementation of cross product

As stated previously, the aerodynamic coefficients often depend on angle of attack of body. This can be calculated based on airspeed. `PS Selector` is used to isolate relevant components of the airspeed. Then the angle of attack is computed based on the output of $\mathrm{atan2}(-\mathrm{v_y}, \mathrm{v_x})$. Such formulation caused performance issues at very low angles of attack. The reason was identified to be related to the solver settings. At low angles of attack, high relative accuracy could not be achieved due to numerical errors. For this reason, when $\frac{|v_y|}{|v_x|} \leq 10^{-8}$, the function the output is set to $\frac{-v_y}{v_x}$ (see Listing 3.1). This solves the performance issues while introducing the relative error below $\frac{10^{-16}}{3}$ which is negligible for all relevant purposes (MATLAB states the error as zero when attempting to calculate it in the command line).

Listing 3.1: Implementation of angle of attack calculator

```
inputs
I1; % :left
I2; % :left
end

outputs
O; % :right
end

equations
  if abs(I2)/abs(I1)>1e-8
      O == atan2(-I2,I1);
  else
      O == -I2/I1;
  end
end
```

As already discussed extensively in sections 1.2, 2.3, and 3.1.4, each of the lifting surfaces impacts the flowfield in the whole fluid domain. Hence, modeling of the interactions is of essence in estimating the forces acting on the aircraft. One of the ways to deal with this challenge is to calculate forces/coefficients for the whole aircraft configuration as presented in subsection 3.1.4. Alternatively, the interactions can be handled in the simulation.

To present this paradigm, the interactions based on a simple horseshoe vortex filament were implemented. The vortex is composed of 3 straight finite filaments. According to Biot-Savart law, the velocity induced at point P by a straight finite filament is in the form of

$$v = \frac{\Gamma}{4\pi a}(\sin\theta_2 - \sin\theta_1)$$

where a is distance of point P from the filament, and θ_i are angles formed by the distance and a vector connecting point P with each of the endpoints of the filament. The simplified representation of geometry is presented in Figure 3.14.

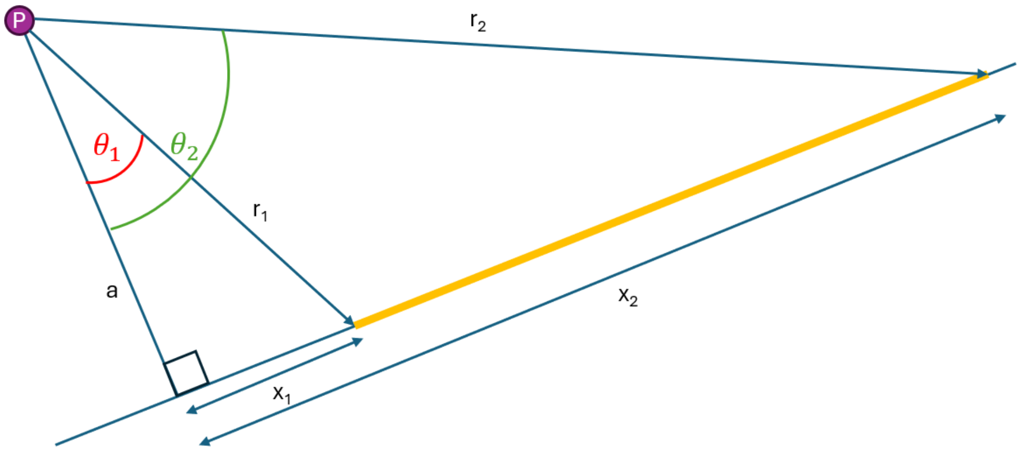


Figure 3.14: Geometrical parameters of a single filament (yellow)

With this notation in mind, the direction of the induced velocity can be denoted as $\frac{\vec{v}}{|\vec{v}|} = \frac{\vec{r_1}\times\vec{r_2}}{|\vec{r_1}\times\vec{r_2}|}$. The distance between point P and the filament is $a = \frac{|\vec{r_1}\times\vec{r_2}|}{|\vec{r_1}-\vec{r_2}|}$. Hence, the induced velocity can be calculated with the following formula.

$$\vec{v} = \frac{\Gamma}{4\pi\frac{|\vec{r_1}\times\vec{r_2}|}{|\vec{r_1}-\vec{r_2}|}}(\frac{x_2}{|\vec{r_2}|} - \frac{x_1}{|\vec{r_1}|})\frac{\vec{r_1}\times\vec{r_2}}{|\vec{r_1}\times\vec{r_2}|}$$

In the simulation, the relative positions $(\vec{r_1}, \vec{r_2})$ and their components (x_1, x_2) are measured with the transform sensors. This was implemented in the simulation as seen in Figure 3.15.

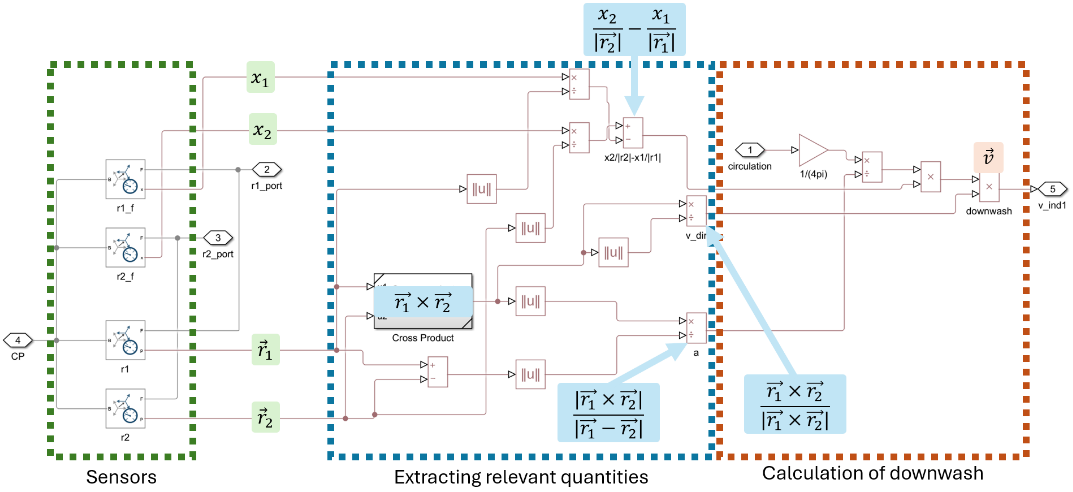

$\frac{x_2}{|\vec{r_2}|} - \frac{x_1}{|\vec{r_1}|}$
x_1
x_2
r1_f
r2_f
r1
r2
4
CP
2
r1_port
3
r2_port
$\vec{r_1}$
$\vec{r_2}$
‖u‖
x2/|r2|-x1/|r1|
$\vec{r_1} \times \vec{r_2}$
Cross Product
v_dir
a
$\frac{|\vec{r_1} \times \vec{r_2}|}{|\vec{r_1} - \vec{r_2}|}$
1
circulation
1/(4pi)
$\vec{v}$
downwash
5
v_ind1
$\frac{\vec{r_1} \times \vec{r_2}}{|\vec{r_1} \times \vec{r_2}|}$
Sensors
Extracting relevant quantities
Calculation of downwash

Figure 3.15: Implementation of calculation of the induced velocity

The velocities induced by all segments of the filament can be added and resolved in the reference frame. In this work, the filaments have been discretized into 3 straight segments but this procedure can be applied for an arbitrary number of segments. In the current simulation the direction of trailing vortices coincides with the chordwise direction and extends 20 times (variable) the span of the aircraft behind the lifting surface. This can be coupled with the analysis described in subsection 3.1.4 to model the filaments more accurately.

The induced velocity at the control point is then added to the sensed airspeed allowing for its inclusion in calculations . The methods above do exist independently but they were also packaged within a single higher level primitive of `Lifting Surface`. At this level, the inputs are relevant aerodynamic coefficients, reference area, air density, the components reference frame with inertial object attached, air reference frame, and perceived downwash as visible in Figure 3.16. The block also automatically outputs the circulation within the filament required for downwash calculation.

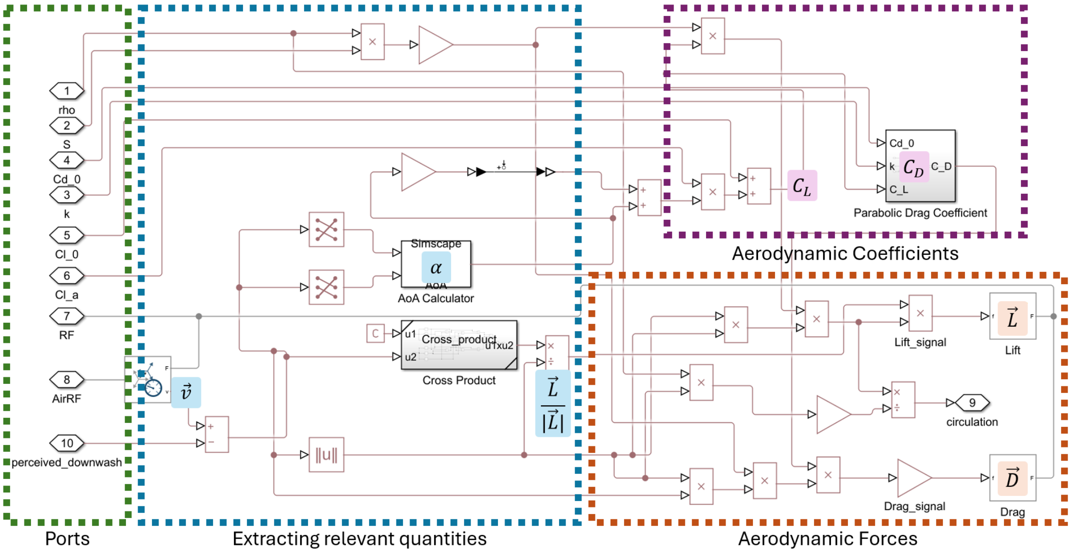


Figure 3.16: Implementation of lifting surface primitive

For some of the analyses presented below (and trimming aircraft), it is useful to find the steady state of the system. This can be done manually by carefully selecting initial conditions. In particular, changing velocity, angle of attack, propulsion force, and horizontal stabilizer's incidence angle allows for finding trimmed conditions.

Alternatively, Simscape offers a built-in tool to find the steady state called `Steady State Manager`. This tool can perform a constrained optimization on state variables attempting to minimize the maximum of the absolute value of the time derivatives of the chosen state variables. While its search domain is limited to states of the system (not related to geometry/model) it can be useful to fine-tune flight conditions for given aircraft. Furthermore, using this tool enables `Model Linearizer` that allows finding state-space representation of the system as well as modeling the effect of perturbations on the system. In this work, the perturbation is introduced as an additional input to the angle of attack. This in principle allows extraction of aerodynamic coefficients from the model should they be needed for external analysis.

Lastly, apart from the built-in visualization, a functionality to connect to Flight Gear was implemented to visualize the flight in the simulator. This does not carry scientific significance but may be used for teaching and popularization of science.

3.3. Case studies

To estimate the impact of including lifting-line-based aerodynamic interactions on the simulated aircraft response a few case studies where envisioned. It should be noted that all of the analyses concern the same aircraft/model and the only differences should come from differences in modeling.

The basic geometry includes a main wing, horizontal stabilizer, vertical stabilizer, and engine/ballast as presented in Figure 3.17. The model data is summarized in Table 3.1. The main wing is positioned at an incidence angle of 3.2°. It is modeled as a cuboid with dimensions of 3.2m x 5m x 0.0625m with a uniform mass of 62.5kg. The horizontal stabilizer is positioned 5m behind and 1m above the main wing. The vertical stabilizer is positioned 0.5m behind horizontal stabilizer. Both stabilizers have dimensions of 0.3m x 1m x 0.05m. The engine/ballast to which propulsion force is applied is placed 1m ahead of the main wing. It is implemented as a cube with a side of 1m and a mass of 250kg. Such a crude modeling is sufficient as in this model there are no aerodynamic forces acting on the engine/ballast. Additionally, this project is not an analysis of any real configuration but rather an analysis of the impact of the

Component	**Dimensions** $(m \times m \times m)$	**Offset** (m, m)	**Incidence angle**	**Mass** (kg)
Wing	3.2 x 5 x 0.0625	(0, 0)	3.2°	62.5
Horizontal stabilizer	0.3 x 1 x 0.05	(5, 1)	Varies	0.9375
Vertical stabilizer	0.3 x 0.05 x 1	(5.5, 1)	0°	10^{-5}
Engine	1 x 1 x 1	(-1, 0)	0°	250

Table 3.1: Data for the aircraft. Dimensions are in the form (length x width x height). Offset is relative to the center of gravity of the main wing in (downstream, upwards) form.

lifting-line-based interaction model. In reality, there would be an aerodynamic interaction between the fuselage and the wing but it is considered out of scope for this project as discussed in section 1.2. The aircraft was modeled at v=89.2m/s and an angle of attack of -2.1°. For the calculation of aerodynamic forces, the lifting surfaces were modeled to have NACA 0012 profile characteristics.

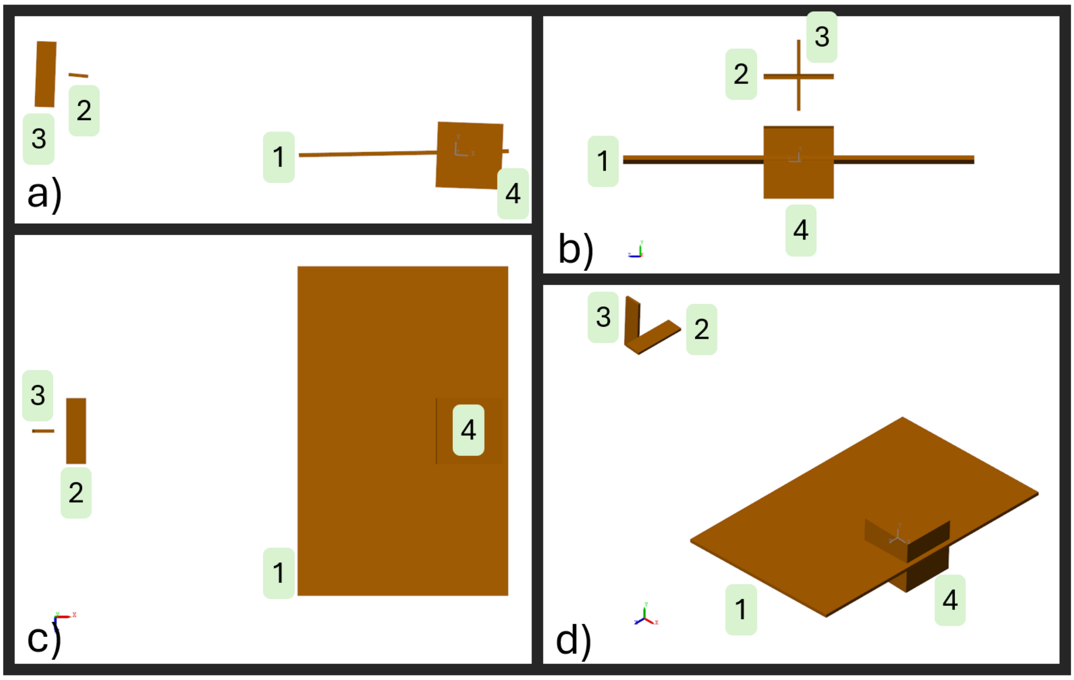


Figure 3.17: Simscape representation of geometry of basic aircraft model: a) Side view b) Front view c) Top view d) Isometric view. The view consist of main wing (1), horizontal stabilizer (2), vertical stabilizer (3), and engine/ballast (4).

The cases considered for this project are described below. Cases 1, 2, and 3 are directed towards answering the overarching research question. Case 4 is aimed to allow comparison with a more traditional approach to aircraft analysis where aerodynamic data is considered at the aircraft level. The additional two cases (1a and 3a) are included to highlight the modularity of the physical network approach and the capabilities of the multi-body modeling where aerodynamic data is prescribed at the component level.

Case 1 - Multiple components with no aerodynamic interaction

Each lifting surface was analyzed separately using DATCOM automated analysis for WING configuration and polars for lift and drag were created. The polars were used to calculate lift slope C_{L_α} and coefficients for parabolic drag polar (k and C_{D_0}). The `Lifting Surfaces` were instantiated with these values with appropriate rigid transformations between them in order to match the geometry from Figure 3.17. In this case, the air velocity perceived by the components was modeled to be the free-stream velocity at the center of gravity of these components.

The incidence angle of the horizontal stabilizer and the propulsive force had to be set in order to maintain a trimmed state of the aircraft. The incidence angle was found manually with a bisection algorithm to achieve a flight without an initial change in pitch angle. In this case, the horizontal stabilizer was set at incidence angle of -3.69°. The required propulsive force was also found with a bisection algorithm to achieve a flight without longitudinal acceleration initially. A propulsive force of 609N was added to trim the aircraft.

Case 1a - Multiple components with no aerodynamic interactions in ISA atmosphere
The same model as Case 1 was simulated with the air density following ISA atmosphere model rather than constant air density.

Case 2 - Multiple components with downwash from main wing
The model with constant density was simulated with model of a downwash of the main wing on the horizontal stabilizer. This means that the velocity perceived by the horizontal stabilizer was composed of free-stream velocity and a downwash induced by the main wing. The sum of the two velocities was used to calculate aerodynamic forces as described in subsection 3.2.4.

Same methodology to find the trimmed state was applied. Due to the change in aerodynamic interaction modeling, the angle of attack of the horizontal stabilizer was increased to -3.25° to keep the configuration trimmed. The power was reduced to 605N to maintain constant speed.

Case 3 - Multiple components with downwash from main wing and horizontal stabilizer
For this case the velocity induced by the main wing and the horizontal stabilizer is considered. This means that the horizontal stabilizer is influenced by the downwash from the main wing. Additionally, the main wing perceives the component of the velocity induced by the horizontal stabilizer. The trimmed conditions were the same as for the Case 2.

The 3 main cases are summarized in Table 3.2. The flowchart visualizing the information exchange is presented in Figure 3.18.

Case 3a - Multiple components connected by flexible beam with downwash from main wing and horizontal stabilizer in ISA atmosphere
The model from Case 3 was simulated with the ISA atmosphere model and the addition of a flexible I-beam instead of a rigid transformation between the main wing and the tail. This case study was meant to present the capability of the approach for more complex cases. Traditionally, the aerodynamic database would need to be created for multiple deflections/twists of the beam and interpolated in between during the simulation. With the multi-body approach with aerodynamic data prescribed at the component level, no additional aerodynamic analysis was required. To fully stretch the wings of the framework, this case is also simulated in asymmetric flight (starting with a banking angle of 5°).

Case	Component	Perceived velocity	Incidence angle
Case 1	Wing	Free-stream	3.2°
	H. Stab	Free-stream	-3.69°
Case 2	Wing	Free-stream	3.2°
	H. Stab	Free-stream + velocity induced by Wing	-3.25°
Case 3	Wing	Free-stream + velocity induced by H. Stab	3.2°
	H. Stab	Free-stream + velocity induced by Wing	-3.25°

Table 3.2: Comparison of the first three cases. Wing refers to the main wing of the aircraft and H. Stab refers to the horizontal stabilizer.

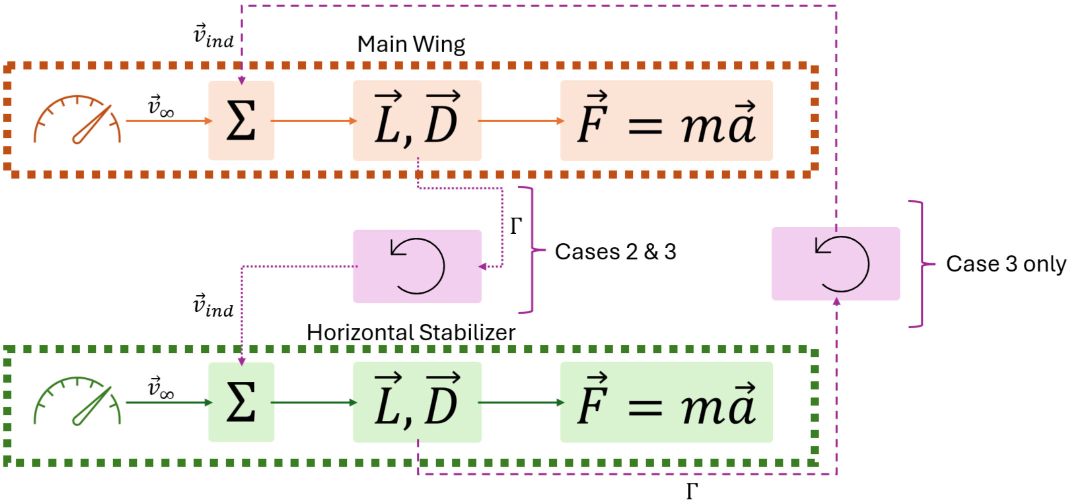


Figure 3.18: Information flowchart for Cases 1, 2, and 3. The information over purple dotted lines is not transferred for Case 1 and purple dashed lines are used only for Case 3.

Case 4 - Single rigid body

The geometry was analyzed in DATCOM (input file presented in Appendix B) as a single WING-BODY-VERTICAL TAIL-HORIZONTAL TAIL configuration. Due to limitations in geometry definition, the engine was modeled as a cylinder. The wings were based on NACA 0012 airfoil. The incidence angle of the horizontal stabilizer was set to be the same as for the Case 3. The data for the whole geometry in the form of polars for C_L, C_m, $C_{L_{\dot{\alpha}}}$, and $C_{m_{\dot{\alpha}}}$ was calculated using automated analysis for the configuration. These were used to calculate lift and moments coefficients. Attached to a single `General Variable Mass` with the same mass and moments of inertia as the original configuration (see Figure 3.17).

3.4. Solution to linearized equations of motion

Analysis for Case 4 provides a valuable opportunity to compare Simscape-based flight dynamics simulation with a more traditional approach of linearized equations of motion. As discussed in subsection 1.1.4, the response of aircraft to small perturbations from steady-state can be described with the fundamental dynamic modes. In this work, the focus is on the symmetric flight so the relevant eigenmodes are short-period oscillations and phugoid. When simulating aircraft's response one can expect the motion to be composed of the eigenmotions. These motions are more pronounced on short and long timescales respectively. The time to half-amplitude for short period oscillations is usually on the order of 1s, whereas the period of phugoid is typically measured in tens of seconds. This means that it should be feasible to distinguish between the eigenmodes when analyzing aircraft's response.

The characteristics of the fundamental motions can be obtained from analytical analysis. The results for short period oscillations are modeled based on a characteristic equation with coefficients provided by Mulder, Staveren, Vaart, *et al.* [89].

$$A = 4K_{y^2}\mu_c^2$$

$$B = -2\mu_c(K_{y^2}C_{z_\alpha} + C_{m_{\dot{\alpha}}} + C_{m_q})$$

$$C = C_{z_\alpha} + C_{m_q} - 2\mu_c C_{m_\alpha}$$

This approximate solution assumes constant velocity over the entire motion. This generally holds valid as the impulse exerted by the net force over a few of times to half-amplitude is small.

The characteristic equation for the phugoid has the following coefficients [89].

$$A = -4\mu_c^2$$

$$B = 2\mu_c C_{X_u} \approx -4\mu_c C_D$$

$$C = -C_{z_u}C_{z_0} \approx -2C_L^2$$

This solution assumes no changes in the angle of attack and negligible pitch rate. This is a rather coarse approximation but it provides valuable comparison of the Simscape-based simulation of long period oscillations with linearized model.

The coefficients are calculated using DATCOM's automated WING-BODY-VERTICAL TAIL-HORIZONTAL TAIL configuration analysis as for Case 4. The approximate solutions for short period motion and phugoid are computed. In particular, time to half-amplitude and period are calculated and used for comparison with the Simscape-based simulations.

4 Results

In this chapter, the simulated response of the aircraft for each of the cases presented in section 3.3 is analyzed. The goal of this section is to estimate the impact of including various lifting-line-based aerodynamic interactions on flight mechanics model when aerodynamic data is prescribed at the component level. Furthermore, the simulated response for the multi-body approach is compared with the case in which the aerodynamic data is prescribed on the aircraft-level and with the period and time to half-amplitude computed using linearized model. Additional comparisons are made to evaluate flexibility of the multi-body approach.

4.1. Case 1 - no aerodynamic interaction

Each of the lifting surfaces was analyzed as an isolated component in DATCOM. The resulting polars are presented in Figures 4.1 and 4.2 for the main wing and stabilizers respectively.

These results were used to model drag with parabolic drag coefficient. For the main wing the drag is estimated by $C_D = 0.0068 + 0.2017C_L^2$. The coefficients for stabilizers are $C_D = 0.0099 + 0.0957C_L^2$.

The response of the aircraft with the data prescribed at the component level with no aerodynamic interactions was simulated. The short period motion is presented in Figure 4.3. Based on the data fit, the time to half-amplitude was estimated to be 0.0238s. The 95% confidence interval on this value is [0.0227, 0.0250]s.

The motion over longer time is presented in Figure 4.4. The time between the first minimum (40.2s) and the first maximum (85.6s) was 45.4s. The amplitude grew from 2.03m to 2.79m (by a factor of 1.33) within this time. This can be used to estimate the time to double amplitude to be 110s with a period of 90.8s.

Note that both the period and the damping change during the motion. For the last two measured extremes, the time difference was 97.4s and the growth ratio of 2.53. This would suggest the time to double amplitude of 72.7s with a period of 194.8s. This may be considered surprising as the analytical solutions to the linearized model are characterized by constant period and damping. However, the linearized model assumes small perturbations from the steady-state. This does not hold true for the simulated response. The larger the oscillations, the higher the significance of the non-linear system response. Since the smallest perturbations occur at the beginning of the oscillations, only the first two extrema are considered for the following cases.

The system's steady state was established for pitch angle of -2.116° and speed of 89.22m/s. Around this point the system was linearized. The state space representation is attached in the Appendix C. From this representation, C_{l_α} and C_{m_α} were computed to be 3.552 and -0.445 respectively.

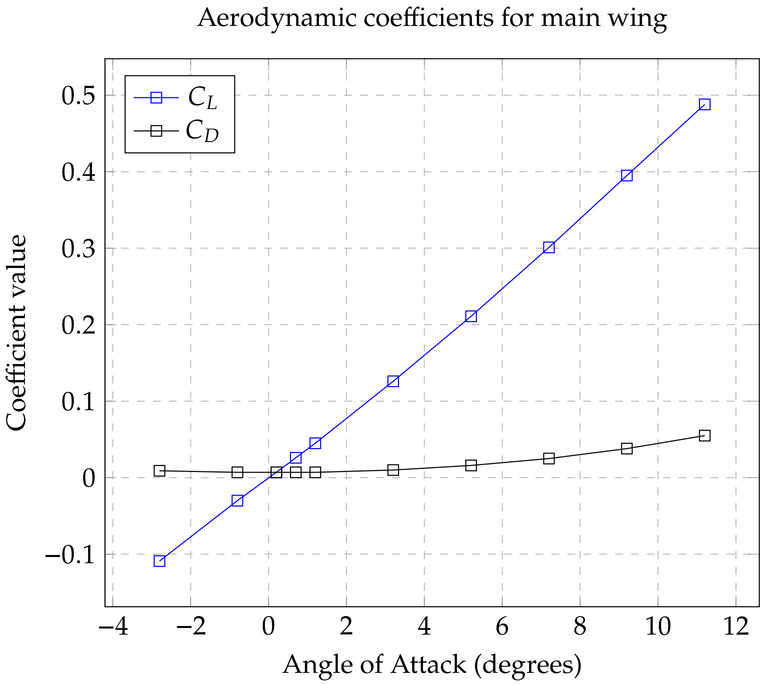


Figure 4.1: Aerodynamic coefficients for the main wing from DATCOM

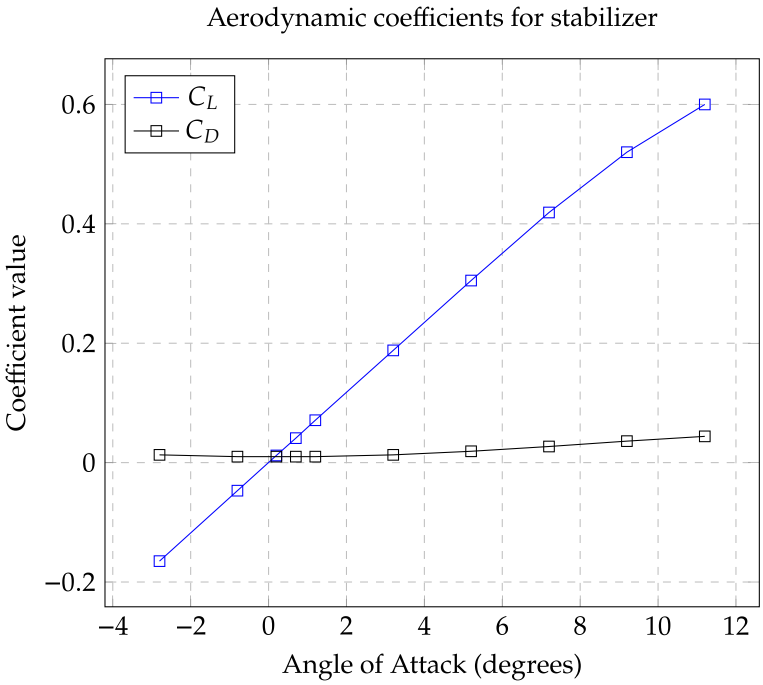


Figure 4.2: Aerodynamic coefficients for the stabilizers from DATCOM

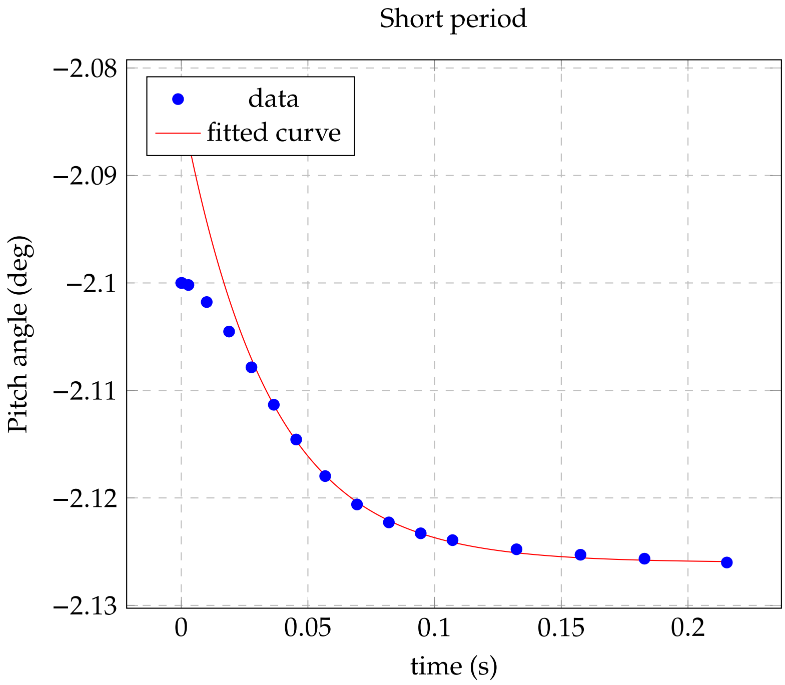


Figure 4.3: Short period motion for Case 1

Case 1a - no aerodynamic interactions in ISA atmosphere

The model above was rerun with the ISA model of atmosphere. As expected, there is no difference for short period motion as the change in altitude and air density is negligible. For longer period, the atmosphere models are compared in Figure 4.5.

The ISA atmosphere model has a damping effect on the oscillations. The period and amplitude is lowered across the motion. This is as expected as during the climb the density is decreasing, lowering the magnitude of aerodynamic forces. This results in smaller upwards acceleration.

4.2. Case 2 - downwash from main wing

The first important result from the analysis of the aircraft with modeled tip vortices from the main wing is that the angle of attack of the horizontal stabilizer had to be increased to trim the aircraft. The main wing vortex filament induces a downwash on the tail, reducing the effective angle of attack.

The parameters of the simulated short period motion are presented in Figure 4.6. With data fitting, the time to half amplitude was estimated to be 0.0236s which is 0.8% smaller compared to Case 1.

The results for longer duration are presented in Figure 4.7. The time interval between the first minimum (59.4s) and the following maximum (114.2s) is 54.9s. In this time interval, the amplitude increased from 4.1m to 9.2m (factor of 2.25). On this basis, the time to double amplitude can be estimated to be 46.9s and the period to be 109.8s. The period of the motion is in accordance with previous simulations but the time to double amplitude decreased significantly. This can be explained by the additional moment generated by the tail due to the downwash. As the aircraft descends, its velocity increases which means that the lift generated by the main wing also increases. This results in a larger downwash at the tail and an additional destabilizing pitch-up moment.

The model's steady state was found to be at a speed of 89.22m/s and an angle of attack of -2.110°. The model was linearized and its state space representation is presented in Appendix C. C_{l_α} and C_{m_α} were computed to be 3.410 and -0.417 respectively. The decrease in the lift slope and increase in the moment coefficient derivative is to be expected. As the angle of attack increases, the lift force generated by the main wing grows causing the downwash perceived by the horizontal tail to rise. Hence, the total lift is reduced and the aircraft experiences an additional pitch-up moment.

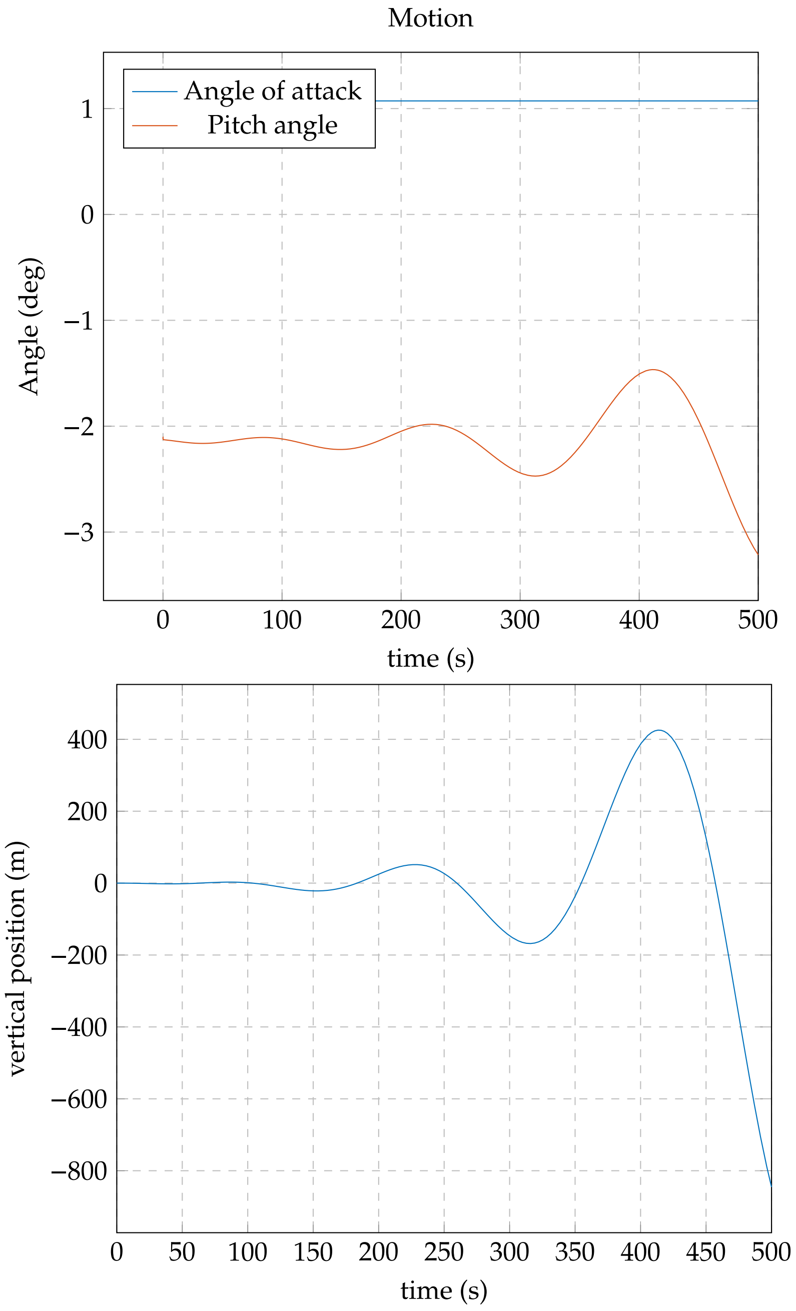


Figure 4.4: Long period motion for Case 1

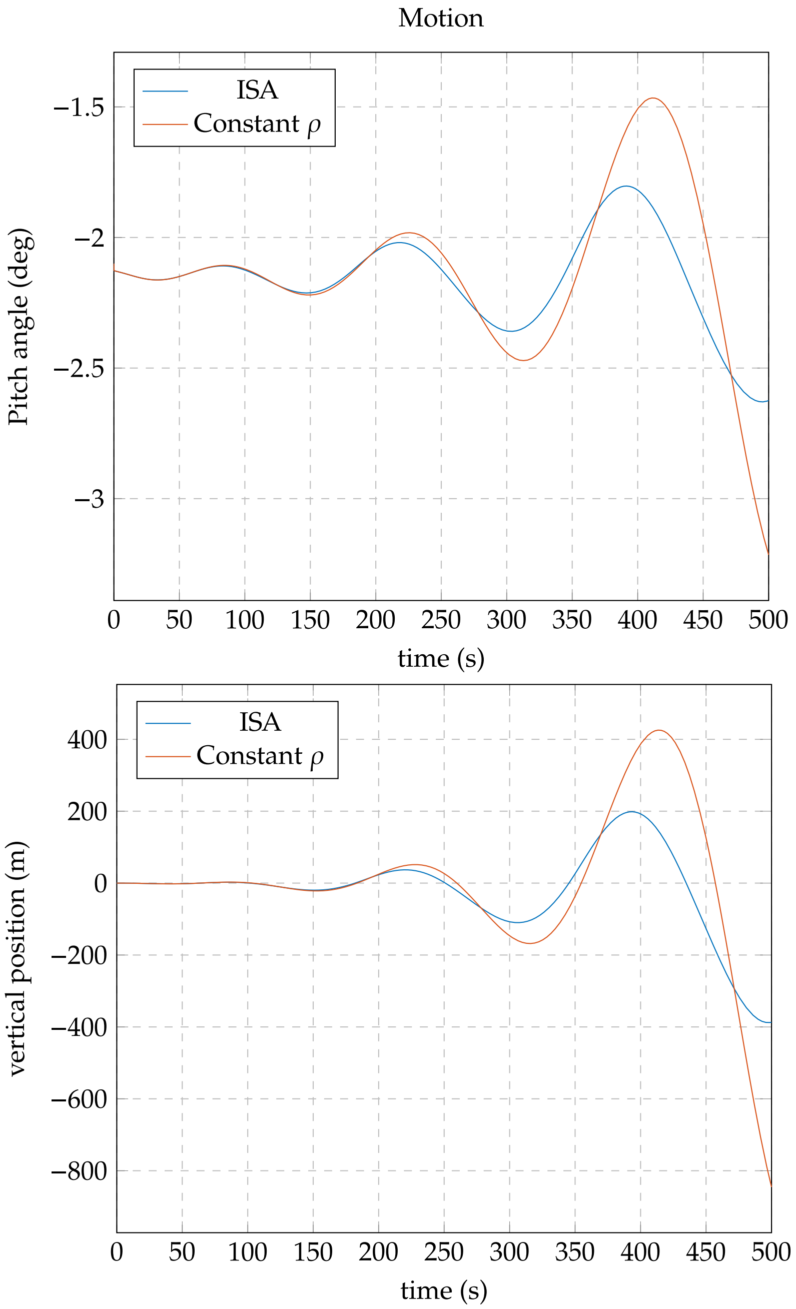


Figure 4.5: Long period motion for Case 1a

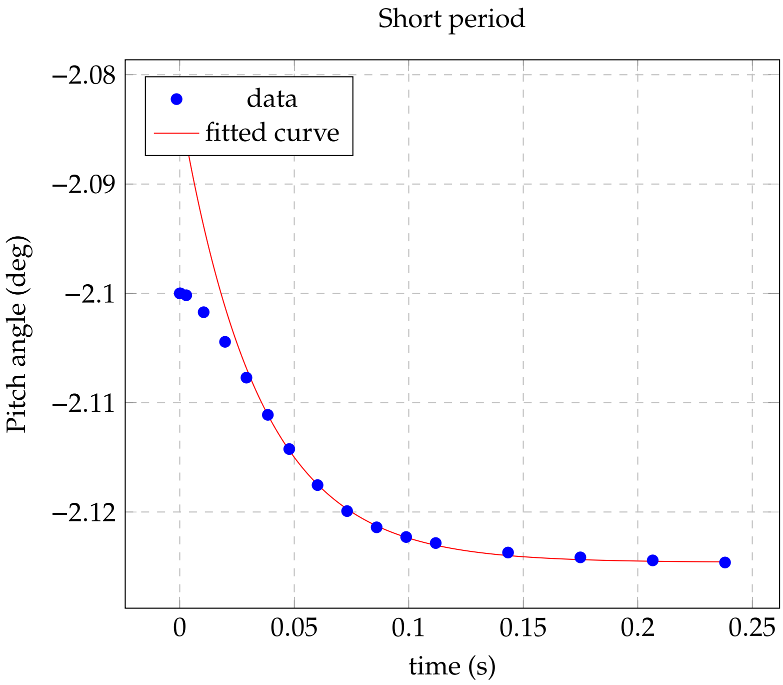


Figure 4.6: Short period motion for Case 2

4.3. Case 3 - downwash from main wing and horizontal stabilizer

In this case, the additional downwash on the main wing causes reduced lift of the main wing compared to the previous case. Based on the short period motion (presented in Figure 4.8), one can estimate the time to half amplitude to be 0.0279s with 95% confidence interval of [0.0247, 0.0319]s. This value is 18.2% larger than the value obtained for simulation without the induced velocity considered at the main wing (Case 2).

The simulated flight data over a longer period compared to the previous case with only downwash from the main wing is presented in Figure 4.9. The time interval between the first minimum (47.4s) and the following maximum (95.2s) is 47.8s. Over this time the amplitude increased from 3.69m to 4.90m (factor of 1.33). This allows the estimation of the period to be 95.6s and the time to double amplitude to be 116s. Yet again, the period is consistent with previous cases.

The model's steady state was found at the speed of 89.21m/s and a pitch angle of -2.105°. Note that the steady-state pitch angle increased compared to the previous case. This is to be expected as the downwash on the main element needs to be offset. The model was linearized and its state space representation is presented in Appendix C. The derivatives of the aerodynamics coefficients are exactly the same for the case in which the downwash from the horizontal stabilizer was ignored.

Case 3a - tail on flexible beam with downwash from main wing and horizontal stabilizer in ISA atmosphere

An advantage of the multibody approach is presented by simulation of the model from Case 3 with a flexible tail and ISA atmosphere. This model is much more complex and would traditionally require additional aerodynamic analysis. With data prescribed at the component level, this case can be simulated with the same aerodynamic data as Cases 1, 2, and 3. There are no expectations of this model being consistent with the previous results but it highlights the flexibility of the framework. The short-period motion is presented in Figure 4.10. The time to half amplitude is estimated to be 0.0211s. The flight parameters for the longer motion are presented in Figure 4.11. The aircraft is steady when moving at a speed of 89.2m/s and a pitch angle of -2.100°. The increase in the pitch angle is likely caused by the bending of the tail and a further increase in the angle of attack of the horizontal stabilizer.

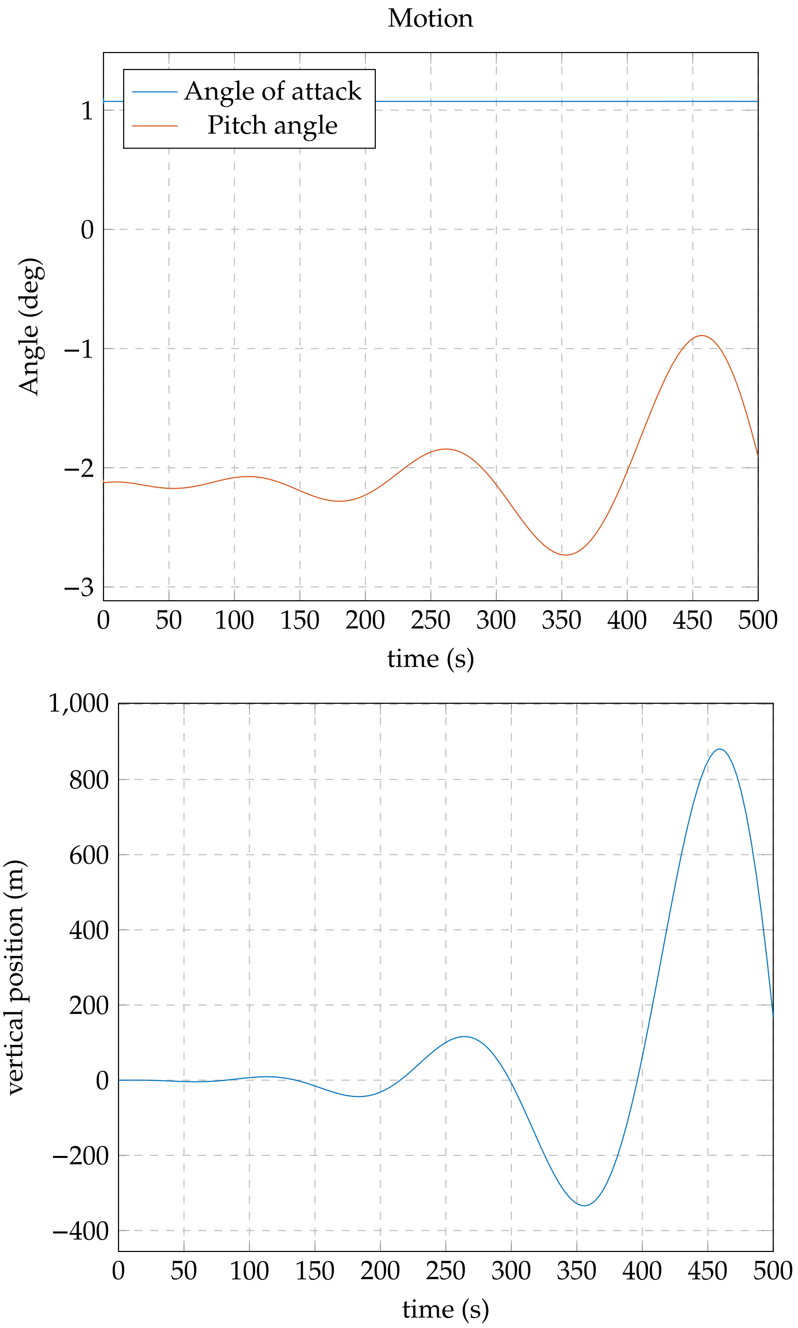


Figure 4.7: Long period motion for Case 2

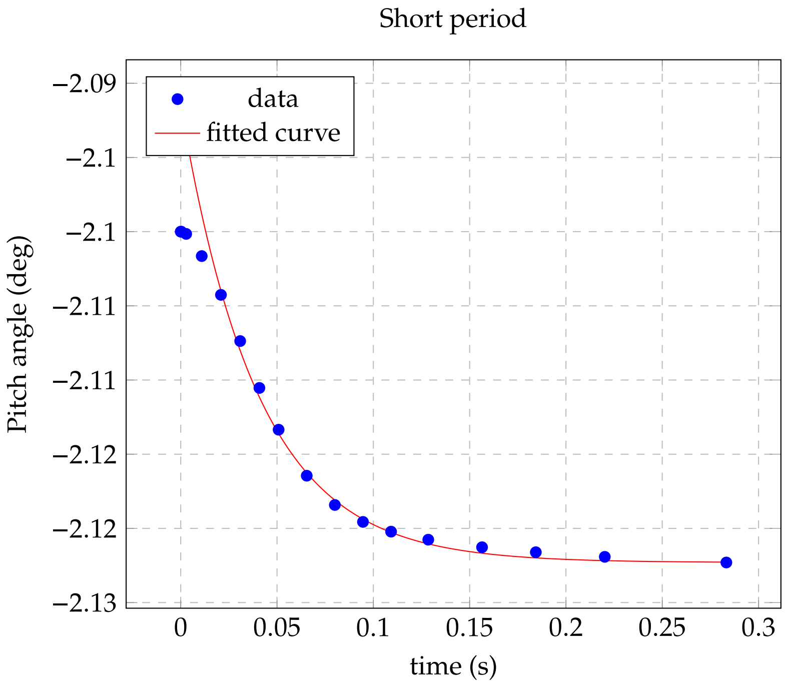


Figure 4.8: Short period motion for Case 3

The trajectory for initial conditions including 5° banking angle are presented in Figure 4.12. The aircraft recovers from the roll and enters the diverging oscillations as expected based on the previous results.

4.4. Case 4 - single rigid body

The geometry was analyzed in DATCOM as a single configuration as presented in Figure 4.13. The resulting coefficients for the whole geometry are presented in Figure 4.14. Note that the $C_{L_{\dot{\alpha}}}$ and $C_{m_{\dot{\alpha}}}$ are derivatives per radian.

One can note that the rotational equilibrium occurs at an angle of attack of around -2.3°. At this point, the lift coefficient is about 0.025 which results in the trimmed airspeed of 110.6 m/s. Note that the presented derivatives are negative which means they have damping effect on the motion.

The short period eigenmotion was isolated by considering the time in which the pitch angle kept decreasing as presented in Figure 4.15. The exponential function was fitted to the data. The resulting function has a form of $0.2306 * 2^{-t/0.1682} - 2.3275$ which corresponds to half-amplitude time of 0.17s. This is far from the value obtained for the multi-body analysis.

However, when the flight path angle is assumed to be small, the phugoid oscillations match the expectations well. The data for vertical position is presented in Figure 4.16. Based on the data fit, the period was calculated to be 53.5s. This is approximately half of the period obtained with the multi-body analysis.

The simulation was also run with no constraint on the path angle. The motion was diverging quickly with the first minimum occurring around t=35.8s as seen in Figure 4.17. To allow for longer motion the simulation was also run with velocity of 112.5m/s which is closer to the equilibrium velocity. The results are presented in Figure 4.18. It took longer for the simulation to diverge but the results were in line with the previous run. The time between the first minimum (46.7s) and the first maximum (112.9s) was 66.1s. The amplitude grew from 28.2m to 182.2m (by a factor of 6.5) within this time. This can be used to estimate the time to double amplitude to be 24.5s with a period of 132.2s. The time to double amplitude is much shorter than any of the runs for multi-body cases. The period is within the same order of magnitude.

With the help of `Steady State Manager`, the trimmed state was found to be $v_x = 112.4$ and $\theta = -2.1°$.

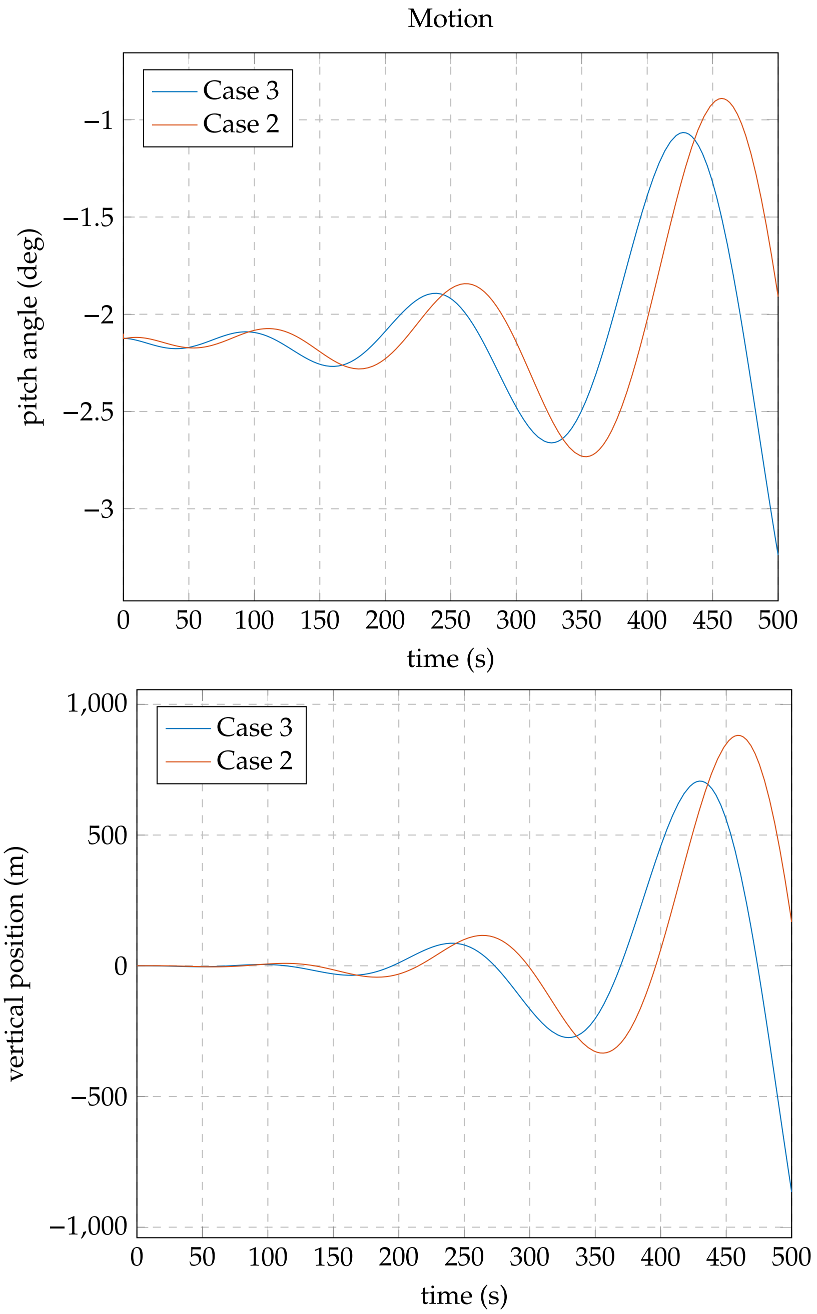


Figure 4.9: Long period motion for Case 3

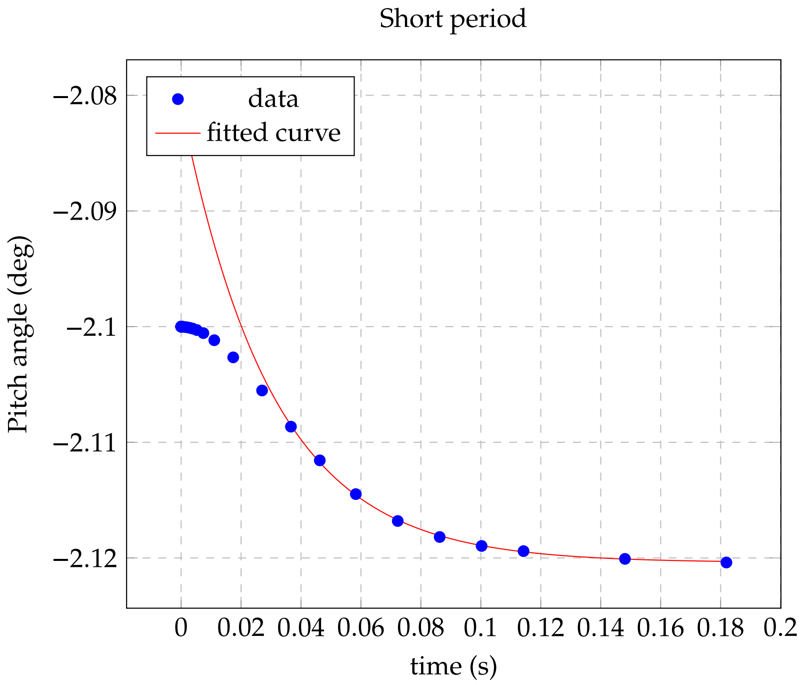


Figure 4.10: Short period motion for Case 3a

This is off from the predicted results by 1.7% and 8.7% respectively. Additionally, the state-space representation was created and is provided in Appendix C. The relevant derivatives of the aerodynamic coefficients were calculated and compared to the source data (Figure 4.14). The result is presented in Table 4.1. The errors for C_{l_α} and C_{m_α} do not exceed 3.4% which increases confidence in the methodology.

	C_{l_α}	$C_{l_{\dot{\alpha}}}$	C_{m_α}	$C_{m_{\dot{\alpha}}}$
Source data	2.254	0.101	-0.664	-0.163
Extracted from State-space	2.177	0.091	-0.664	-0.165
Relative error	-3.4%	-9.9%	0.0%	1.2%

Table 4.1: Comparison of coefficient derivatives

4.5. Solution to linearized equations of motion

Based on the coefficients presented in Figure 4.14 and the geometry, two solutions to the linearized equations of motion describing short-period oscillations were obtained as described in section 3.4. The time to half-amplitude was calculated to be 0.0288s or 0.0033s. In reality, the motion would be composed of these two solutions.

$$\theta = 2^{-\frac{t}{0.0288}}A + 2^{-\frac{t}{0.0033}}B$$

where A and B are the coefficients representing the relative significance of each of the solutions. The first one will be the one more pronounced at larger timescales. Note that it is within 3.2% of the time calculated for the case when downwash from the main wing and horizontal stabilizer was considered and falls well within the 95% confidence interval.

For the phugoid eigenmotion, the oscillations have a period of 50.09s. This is close to the estimated value of $0.453V = 50.10$s as predicted by Mulder, Staveren, Vaart, *et al.* [89]. There is no damping of the motion as the drag is set to 0.

4.6. Synthesis

This section provides a comparison of the cases and a discussion of the results. A table comparing the relevant coefficient derivatives for the case studies is presented in Table 4.2. These were extracted from

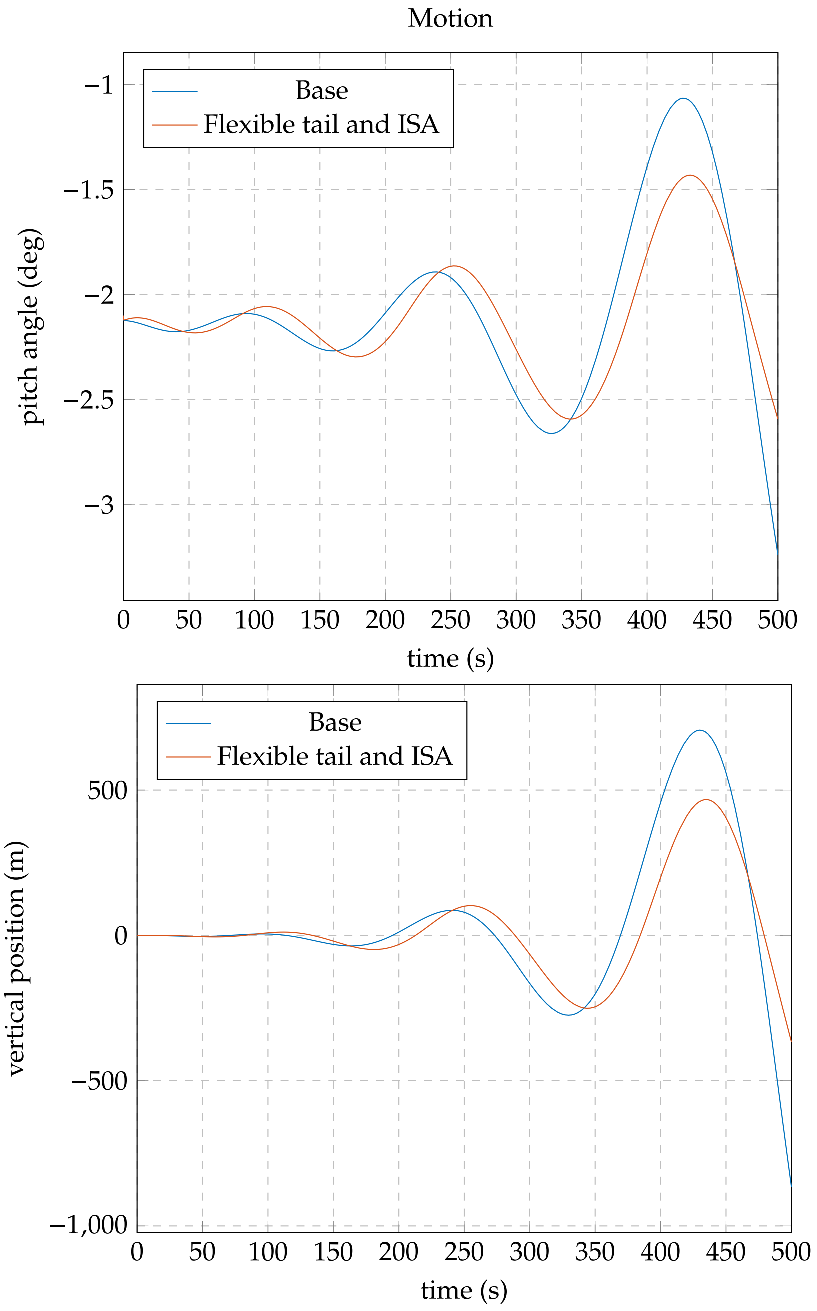


Figure 4.11: Long period motion for Case 3a

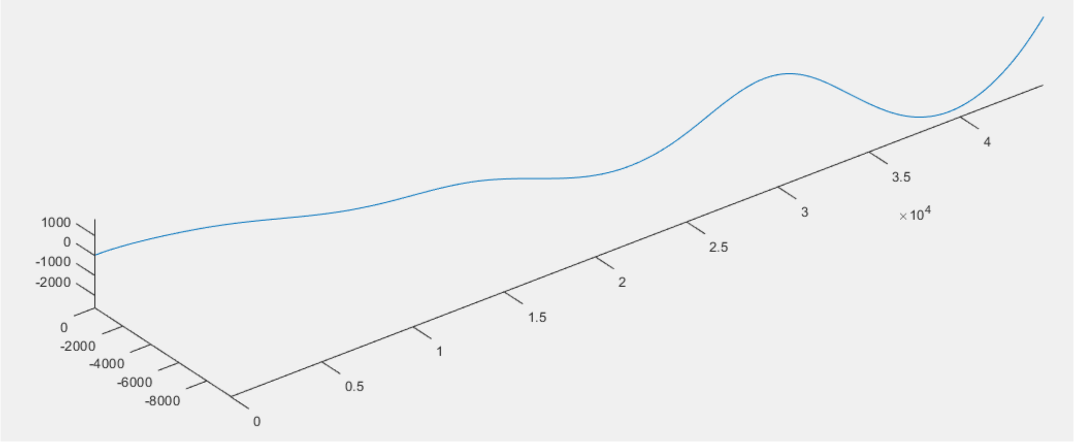


Figure 4.12: Banking trajectory for Case 3a

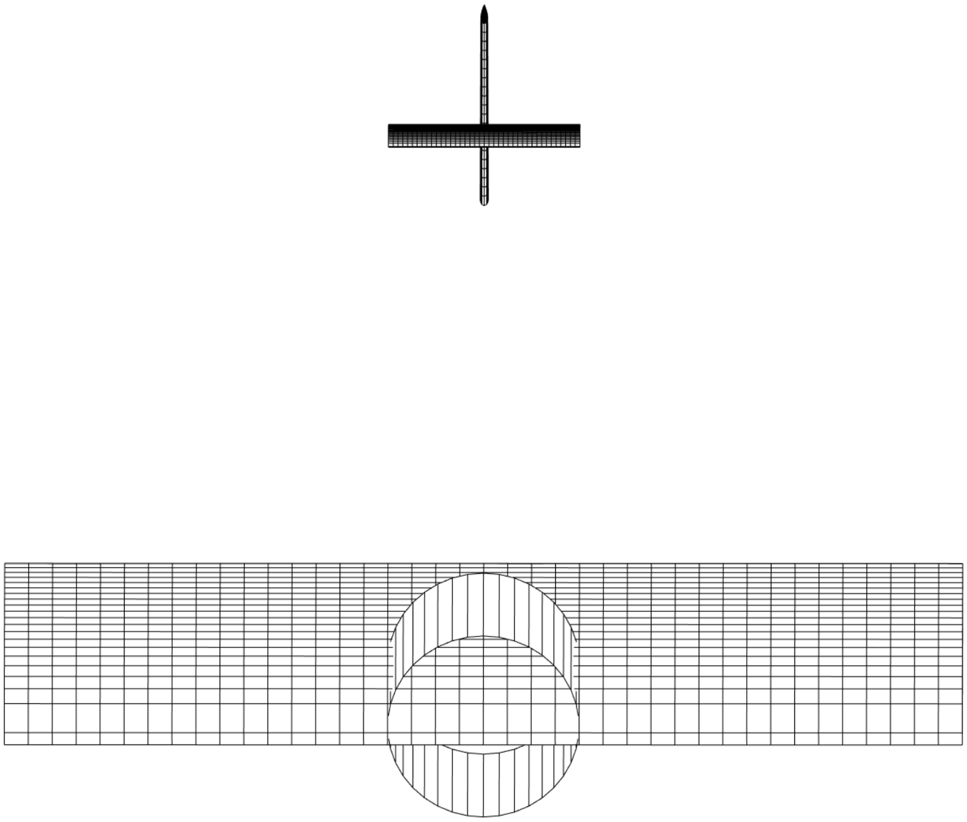


Figure 4.13: Geometry analyzed in DATCOM plotted in datcom3d [90]

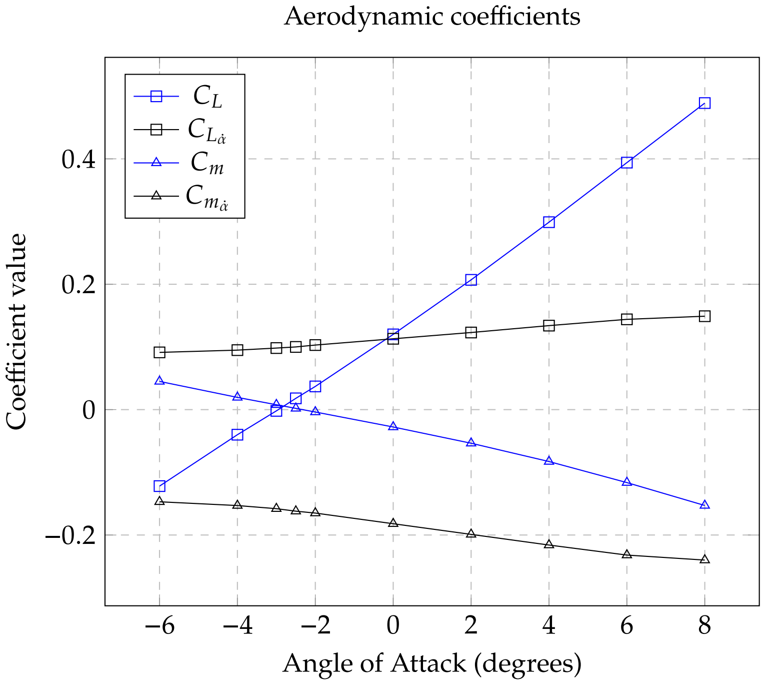


Figure 4.14: Aerodynamic coefficients for the whole geometry from DATCOM

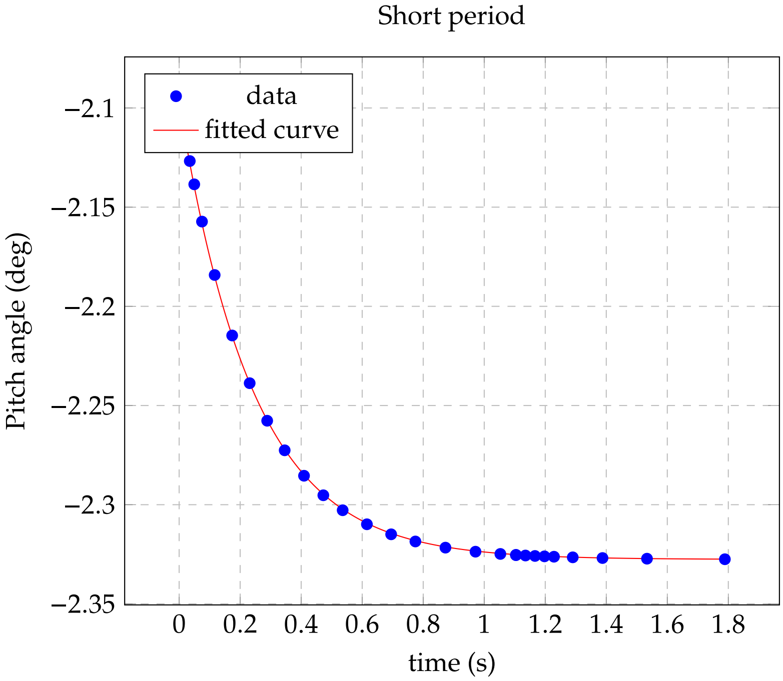


Figure 4.15: Short period eigenmotion for Case 4

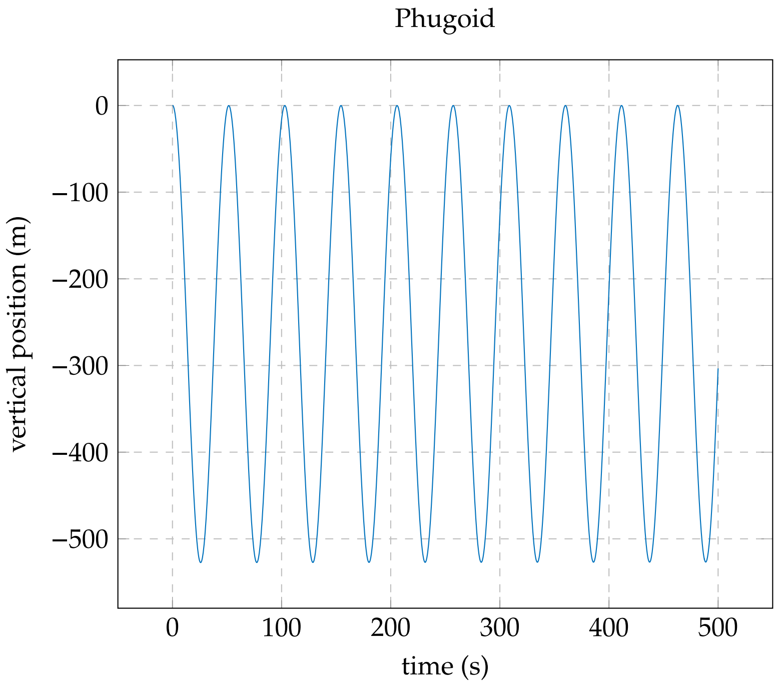


Figure 4.16: Long period eigenmotion for Case 4

state-space representations of the studied systems. It is worth noting that the derivatives with respect to the rate of change of angle of attack are zero for all of the studied cases involving multiple components. This is because no unsteady aerodynamic effects have been included in the multibody model.

	C_{l_α}	$C_{l_{\dot\alpha}}$	C_{m_α}	$C_{m_{\dot\alpha}}$
Case 1	3.552	0.000	-0.445	0.000
Case 2	3.410	0.000	-0.417	0.000
Case 3	3.410	0.000	-0.417	0.000
Case 4	2.177	0.091	-0.664	-0.165

Table 4.2: Comparison of coefficient derivatives

The multibody modeling (Cases 1-3) does not seem consistent with the single-body model (Case 4). The multibody cases, however, are consistent with each other. Consideration of the downwash from the main wing at the location of the horizontal stabilizer resulted in a reduction of C_{l_α} by 4% and an increase in C_{m_α} of 6%. Introducing downwash from the horizontal stabilizer on the main wing has negligible impact on the coefficient derivatives. This is to be expected as the lift of the main wing is larger and the horizontal stabilizer is downstream of the main wing which results in a small distance from trailing vortices to the stabilizer.

Furthermore, the incidence angle of the horizontal stabilizer in the cases with interactions between aerodynamic components had to be increased to keep the aircraft trimmed. This is to be expected as the downwash reduces the effective angle of attack of the stabilizer. Implementation of the ISA atmosphere model results in the damping effect of decreasing air density as expected. This increases confidence in the multibody paradigm.

The characteristics of the eigenmotions of the aircraft were also compared. The long-period oscillations were difficult to quantify. The data-fitting algorithm did not produce any meaningful results. This is likely caused by the non-linearity of the system. The period based on the first two extrema of the motion was within the same order of magnitude as presented in Table 4.3. Considering the period varied by a factor of about 2 (see section 4.1) within the simulated motion, the exact effect of the aerodynamic interactions was not possible to be quantified. The time to double-amplitude was not

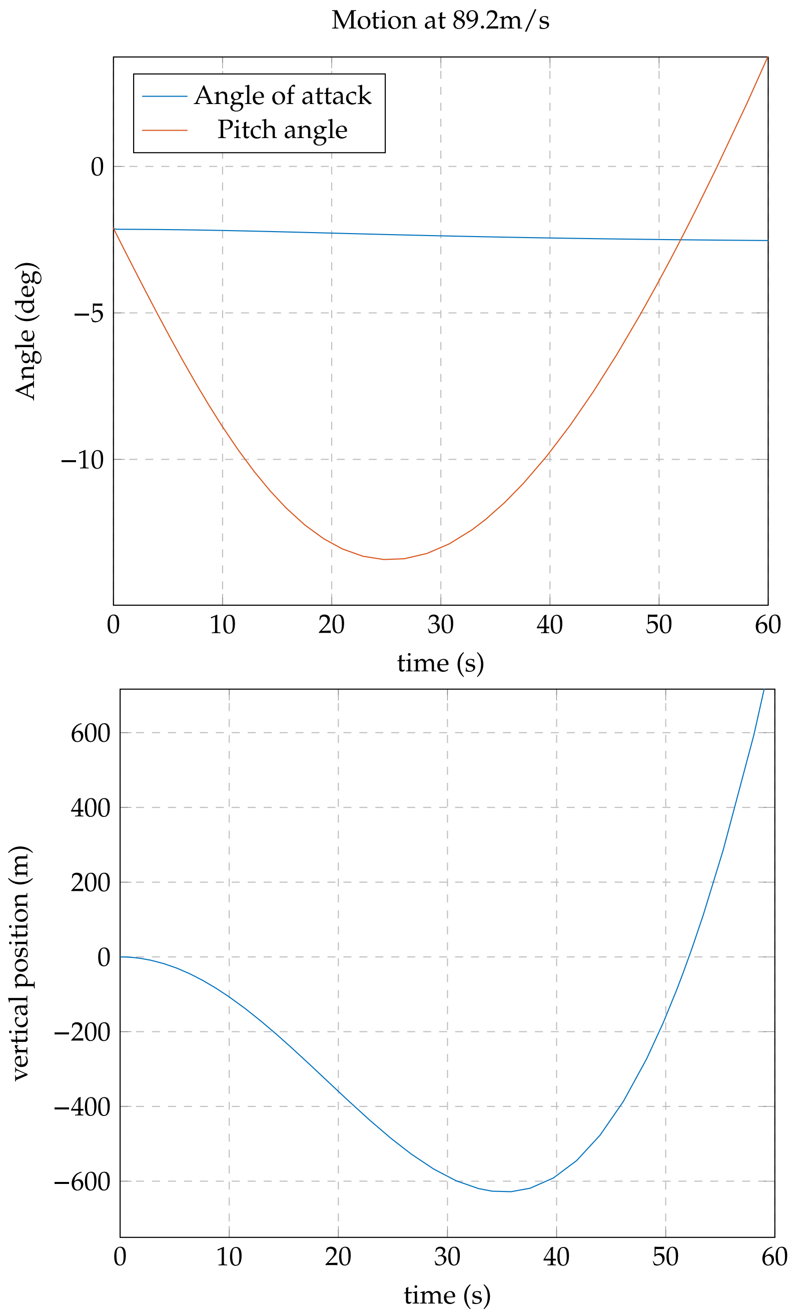


Figure 4.17: Simulated motion at 89.2m/s for Case 4

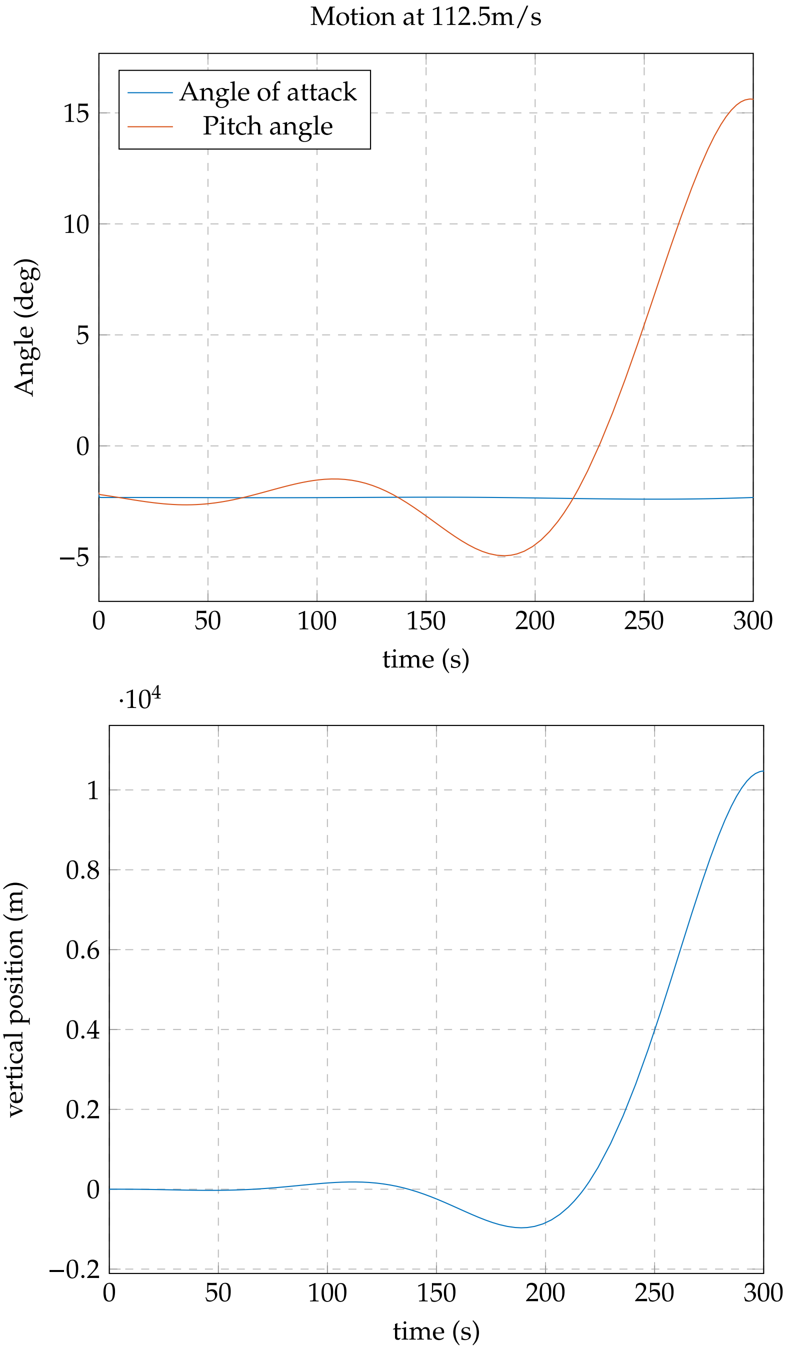


Figure 4.18: Simulated motion at 112.5m/s for Case 4

consistent within motions. It can be said, however, that for all cases the motion could be described as diverging oscillations.

	Case 1	Case 2	Case 3	Case 4
period(s)	90.8	109.8	95.6	132.2

Table 4.3: Comparison of period for phugoid

The short-period motion was more suitable for analysis. The time to half-amplitude is compared in Table 4.4. Interestingly, the simulated response of the single rigid body with coefficients from DATCOM (Case 4) did not match the result at all. It has been noted that the angle of attack and its time derivative have been inconsistent depending on the method of calculation. The analytical formulation for the latter resulted in a seemingly unrelated error (`Degenerate mass distribution`). This could be the reason why the single-body case does not match the analytical value well but more research is needed to fully understand the phenomenon. For the multibody case, the addition of downwash reduced the error relative to the solution of the linearized equation of motion from 18% to 3%. This could suggest that the more the model represents reality, the lower the error.

	Linearized	Case 1	Case 2	Case 3	Case 4
half-amplitude time(s)	0.0288	0.0238	0.0236	0.0279	0.1682
relative error	NA	-17%	-18%	-3%	484%

Table 4.4: Comparison of time to half-amplitude for short period

5 Conclusions

This work explored a paradigm in which isolated aircraft components are analyzed separately and combined together in a multi-body system representative of the whole geometry. The FLAPERON toolbox was developed and employed to assess the impact of vortex-filament-based aerodynamic interactions on the simulated aircraft's response when considering aerodynamic data only at the component level. Physical networks created in Simscape were used to simulate symmetrical flight and assess key stability characteristics of a configuration including a main wing, a horizontal stabilizer, and a vertical stabilizer. To achieve this goal, wings and the model of their downwash were added to the FLAPERON toolbox working towards the high-level primitives.

For the studied geometry with aerodynamic data prescribed separately for the main wing, horizontal stabilizer, and vertical stabilizer, the consideration of lifting-line-based aerodynamic interactions had an impact on the simulated aircraft response and aerodynamic coefficient derivatives. Consideration of the downwash from the main wing for the calculation of aerodynamic forces acting on the horizontal stabilizer resulted in a decrease of half-amplitude time of short-period motion by less than 1%. The derivatives of the aerodynamic coefficients obtained from state-space representations of the models were also impacted. C_{l_α} decreased by 4.0% (0.142), while C_{m_α} increased by 6.3% (0.028). Additionally, the incidence angle of the horizontal stabilizer had to be increased by 0.44° to keep the aircraft trimmed. Consideration of the downwash from the horizontal stabilizer for the calculation of aerodynamic forces acting on the main wing resulted in an increase of half-amplitude time of short-period motion by over 18%. This aerodynamic interaction did not result in a change of aerodynamic coefficient derivatives and did not require adjustment of the incidence angle of the horizontal stabilizer.

The results suggest that neglecting the aerodynamic impact the horizontal stabilizer has on the main wing is justified to calculate aerodynamic coefficients for this conventional configuration. This supports the validity of the methodology used by Soikkeli, Matko, and Koopman [68]. On the other hand, the velocity induced by the horizontal stabilizer has a significant impact on the simulated response of the aircraft. This suggests that the evaluation of the flying qualities of the aircraft with a multi-body approach should include consideration of seemingly irrelevant aerodynamic interactions.

It is important to consider the limitations of this research. The conclusions are based only on the analysis of this particular geometry. The exact effect of various combinations of interactions is likely to be aircraft-configuration-dependent. This research can be considered a proof-of-concept as the methodology used is configuration-agnostic and is expected to allow insight into an arbitrary geometry.

Another limitation of this work is very simple aerodynamic modeling. Each of the vortex filaments is composed of 3 straight segments fixed in space (relative to the aircraft). More advanced modeling techniques allow the vortex filaments to be convected by the flowfield (free-wake). Additionally, one vortex filament per lifting surface assumes that lift distribution on this surface is constant. This is in general not true. A more representative model could include more vortex filaments per lifting surface or discretizing wings/stabilizers into more lifting surfaces. This would also permit the calculation of the angle of attack for each section separately making it effectively equivalent to the nonlinear vortex step method [58].

Considering the limitations, looking at this project in a wider context is wise. This work lays a foundation for a new paradigm in flight mechanics modeling in which aerodynamic data is prescribed at the component level. The aircraft's response was successfully simulated with aerodynamic forces calculated for each lifting surface. The lifting-line-based aerodynamic interactions were also implemented in the physical networks. This shows that the multi-body paradigm can be used to develop flight mechanics models.

Additionally, the work completed allows further research into the topic. New modules were implemented in the Simscape environment enabling one to model lifting surfaces and their downwash based on high-level inputs. The modularity and flexibility of the approach were confirmed with the variety of considered cases. Additionally, the FLAPERON toolbox was developed working towards high-level primitives for wings. These developments allow one to set up flight simulations for other aircraft configurations and discretization.

Further research into multi-body aircraft modeling with the aerodynamic data prescribed at the component level could result in a more efficient evaluation of the flying qualities of new aircraft designs or improvement of the current ones. The efficiency of the approach is highlighted by the analysis of Case 3a with a tail on a flexible beam. No additional aerodynamic analysis was required to model the variable geometry. This would also be the case if the relative positions of the components were changed. This offers a large benefit compared to the traditional approach in which the geometry needs to be re-analyzed with every change.

The paradigm explored in this work is far from maturity. Further research is required to prove the validity of the approach. This work has shown that there are significant discrepancies between single-body and multi-body modeling. The modeling of interactions in DATCOM (used for single-body analysis) is based on semi-empirical methods rather than the lifting-line approach used in this work. Hence, it is expected that some discrepancies will be present. It would be worth considering comparing the results with an analysis suite that is based on vortex filaments to evaluate the validity of the approach.

Furthermore, the sensitivity analysis would be helpful to increase confidence and gain insight into the generalizability of the results. In particular, the sensitivity of the starting conditions would be worth considering. Each of the models was started close to their trimmed condition. The exact difference between the starting point and the trimmed state could have an impact on the results that were not considered in this work. Additionally, it might be valuable to evaluate sensitivity to the design of the geometry. This could help in evaluating to what extent the conclusions are valid only for the considered configuration.

There exist more valuable extensions to this work. Gaining insight into multi-body modeling of diverse configurations and evaluating appropriate modeling of aerodynamic interactions may prove crucial when developing a configuration-agnostic flight mechanics model. This research can also be extended to consider asymmetrical flight. This would provide the researchers with more opportunities to validate the multi-body approach and help it gain maturity. Furthermore, considering the simplicity of modeling a configuration with a flexible tail (Case 3a), it would be valuable to evaluate the capability of the multi-body approach to model aeroelastic effects.

Finally, to aid the potential research directions mentioned above, it would be valuable to further develop the FLAPERON toolbox to support an automatic generation of flight mechanics models. Manually setting up the simulations in Simscape is a time-consuming process. If it could be (partially) automated, the researchers could focus more on gaining insight into the approach. In principle, the Simscape components developed in this work can be placed and connected automatically. A series of steps that the author recommends to automate the process is provided in Appendix D. If this is developed further to include dependency tracking and lazy evaluation, it may be possible to move towards a future in which knowledge-based flight mechanics modeling is a reality. This would allow for the more efficient inclusion of the flight model in the design and analysis of aircraft.

References

[1] G. Duranton and M. A. Turner, “The fundamental law of road congestion: Evidence from us cities,” *American Economic Review*, vol. 101, no. 6, pp. 2616–52, 2011. DOI: 10.1257/aer.101.6.2616. [Online]. Available: https://www.aeaweb.org/articles?id=10.1257/aer.101.6.2616%20https://pubs.aeaweb.org/doi/pdfplus/10.1257/aer.101.6.2616.

[2] S. Matthes, U. Schumann, V. Grewe, *et al.*, “Climate optimized air transport,” in *Atmospheric Physics: Background–Methods–Trends*. Springer, 2012, pp. 727–746. DOI: 10.1007/978-3-642-30183-4_44.

[3] A. P. Cohen, S. A. Shaheen, and E. M. Farrar, “Urban air mobility: History, ecosystem, market potential, and challenges,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 6074–6087, 2021, ISSN: 1524-9050. DOI: 10.1109/TITS.2021.3082767.

[4] A. Hahn, “Vehicle sketch pad: A parametric geometry modeler for conceptual aircraft design,” in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, p. 657. DOI: 10.2514/6.2010-657.

[5] V. R. C. Munjulury, “Knowledge based integrated multidisciplinary aircraft conceptual design,” Thesis, 2014.

[6] A. Wald, C. Fay, and R. Gleich, *Introduction to aviation management*. LIT Verlag Münster, 2010, vol. 3.

[7] S. Khosravi and D. W. Zingg, “Aerostructural perspective on winglets,” *Journal of Aircraft*, vol. 54, no. 3, pp. 1121–1138, 2017. DOI: 10.2514/1.c033914.

[8] D. MacManus and M. Slaby, “Intake ground vortex and computational modelling of foreign object ingestion,” *The Aeronautical Journal*, vol. 119, no. 1219, pp. 1123–1145, 2015. DOI: 10.1017/s0001924000011167.

[9] P. Kardasz, J. Doskocz, M. Hejduk, P. Wiejkut, and H. Zarzycki, “Drones and possibilities of their using,” *J. Civ. Environ. Eng*, vol. 6, no. 3, pp. 1–7, 2016. DOI: 10.4172/2165-784x.1000233.

[10] J. Baichtal, *Building Your Own Drones: A Beginners' Guide to Drones, UAVs, and ROVs*. Que Publishing, 2015, ISBN: 9780134000152.

[11] T. R. Yechout, S. Morris, D. Bossert, W. Hallgren, and J. Hall, “Introduction to aircraft flight mechanics: Performance, static stability, dynamic stability, classical feedback control, and state-space foundations. ed. aiaa education series. 2014, reston, virginia: American institute of aeronautics and astronautics,” *Inc. xvii*, 2003.

[12] D. P. Witkowski, A. K. Lee, and J. P. Sullivan, “Aerodynamic interaction between propellers and wings,” *Journal of Aircraft*, vol. 26, no. 9, pp. 829–836, 1989. DOI: 10.2514/6.1988-665.

[13] D. Grant and R. Lind, “Effects of time-varying inertias on flight dynamics of an asymmetric variable-sweep morphing aircraft,” in *AIAA atmospheric flight mechanics conference and exhibit*, 2007, p. 6487. DOI: 10.2514/6.2007-6487.

[14] J. Anderson, *EBOOK: Fundamentals of Aerodynamics (SI units)*. McGraw hill, 2011.

[15] B. Peerlings, “Holistically improving screening decisions under uncertainty in aircraft conceptual design and technology assessment: Insights on bottom-up uncertainty quantification and propagation and integrated socio-technical group decision making,” Thesis, Delft University of Technology, 2019.

[16] P. Moin and K. Mahesh, “Direct numerical simulation: A tool in turbulence research,” *Annual review of fluid mechanics*, vol. 30, no. 1, pp. 539–578, 1998. DOI: 10.1146/annurev.fluid.30.1.539.

[17] C. D. Argyropoulos and N. Markatos, “Recent advances on the numerical modelling of turbulent flows,” *Applied Mathematical Modelling*, vol. 39, no. 2, pp. 693–732, 2015. DOI: 10.1016/j.apm.2014.07.001.

[18] P. R. Spalart, "Detached-eddy simulation," *Annual review of fluid mechanics*, vol. 41, pp. 181–202, 2009. DOI: 10.1146/annurev.fluid.010908.165130.

[19] G. Alfonsi, "Reynolds-averaged navier–stokes equations for turbulence modeling," 2009. DOI: 10.1115/1.3124648.

[20] H.-S. Tsien and L. Lees, "The glauert-prandtl approximation for subsonic flows of a compressible fluid," *Journal of the Aeronautical Sciences*, vol. 12, no. 2, pp. 173–187, 1945. DOI: 10.2514/8.11219.

[21] H. Wagner, "On the origin of the dynamic lift of airfoils," 1924.

[22] T. Theodorsen, "General theory of aerodynamic instability and the mechanism of flutter," *NASA reference publication*, 1934.

[23] T. von Karman and W. R. Sears, "Airfoil theory for non-uniform motion," *Journal of the aeronautical sciences*, vol. 5, no. 10, pp. 379–390, 1938. DOI: 10.2514/8.674.

[24] H. E. Taha and A. S. Rezaei, "On the dynamics of unsteady lift and circulation and the circulatory-non-circulatory classification," in *AIAA Scitech 2019 Forum*, 2019, p. 1853. DOI: 10.2514/6.2019-1853.

[25] D. Greenwell, "A review of unsteady aerodynamic modelling for flight dynamics of manoeuvrable aircraft," in *AIAA atmospheric flight mechanics conference and exhibit*, 2004, p. 5276. DOI: 10.2514/6.2004-5276.

[26] F. Tian and M. Voskuijl, "Automated generation of multiphysics simulation models to support multidisciplinary design optimization," *Advanced Engineering Informatics*, vol. 29, no. 4, pp. 1110–1125, 2015. DOI: 10.1016/j.aei.2015.07.004.

[27] H. M. Paynter, "Analysis and design of engineering systems," *MIT press*, 1961.

[28] S. Karimian and Z. Jahanbin, "Aerodynamic modeling of a flexible flapping-wing micro-air vehicle in the bond graph environment with the aim of assessing the lateral control power," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 233, no. 13, pp. 4998–5015, 2019. DOI: 10.1177/0954410019835980.

[29] A. W. Wymore, *Model-based systems engineering*. CRC press, 2018. DOI: 10.1201/9780203746936.

[30] G. Ferretti, G. Magnani, and P. Rocco, "Virtual prototyping of mechatronic systems," *Annual Reviews in Control*, vol. 28, no. 2, pp. 193–206, 2004. DOI: 10.1016/j.arcontrol.2004.02.002.

[31] G. La Rocca and M. Van Tooren, "Enabling distributed multi-disciplinary design of complex products: A knowledge based engineering approach," *Journal of Design Research*, vol. 5, no. 3, pp. 333–352, 2007. DOI: 10.1504/JDR.2007.014880.

[32] A. Bérard, A. Rizzi, and A. T. Isikveren, "Cadac: A new geometry construction tool for aerospace vehicle pre-design and conceptual design," in *Collection of Technical Papers - AIAA Applied Aerodynamics Conference*, Export Date: 28 April 2024; Cited By: 15. DOI: 10.2514/6.2008-6219. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-77957759649&doi=10.2514%2f6.2008-6219&partnerID=40&md5=25e5d3e6491198bf7c5ad8252b3f4969.

[33] R. Sinha, C. J. Paredis, and P. K. Khosla, "Integration of mechanical cad and behavioral modeling," in *Proceedings 2000 IEEE/ACM International Workshop on Behavioral Modeling and Simulation*, IEEE, 2000, pp. 31–36. DOI: 10.1109/BMAS.2000.888361.

[34] P. Bhattacharya, N. S. Welakwe, R. Makanaboyina, and A. Chimalakonda, "Integration of catia with modelica," in *proceedings of Modelica conference*, 2006.

[35] A. Biahmou, A. Fröhlich, and J. Stjepandic, "Improving interoperability in mechatronic product development," *Proceedings of PLM*, pp. 510–521, 2010.

[36] A. Morris, P. Arendsen, G. LaRocca, M. Laban, R. Voss, and H. Hönlinger, "Mob-a european project on multidisciplinary design optimisation," in *24th ICAS Congress*, ICAS Yokohama, Japan, 2004. DOI: 10.4172/2165-784x.1000233.

[37] G. La Rocca, "Knowledge based engineering techniques to support aircraft design and optimization," Thesis, Delft University of Technology, 2011.

[38] G. La Rocca, "Knowledge based engineering: Between ai and cad. review of a language based technology to support engineering design," *Advanced engineering informatics*, vol. 26, no. 2, pp. 159–179, 2012. DOI: 10.1016/j.aei.2012.02.002.

[39] F. Tian and M. Voskuijl, "Knowledge based engineering to support automotive conceptual design and automatic control software development," in *Proceedings of the FISITA 2012 World Automotive Congress: Volume 6: Vehicle Electronics*, Springer, 2013, pp. 393–405. DOI: 10.1007/978-3-642-33829-8_37.

[40] R. Elmendorp, R. Vos, and G. La Rocca, "A conceptual design and analysis method for conventional and unconventional airplanes," in *ICAS 2014: Proceedings of the 29th Congress of the International Council of the Aeronautical Sciences, St. Petersburg, Russia, 7-12 September 2014*, International Council of Aeronautical Sciences, 2014.

[41] R. Elmendorp and G. La Rocca, "Comparative design & sensitivity studies on box-wing airplanes," in *Italian Association of Aeronautics and Astronautics-XXV International Congress, Rome*, 2019.

[42] A. Rizzi, P. Eliasson, T. Goetzendorf-Grabowski, J. B. Vos, M. Zhang, and T. S. Richardson, "Design of a canard configured transcruiser using ceasiom," *Progress in Aerospace Sciences*, vol. 47, no. 8, pp. 695–705, 2011. DOI: 10.1016/j.paerosci.2011.08.011.

[43] B. Mialon, A. Khrabrov, S. B. Khelil, *et al.*, "Validation of numerical prediction of dynamic derivatives: The dlr-f12 and the transcruiser test cases," *Progress in Aerospace Sciences*, vol. 47, no. 8, pp. 674–694, 2011. DOI: 10.1016/j.paerosci.2011.08.010.

[44] M. Alder, E. Moerland, J. Jepsen, and B. Nagel, "Recent advances in establishing a common language for aircraft design with cpacs," 2020.

[45] S. Dominka, E. Broecker, and F. Schiller, "Automated execution of simulation studies demonstrated via a simulation of a car," in *2008 Winter Simulation Conference*, IEEE, 2008, pp. 2916–2924. DOI: 10.1109/WSC.2008.4736414.

[46] K. Kural, M. Voskuijl, T. Fengnian, and J. Pauwelussen, "Determination of representative loading conditions for effective semitrailer design," *Transport*, vol. 29, no. 4, pp. 363–375, 2014. DOI: 10.3846/16484142.2014.982174.

[47] M. Voskuijl, J. de Klerk, and D. van Ginneken, "Flight mechanics modeling of the prandtlplane for conceptual and preliminary design," in *Variational Analysis and Aerospace Engineering: Mathematical Challenges for Aerospace Design: Contributions from a Workshop held at the School of Mathematics in Erice, Italy*, Springer, 2012, pp. 435–462. DOI: 10.1007/978-1-4614-2435-2_19.

[48] M. Rose, H. Yaralian, J. Wagster, and S. Bhandari, "Development and validation of flight dynamics model of a uav airplane," in *Infotech@ Aerospace 2012*, 2012, p. 2592. DOI: 10.2514/6.2012-2592.

[49] R. Lykins, R. Riley, G. Garcia, and S. Keshmiri, "Modal analysis of 1/3-scale yak-54 aircraft through simulation and flight testing," in *AIAA Atmospheric Flight Mechanics Conference*, 2011, p. 6443. DOI: 10.2514/6.2011-6443.

[50] P. Thomas, T. Richardson, and A. Cooke, "Estimation of stability and control derivatives for a piper cub j-3 remotely piloted vehicle," in *AIAA Modeling and Simulation Technologies Conference*, 2012, p. 5013. DOI: 10.2514/6.2012-5013.

[51] O. Dantsker and M. Vahora, "Comparison of aerodynamic characterization methods for design of unmanned aerial vehicles," in *2018 AIAA Aerospace Sciences Meeting*, 2018, p. 0272. DOI: 10.2514/6.2018-0272.c1.

[52] M. Moelyadi, Y. Rohmahwati, A. Nugraha, *et al.*, "Cfd based determination of longitudinal static and dynamic stability derivatives of twin boom uav," *Journal of Applied Science and Engineering*, vol. 22, no. 2, pp. 259–266, 2019.

[53] E. T. Yerly, "Investigation into active spanwise camber deformation on the lateral stability and roll control of the x-56a compared to conventional ailerons," 2016.

[54] A. Da Ronch, K. Badcock, C. McFarlane, *et al.*, "Benchmarking ceasiom software to predict flight control and flying qualities of the b-747," 2010.

[55] S. K. Rahmani, Z. J. Wang, J. J. Matt, H. Chao, and C. Zheng, "Comparison of low-and high-order cfd based estimates of forces, moments and aerodynamic coefficients with uas flight test data," in *AIAA AVIATION 2022 Forum*, 2022, p. 4065. DOI: 10.2514/6.2022-4065.

[56] M. W. Lee and K. D. Visser, "Towards an effective nonplanar wing design strategy," in *34th AIAA Applied Aerodynamics Conference*, 2016, p. 4328. DOI: 10.2514/6.2016-4328.

[57] M. E. Pérez Segura, D. T. Mook, and S. Preidikman, "General-purpose object-oriented framework for vorticity-dominated flow simulation," *Journal of Aerospace Information Systems*, vol. 17, no. 10, pp. 562–580, 2020. DOI: 10.2514/1.I010818.

[58] C. Van Dam, J. V. Kam, and J. Paris, "Design-oriented high-lift methodology for general aviation and civil transport aircraft," *Journal of aircraft*, vol. 38, no. 6, pp. 1076–1084, 2001. DOI: 10.2514/2.2875.

[59] S. Gallay and E. Laurendeau, "Preliminary-design aerodynamic model for complex configurations using lifting-line coupling algorithm," *Journal of Aircraft*, vol. 53, no. 4, pp. 1145–1159, 2016. DOI: 10.2514/1.C033460.

[60] M. Parenteau, K. Sermeus, and E. Laurendeau, "Vlm coupled with 2.5 d rans sectional data for high-lift design," in *2018 AIAA Aerospace Sciences Meeting*, 2018, p. 1049.

[61] D. Marten, M. Lennie, G. Pechlivanoglou, C. N. Nayeri, and C. O. Paschereit, "Implementation, optimization, and validation of a nonlinear lifting line-free vortex wake module within the wind turbine simulation code qblade," *Journal of Engineering for Gas Turbines and Power*, vol. 138, no. 7, p. 072 601, 2016. DOI: 10.1115/1.4031872.

[62] A. Deperrois, "Xflr5 analysis of foils and wings operating at low reynolds numbers," *Guidelines for XFLR5*, vol. 142, 2009.

[63] M. Hassanalian and A. Abdelkefi, "Design, manufacturing, and flight testing of a fixed wing micro air vehicle with zimmerman planform," *Meccanica*, vol. 52, pp. 1265–1282, 2017. DOI: 10.1007/s11012-016-0475-2.

[64] P. Kurukularachchi, S. Munasinghe, and H. De Silva, "Stability analysis for a twin boom h-tail medium scale uav through simulated dynamic model," in *2016 Moratuwa Engineering Research Conference (MERCon)*, IEEE, 2016, pp. 415–420. DOI: 10.1109/MERCon.2016.7480177.

[65] J. Mariens, A. Elham, and M. Van Tooren, "Quasi-three-dimensional aerodynamic solver for multidisciplinary design optimization of lifting surfaces," *Journal of Aircraft*, vol. 51, no. 2, pp. 547–558, 2014. DOI: 10.2514/1.C032261.

[66] A. Elham and M. J. van Tooren, "Coupled adjoint aerostructural wing optimization using quasi-three-dimensional aerodynamic analysis," *Structural and Multidisciplinary Optimization*, vol. 54, pp. 889–906, 2016. DOI: 10.2514/6.2015-2487.

[67] V. Mosca, S. Karpuk, A. Sudhi, C. Badrya, and A. Elham, "Multidisciplinary design optimisation of a fully electric regional aircraft wing with active flow control technology," *The Aeronautical Journal*, vol. 126, no. 1298, pp. 730–754, 2022. DOI: 10.1017/aer.2021.101.

[68] J. S. Soikkeli, D. Matko, and T. Koopman, "Cascaded nonlinear dynamic inversion applied to a fixed-wing distributed electric propulsion aircraft," in *AIAA AVIATION 2023 Forum*, 2023, p. 4048. DOI: 10.2514/6.2023-4048.

[69] J. Katz and B. Maskew, "Unsteady low-speed aerodynamic model for complete aircraft configurations," *Journal of Aircraft*, vol. 25, no. 4, pp. 302–310, 1988. DOI: 10.2514/6.1986-2180.

[70] J. Murua, R. Palacios, and J. M. R. Graham, "Applications of the unsteady vortex-lattice method in aircraft aeroelasticity and flight dynamics," *Progress in Aerospace Sciences*, vol. 55, pp. 46–72, 2012. DOI: 10.1016/j.paerosci.2012.06.001.

[71] A. B. G. Neto, R. G. da Silva, and P. Paglione, "Control-point-placement method for the aerodynamic correction of the vortex-and the doublet-lattice methods," *Aerospace Science and Technology*, vol. 37, pp. 117–129, 2014.

[72] M. Goman and A. Khrabrov, "State-space representation of aerodynamic characteristics of an aircraft at high angles of attack," *Journal of Aircraft*, vol. 31, no. 5, pp. 1109–1115, 1994. DOI: 10.2514/3.46618.

[73] S. Schmidt and D. M. Newman, “Estimation of dynamic stability derivatives of a generic aircraft,” in *Proceedings of the 17th Australasian Fluid Mechanics Conference*, 2010.

[74] J. F. Le Roy and S. Morgand, “Saccon cfd static and dynamic derivatives using elsa,” in *28th AIAA applied aerodynamics conference*, 2010, p. 4562. DOI: `10.2514/6.2010-4562`.

[75] B.-g. Mi, H. Zhan, B.-b. Chen, *et al.*, “New systematic cfd methods to calculate static and single dynamic stability derivatives of aircraft,” *Mathematical Problems in Engineering*, vol. 2017, 2017. DOI: `10.1155/2017/4217217`.

[76] L. Fitzgerald, A. Niven, and P. Griffin, “A static stability comparison of wind tunnel and computational fluid dynamics methods,” in *33th ICAS Congress 2022*, 2022.

[77] A. D. Ronch, D. Vallespin, M. Ghoreyshi, and K. Badcock, “Evaluation of dynamic derivatives using computational fluid dynamics,” *AIAA journal*, vol. 50, no. 2, pp. 470–484, 2012. DOI: `10.2514/1.J051304`.

[78] C. A. Mader and J. R. Martins, “Computing stability derivatives and their gradients for aerodynamic shape optimization,” *AIAA journal*, vol. 52, no. 11, pp. 2533–2546, 2014. DOI: `10.2514/1.J052922`.

[79] J. W. Quitter, M. Marino, and J. M. Bauschat, “Comparison of aerodynamic methods for flight mechanical derivative estimation of unconventional aircraft,” in *AIAA Scitech 2021 Forum*, 2021, p. 0324. DOI: `10.2514/6.2021-0324`.

[80] C. M. Pappalardo, M. Curcio, and D. Guida, “Modeling the longitudinal flight dynamics of a fixed-wing aircraft by using a multibody system approach.,” *IAENG International Journal of Computer Science*, vol. 50, no. 1, 2023.

[81] T. S. Chyczewski, M. Dubiel, D. R. McDaniel, *et al.*, “A position on current stability and control prediction capabilities and a path forward,” in *AIAA AVIATION 2020 FORUM*, 2020, p. 2678. DOI: `10.2514/6.2020-2678`.

[82] G. Looye, “The new dlr flight dynamics library,” in *Proceedings of the 6th International Modelica Conference*, Citeseer, vol. 1, 2008, pp. 193–202.

[83] C. Heimans, “Aerodynamic analysis of engine integration during the preliminary design phase,” Thesis, Delft University of Technology, 2021.

[84] C. Varriale, A. Raju Kulkarni, G. La Rocca, and M. Voskuijl, “A hybryd, configuration-agnostic approach to aircraft control surface sizing,” in *25th International Congress of the Italian Association of Aeronautics and Astronautics (AIDAA)*, 2019.

[85] K. Hameeteman, “Unconventional propulsive empennage-future or fiction?: Stability and control analysis and the effect of scaling of the duuc,” Thesis, Delft University of Technology, 2017.

[86] C. Varriale, K. Hameeteman, M. Voskuijl, and L. L. Veldhuis, “A thrust-elevator interaction criterion for aircraft optimal longitudinal control,” in *AIAA Aviation 2019 Forum*, 2019, p. 3001. DOI: `10.2514/6.2019-3001`.

[87] J. Katz and A. Plotkin, *Low-speed aerodynamics*. Cambridge university press, 2001, vol. 13.

[88] R. Damiani, F. F. Wendt, J. M. Jonkman, and J. Sicard, “A vortex step method for nonlinear airfoil polar data as implemented in kiteaerodyn,” in *AIAA Scitech 2019 Forum*, 2019, p. 0804.

[89] J. Mulder, W. van Staveren, J. van der Vaart, *et al.*, *Ae3202 flight dynamics lecture notes*, Lecture notes, Mar. 2013.

[90] J. Mohammed, *Datcom3d*, GitHub repository, last modified on Nov 9, 2021, Nov. 2021. [Online]. Available: `https://github.com/robojafar/datcom3d`.

A

Relevant Code

Listing A.1: Twist, sweep, and dihedral

```
function sections = get.sections(w)
    arguments, w (1,1) Wing, end
    for i = numel(w.airfs):-1:1
        sections(i).airf = w.airfs(i);
        sections(i).twist = w.twist(i);
        sections(i).sweep = w.sweep(i);
        sections(i).dihedral = w.dihedral(i);
    end
end

function twist = get.twist(w)
    for i = numel(w.airfs):-1:1
        a = w.airfs(i);
        ang = a.base.ori.ang;
        if isequal(ang(1), ang(3))
            twist(i,1) = (ang(2)) * cos(ang(1));
        end
    end
end

function sweep = get.sweep(w)
    for i = numel(w.airfs):-1:1
        a = w.airfs(i);
        sweep(i,1) = asin((a.base.0.x)/abs(a.base.0.y));
    end
end

function dihedral = get.dihedral(w)
    for i = numel(w.airfs):-1:1
        a = w.airfs(i);
        dihedral(i,1) = asin(a.base.0.z/abs(a.base.0.y));
    end
end
```

Listing A.2: Mirrored wing

```
function set.airfs(w, airfs)
    arguments, w (1,1) Wing, airfs Airfoil {mustBeVector}, end
    if w.isMirrored
        for i = numel(airfs)-1:-1:1
            mirrored_airfs(i) = copyElement(airfs(i+1));
        end
        w.airfs = [flip(mirrored_airfs(:)); airfs(:)];
    else
        w.airfs = airfs(:);
    end
end

```

```
function set_airf_base(w, airfs, y, twist, sweep, dihedral)
    %SET_AIRFS_BASE Set the position and orientation of the base
    % of each airfoil related the Wing. The Airfoil is set so
    % that its leading edge lies on the Wing Base y axis, and is
    % rotated about -y by an angle equal to the twist
    if w.isMirrored
        j=(numel(airfs)+1)/2;
        for i = numel(airfs):-1:1
            if j==0
                j=1;
            end
            if i<numel(airfs)/2
                j=j+1;
                airfs(i).base.O.set([y(j)*sin(sweep(j)); -y(j);
                    abs(y(j))*sin(dihedral(j))], w.base);
                airfs(i).base.ori.set([0, 0, 0], "321", w.base);
                airfs(i).base.revol( ...
                    Orien([0; -twist(j); 0], "321", w.base), ...
                    airfs(i).fraction_chord(0.0));
            else
                airfs(i).base.O.set([y(j)*sin(sweep(j)); y(j);
                    abs(y(j))*sin(dihedral(j))], w.base);
                airfs(i).base.ori.set([0, 0, 0], "321", w.base);
                airfs(i).base.revol( ...
                    Orien([0; -twist(j); 0], "321", w.base), ...
                    airfs(i).fraction_chord(0.0));
                j=j-1;
            end
        end
    else
        for i = numel(airfs):-1:1
            airfs(i).base.O.set([y(i)*sin(sweep(i)); y(i); y(i)*sin(dihedral(i))], w.base);
            airfs(i).base.ori.set([0, 0, 0], "321", w.base);
            airfs(i).base.revol( ...
                Orien([0; -twist(i); 0], "321", w.base), ...
                airfs(i).fraction_chord(0.0));
        end
    end
end
```

Listing A.3: Implementation of normal vectors at control point

```
function [normal_vectors] = normal_vectors(w)
    arguments
        w (1,1) LiftingLineWing
    end
    % normal vectors represent the normal to the surface at the control points
    front_auxilary_points = w.front_auxilary_points;
    back_guiding_points = w.back_guiding_points;
    midpoints = ([front_auxilary_points.xyz] + [back_guiding_points.xyz])/2;
    for i = numel(w.airfs):-1:1
        v1 = Vector(midpoints(i) - back_guiding_points(i+1).xyz, w.root_airfoil.base);
        v2 = Vector(midpoints(i+1) - back_guiding_points(i).xyz, w.root_airfoil.base);
        normal_vectors(i) = v1.cross(v2);
        normal_vectors(i) = normal_vectors(i)/norm(normal_vectors(i).xyz);
    end
end
```

Listing A.4: Implementation of induced velocity at control point

```
function [induced_velocity] = v_ind_coefs(w,p)
    arguments
        w (1,1) LiftingLineWing
        p (1,1) Point
    end
    % induced velocity coefficients per unit gamma at point p due to each of the filaments
    vortex_filaments = w.vortex_filaments;
    for i = numel(vortex_filaments(:,1)):-1:1
        induced_velocity_xyz(:,i) = [0; 0; 0];
        for j = 1:numel(vortex_filaments(1,:))-1
            for k=1:10
```

```
                p_copy = Point(p.xyz, p.wrt).wrt_rf(w.root_airfoil.base);
                r = p_copy.xyz - vortex_filaments(i,j).xyz*(1-(k-0.5)/10)
                    - vortex_filaments(i,j+1).xyz*((k-0.5)/10);
                dl = (vortex_filaments(i,j+1).xyz - vortex_filaments(i,j).xyz)/10;
                induced_velocity_xyz(:,i) = induced_velocity_xyz(:,i)
                    + (1/(4*pi))*cross(dl, r)/(norm(r)^3);
            end
        end
        induced_velocity(i) = Vector(induced_velocity_xyz(:,i)', w.root_airfoil.base);
    end
end
```

Listing A.5: Computation of coefficient matrix

```
function [coefficient_matrix] = compute_coefficient_matrix(l)
    arguments
        l (1,1) LiftingLineSolver
    end
    % compute the coefficient matrix for the system of equations
    % to solve the circulation distribution
    components = l.wind_profile.aerocomponents;

    cps = l.compute_control_points();
    norm_vectors = l.compute_normal_vectors();

    coefficient_matrix = [];

    for i = numel(cps):-1:1
        % compute the coefficient matrix for each component
        coefficients = [];
        for j = numel(components):-1:1
            % compute the influence of the j-th component on the i-th
            % control point
            coefs_vector = components(j).v_ind_coefs(cps(i));
            for k = 1:numel(coefs_vector)
                coefs_vector(k) = Vector(coefs_vector(k).xyz, coefs_vector(k).wrt);
                coefs_vector(k).wrt_rf(l.wrt_rf);
                coefs(k) = coefs_vector(k).dot(norm_vectors(i));
            end
            coefficients = [coefficients, coefs];
        end
        coefficient_matrix(i,:) = coefficients;
    end
end
```

Listing A.6: Calculation of circulation distribution

```
function [circulations] = compute_circulations(l)
    arguments
        l (1,1) LiftingLineSolver
    end
    % compute the circulation distribution on the wing
    % based on the lifting line theory
    A = l.compute_coefficient_matrix();
    B = l.compute_airspeeds();

    B=B';
    circulations = A\B;
end
```

B

DATCOM input file

```
 $FLTCON NALPHA=10.0$
 $FLTCON NMACH=2.0,MACH(1)=0.26,0.27$
 $FLTCON ALSCHD(1)=-6.0,-4.0,-3.0,-2.5,-2.0,0.0,2.0,4.0,6.0,8.0$
 $FLTCON NALT=1.0,ALT(1)=0.0, WT=3074.8,LOOP=2.$
 $SYNTHS XCG=-0.782652,ZCG=0.003,XW=-1.6,ZW=0.0,ALIW=3.2,
    XH=4.85, ZH=1.0,ALIH=-3.25,XV=5.35,ZV=0.5$
 $BODY NX = 2.0,
   X(1)=-1.5,-0.5,
   R(1)=0.5,0.5,
   ZU(1)=0.5,0.5,
   ZL(1)=-0.5,-0.5$
 $WGPLNF CHRDTP=3.2,SSPNE=2.5,SSPN=2.5,
    CHRDR=3.2,SAVSI=0.0,CHSTAT=0.25, TWISTA=0.0,
    DHDADI=0.0,TYPE=1.0$
 $HTPLNF CHRDTP=0.3,SSPNE=0.5,SSPN=0.5,CHRDR=0.3,SAVSI=0.0,
    CHSTAT=1.0,TWISTA=0.0,TYPE=1.0$
 $VTPLNF CHRDTP=0.3,SSPNE=1.0,SSPN=1.0,CHRDR=0.35,SAVSI=0.0,
    CHSTAT=0.3,TWISTA=0.0,TYPE=1.0$
NACA-W-4-0012
NACA-H-4-0012
NACA-V-4-0012
DIM M
CASEID SKYHOGG WING-HORIZONTAL TAIL-VERTICAL TAIL CONFIG
DAMP
NEXT CASE
```

This corresponds to data presented in Table 3.1 with the incidence angle of the horizontal stabilizer set at -3.25°. The profile of the wing is set to NACA 0012. The engine/ballast is modeled as cylinder rather than cuboid. The resulting geometry is presented in Figure 4.13.

C

State Space Representations

The general equation for the state-space representation is provided by:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

Matrices A and B for cases introduced in section 3.3 are presented. They follow notation provided in Table C.1. Particularly interesting in the context of this work are rows denoted by x8 and x12 as these correspond to vertical acceleration (forces) and change of pitch rate (moment) respectively. By non-dimensionalizing these, one can obtain derivatives of aerodynamic coefficients with respect to state/input variables of interest.

Notation	**State**	**Notation**	**Input**
x1	x-coordinate in `World`	u1	Horizontal Stabilizer AoA
x2	y-coordinate in `World`	u2	Main Wing AoA
x3	z-coordinate in `World`	u3	Vertical Stabilizer AoA
x4	Roll angle		
x5	Yaw angle		
x6	Pitch angle		
x7	$\dot{x1}$		
x8	$\dot{x2}$		
x9	$\dot{x3}$		
x10	$\dot{x4}$		
x11	$\dot{x5}$		
x12	$\dot{x6}$		

Table C.1: Legend for state-space representations

C.1. Case 1 - no aerodynamic interaction

$$\begin{array}{ccccccc}
 & x1 & x2 & x3 & x4 & x5 & x6 \\
x1 & 0 & 0 & 0 & 0 & 0 & 0 \\
x2 & 0 & 0 & 0 & 0 & 0 & 0 \\
x3 & 0 & 0 & 0 & 0 & 0 & 0 \\
x4 & 0 & 0 & 0 & 0 & 0 & 0 \\
x5 & 0 & 0 & 0 & 0 & 0 & 0 \\
x6 & 0 & 0 & 0 & 0 & 0 & 0 \\
x7 & 0 & 0 & 0 & 0 & 5.14e-16 & 15.25 \\
x8 & 0 & 0 & 0 & 0 & -3.07e-16 & 1404 \\
x9 & 0 & 0 & 0 & 10.1 & -108.2 & -5.22e-13 \\
x10 & 0 & 0 & 0 & 3.24e-17 & -34.2 & -1.70e-13 \\
x11 & 0 & 0 & 0 & -1.30e-15 & -112.5 & -5.50e-13 \\
x12 & 0 & 0 & 0 & 0 & 4.10e-16 & -995.5
\end{array}$$

A =

$$\begin{array}{ccccccc}
 & x7 & x8 & x9 & x10 & x11 & x12 \\
x1 & 1 & 0 & 0 & 0 & 0 & 0 \\
x2 & 0 & 1 & 0 & 0 & 0 & 0 \\
x3 & 0 & 0 & 1 & 0 & 0 & 0 \\
x4 & 0 & 0 & 0 & 1 & 0 & 0 \\
x5 & 0 & 0 & 0 & 0 & 1 & 0 \\
x6 & 0 & 0 & 0 & 0 & 0 & 1 \\
x7 & -0.04352 & -0.2842 & -3.02e-18 & 4.88e-17 & -2.54e-31 & 1.37e-13 \\
x8 & 0.2282 & -15.74 & -3.44e-18 & 1.31e-15 & -2.89e-31 & 3.53e-12 \\
x9 & 2.51e-17 & -5.22e-13 & -1.213 & 3.09e-13 & -9.88e-14 & -5.20e-13 \\
x10 & 8.15e-18 & -1.69e-13 & -0.3833 & 1.00e-13 & -3.21e-14 & -1.69e-13 \\
x11 & 2.65e-17 & -5.50e-13 & -1.261 & 3.26e-13 & -1.04e-13 & -5.49e-13 \\
x12 & -0.004807 & 11.16 & 4.59e-18 & -1.50e-15 & 3.86e-31 & -4.01e-12
\end{array}$$

$$B = \begin{array}{cccc}
 & u1 & u2 & u3 \\
x1 & 0 & 0 & 0 \\
x2 & 0 & 0 & 0 \\
x3 & 0 & 0 & 0 \\
x4 & 0 & 0 & 0 \\
x5 & 0 & 0 & 0 \\
x6 & 0 & 0 & 0 \\
x7 & 4.887 & 10.42 & 7.46e-17 \\
x8 & 134.6 & 1267 & 5.67e-17 \\
x9 & -1.92e-20 & -2.17e-20 & 104.4 \\
x10 & -6.63e-21 & -1.71e-22 & 33.87 \\
x11 & -2.00e-20 & -1.07e-20 & 110.1 \\
x12 & -153.1 & -841.1 & -8.13e-17
\end{array}$$

C.2. Case 2 - downwash from main wing

$$
\begin{array}{c|cccccc}
 & x1 & x2 & x3 & x4 & x5 & x6 \\
x1 & 0 & 0 & 0 & 0 & 0 & 0 \\
x2 & 0 & 0 & 0 & 0 & 0 & 0 \\
x3 & 0 & 0 & 0 & 0 & 0 & 0 \\
x4 & 0 & 0 & 0 & 0 & 0 & 0 \\
x5 & 0 & 0 & 0 & 0 & 0 & 0 \\
x6 & 0 & 0 & 0 & 0 & 0 & 0 \\
x7 & 0 & 0 & 0 & 0 & -4.862e-17 & 13.59 \\
x8 & 0 & 0 & 0 & 0 & -1.307e-15 & 1348 \\
x9 & 0 & 0 & 0 & 10.21 & -108.2 & 5.215e-13 \\
x10 & 0 & 0 & 0 & 5.854e-17 & -34.18 & 1.691e-13 \\
x11 & 0 & 0 & 0 & 9.337e-16 & -112.5 & 5.502e-13 \\
x12 & 0 & 0 & 0 & 0 & 1.5e-15 & -931.6
\end{array}
$$

A =

$$
\begin{array}{c|cccccc}
 & x7 & x8 & x9 & x10 & x11 & x12 \\
x1 & 1 & 0 & 0 & 0 & 0 & 0 \\
x2 & 0 & 1 & 0 & 0 & 0 & 0 \\
x3 & 0 & 0 & 1 & 0 & 0 & 0 \\
x4 & 0 & 0 & 0 & 1 & 0 & 0 \\
x5 & 0 & 0 & 0 & 0 & 1 & 0 \\
x6 & 0 & 0 & 0 & 0 & 0 & 1 \\
x7 & -0.04324 & -0.2668 & -2.165e-22 & 5.351e-35 & 7.185e-35 & -1.248e-13 \\
x8 & 0.2307 & -15.11 & -2.312e-21 & 1.034e-34 & 1.388e-34 & -3.795e-12 \\
x9 & -5.215e-13 & -5.215e-13 & -1.213 & 4.339e-13 & 5.826e-13 & 5.229e-13 \\
x10 & -1.691e-13 & -1.691e-13 & -0.3831 & 1.407e-13 & 1.889e-13 & 1.696e-13 \\
x11 & -5.502e-13 & -5.502e-13 & -1.261 & 4.578e-13 & 6.146e-13 & 5.516e-13 \\
x12 & -0.006313 & 10.44 & 2.661e-21 & -1.297e-34 & -1.742e-34 & 4.311e-12
\end{array}
$$

$$
B= \begin{array}{c|ccc}
 & u1 & u2 & u3 \\
x1 & 0 & 0 & 0 \\
x2 & 0 & 0 & 0 \\
x3 & 0 & 0 & 0 \\
x4 & 0 & 0 & 0 \\
x5 & 0 & 0 & 0 \\
x6 & 0 & 0 & 0 \\
x7 & 4.753 & 8.938 & -2.051e-21 \\
x8 & 134.5 & 1210 & 5.743e-21 \\
x9 & 9.419e-20 & 1.518e-19 & 104.4 \\
x10 & 3.253e-20 & -1.225e-20 & 33.85 \\
x11 & 9.788e-20 & 5.292e-20 & 110.1 \\
x12 & -152.9 & -777.2 & -6.082e-21
\end{array}
$$

C.3. Case 3 - downwash from main wing and horizontal stabilizer

$$
A = \begin{array}{ccccccc}
 & x1 & x2 & x3 & x4 & x5 & x6 \\
x1 & 0 & 0 & 0 & 0 & 0 & 0 \\
x2 & 0 & 0 & 0 & 0 & 0 & 0 \\
x3 & 0 & 0 & 0 & 0 & 0 & 0 \\
x4 & 0 & 0 & 0 & 0 & 0 & 0 \\
x5 & 0 & 0 & 0 & 0 & 0 & 0 \\
x6 & 0 & 0 & 0 & 0 & 0 & 0 \\
x7 & 0 & 0 & 0 & 0 & -2.657e-14 & 13.46 \\
x8 & 0 & 0 & 0 & 0 & 1.105e-14 & 1348 \\
x9 & 0 & 0 & 0 & 10.24 & -108.2 & 5.028e-13 \\
x10 & 0 & 0 & 0 & 3.132e-17 & -34.17 & 2.009e-13 \\
x11 & 0 & 0 & 0 & -1.919e-15 & -112.5 & 5.184e-13 \\
x12 & 0 & 0 & 0 & 0 & -1.453e-14 & -931.7 \\
 & x7 & x8 & x9 & x10 & x11 & x12 \\
x1 & 1 & 0 & 0 & 0 & 0 & 0 \\
x2 & 0 & 1 & 0 & 0 & 0 & 0 \\
x3 & 0 & 0 & 1 & 0 & 0 & 0 \\
x4 & 0 & 0 & 0 & 1 & 0 & 0 \\
x5 & 0 & 0 & 0 & 0 & 1 & 0 \\
x6 & 0 & 0 & 0 & 0 & 0 & 1 \\
x7 & -0.04324 & -0.2657 & 9.699e-17 & -9.696e-17 & 4.848e-17 & -9.218e-14 \\
x8 & 0.2314 & -15.11 & 1.092e-16 & -2.614e-15 & 1.307e-15 & -2.457e-12 \\
x9 & 5.207e-13 & -5.212e-13 & -1.213 & -2.605e-13 & -1.864e-13 & 1.4e-15 \\
x10 & 1.688e-13 & -1.69e-13 & -0.383 & -8.445e-14 & -6.043e-14 & 4.54e-16 \\
x11 & 5.493e-13 & -5.499e-13 & -1.261 & -2.748e-13 & -1.966e-13 & 1.477e-15 \\
x12 & -0.006887 & 10.44 & -1.461e-16 & 2.999e-15 & -1.499e-15 & 2.791e-12
\end{array}
$$

$$
B= \begin{array}{cccc}
 & u1 & u2 & u3 \\
x1 & 0 & 0 & 0 \\
x2 & 0 & 0 & 0 \\
x3 & 0 & 0 & 0 \\
x4 & 0 & 0 & 0 \\
x5 & 0 & 0 & 0 \\
x6 & 0 & 0 & 0 \\
x7 & 4.756 & 8.83 & -2.392e-15 \\
x8 & 135.3 & 1210 & -1.703e-15 \\
x9 & 1.05e-18 & 6.182e-16 & 104.4 \\
x10 & 2.298e-19 & 8.995e-18 & 33.83 \\
x11 & 8.76e-19 & 3.11e-16 & 110.1 \\
x12 & -153.4 & -776.9 & 2.475e-15
\end{array}
$$

C.4. Case 4 - single rigid body

$$\begin{array}{ccccccc}
 & x1 & x2 & x3 & x4 & x5 & x6 \\
x1 & 0 & 0 & 0 & 0 & 0 & 0 \\
x2 & 0 & 0 & 0 & 0 & 0 & 0 \\
x3 & 0 & 0 & 0 & 0 & 0 & 0 \\
x4 & 0 & 0 & 0 & 0 & 0 & 0 \\
x5 & 0 & 0 & 0 & 0 & 0 & 0 \\
x6 & 0 & 0 & 0 & 0 & 0 & 0 \\
x7 & 1.62e-33 & 7.308e-34 & 7.68e-12 & 0.0005142 & -4.777e-14 & 2.494e-09 \\
x8 & 5.468e-22 & 2.467e-22 & 4.699e-10 & 0.03372 & -4.24e-10 & 860.6 \\
x9 & -9.915e-35 & -4.472e-35 & -3.009e-16 & 13.12 & 1.843e-16 & -1.56e-10 \\
x10 & 0 & 0 & 0 & 0 & -5.25e-16 & -3.281e-17 \\
x11 & -6.783e-38 & -3.059e-38 & -1.178e-25 & -7.25e-18 & 2.488e-17 & -2.049e-13 \\
x12 & -4.917e-22 & -2.218e-22 & -8.54e-10 & -0.05819 & 1.323e-12 & -1485
\end{array}$$

A =

$$\begin{array}{ccccccc}
 & x7 & x8 & x9 & x10 & x11 & x12 \\
x1 & 1 & 0 & 0 & 0 & 0 & 0 \\
x2 & 0 & 1 & 0 & 0 & 0 & 0 \\
x3 & 0 & 0 & 1 & 0 & 0 & 0 \\
x4 & 0 & 0 & 0 & 1 & 0 & 0 \\
x5 & 0 & 0 & 0 & 0 & 1 & 0 \\
x6 & 0 & 0 & 0 & 0 & 0 & 1 \\
x7 & 6.656e-13 & -0.1167 & -1.341e-14 & -2.772e-08 & -5.587e-14 & 1.037e-10 \\
x8 & 0.2334 & -7.654 & -9.146e-06 & -1.818e-06 & -5.545e-08 & 36.01 \\
x9 & -4.232e-14 & 4.573e-06 & 2.183e-18 & 1.086e-12 & 2.199e-18 & -6.529e-12 \\
x10 & 0 & 0 & 0 & 9.483e-15 & -4.479e-15 & -4.363e-11 \\
x11 & -1.448e-17 & 1.822e-15 & 5.672e-22 & -5.031e-16 & 2.154e-16 & -7.807e-11 \\
x12 & -0.1049 & 13.21 & 4.112e-06 & 3.137e-06 & 2.507e-08 & -368.3
\end{array}$$

$$\mathrm{B} = \begin{array}{cc}
 & \text{Angle of Attack} \\
x1 & 0 \\
x2 & 0 \\
x3 & 0 \\
x4 & 0 \\
x5 & 0 \\
x6 & 0 \\
x7 & 2.55e-09 \\
x8 & 860.6 \\
x9 & -1.56e-10 \\
x10 & 0 \\
x11 & -2.049e-13 \\
x12 & -1485
\end{array}$$

D Recommendations for KBE Approach

The creation of the simulation should be based on an input file. The file should include information about environment such as the model of atmosphere that should be included in the simulation. In this file, the list of components with their mass and moments of inertia tensor should be provided. The position of each of the components should be included as well. The easiest way to implement this would be to use a master reference frame. If the aim of the software is to be more user-friendly, the specification of an auxiliary reference frame (and their definition in the parent component) should be allowed. This has already been implemented in the FLAPERON framework. This part of the input file should be used to set up all of the components as `General Variable Mass` and `Rigid Transformations` within Simscape.

Additionally, each of the components to which external force should be applied should include a definition of that force. It can be constant, as is the case for the propulsion in this project. Additionally, the aerodynamic properties should be allowed to be set in the form of a lookup table or derivatives of the coefficients. These components can be included in the Simscape simulation with the help of `Lifting Surface` component developed as part of this work.

For the aerodynamic components, the interactions should be allowed to be specified. In particular, each of these components should be able to get a list of components that should be included when calculating induced velocity at this component. This could also be used to model the shape of the vortex filaments of each of the surfaces. Based on these inputs, the reference frames defining vortex filaments can be placed in Simscape environment along with `Downwash Calculator` developed in this work to provide each of the aerodynamic components with a signal of the induced velocity.

An example of the simplified input file is provided below in Listing D.1. Based on this input file the Simscape model file should be created.

Listing D.1: Example of input file

```
environment.atmosphere = "ISA";
environment.acceleration = [0, -9.81, 0];

%% Example for wing
wing = LiftingLineWing("XPROP Wing");
wing.isMirrored = true;

%% Geometry
wing.span = 2*203.2 * 1e-3; % m

for i = numel(airfoils):-1:1
    y(i) = section_data(i).r * 1e-3;
    twist(i) = -section_data(i).twist * pi/180;
    sweep(i) = section_data(i).sweep * pi/180;
    dihedral(i) = section_data(i).dihedral * pi/180;
end
wing.airfs = airfoils;
wing.set_airf_base(wing.airfs, y, twist, sweep,dihedral)
wing.gen_elems("geom")

%% Mass and inertia
wing.add_rf("iner", RefFrame( ...
    "Aux Wing Iner", ...
    ["O at the centroid of the root airfoil";
    "X from the leading edge to the trailing edge of the root airfoil";
    "Y towards the tip of the wing"], ...
    Point([0 0 0], wing.base), ...
    Orien([0; pi/2; 0], "321", wing.base)));
wing.mass = 0.029;  % kg
wing.cg = Point([-0.056, 0.0, -0.106]*1e-3, wing.rf.iner); % m
wing.I = Tensor.from_symm( ...
    [5.42e-2, 1.60e-3, 5.42e-2], [-2.73e-4, -7.13e-4, -1.72e-4], ...
    wing.rf.iner); % kg*m^2

% Repeat for all lifting surfaces...

aircraft.aero_componenents = [wing, h_stab, v_stab];
aircraft.interactions = [[wing, h_stab],[h_stab, wing]];

% Set up system
Analysis = System();
Analysis.env = environment;
Analysis.plant = aircraft;

Analysis.to_slx();
```