Propeller Effects on Very Flexible Aircraft

by

Patricia Capistrano Teixeira

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2019

Doctoral Committee:

Professor Carlos E. S. Cesnik, Chair
Professor Peretz P. Friedmann
Professor Daniel J. Inman
Assoc. Professor Kevin J. Maki

Patricia Capistrano Teixeira

pct@umich.edu

ORCID iD: 0000-0003-4400-206X

To God, my family and friends.

ACKNOWLEDGEMENTS

All the journey that culminated in this dissertation would not have been possible without the vital support of people that I had the good fortune to meet and interact. Therefore, I could not finish without saying a special thank you for each one of them.

First, my advisor, Professor Carlos Cesnik: words would not be enough to express my gratitude for this unique opportunity. Thank you so much for believing in me, for your guidance and friendship. You can be sure you made an important difference in my life, and I'll be forever grateful.

A special acknowledgment also to my committee: thank you for accepting the invitation to participate in my defense and for your valuable time, feedback, and attention. It was an honor having each of you as a committee member.

Thank you so much to Puneet Singh, Markus Ritter, Cristina Riso and Alfonso del Carre for your important collaboration to this work and, mainly, your special friendship.

I would not be here without the help and inspiration of my family. Thank you so much to: my husband, Renato, for your super special company and essential support! Once upon a time, all of this was just a dream to us. Together, we made it a reality and are now expecting a precious gift: our first son, Gabriel.

To my little miracle: Gabriel. Even if you have not yet arrived into this world, you already inspired me so much, and gave me the strength to finish this dissertation.

To my father Crisóstomo, my mother Rosane and my siblings Cristiana, Suzana, Caio, Álvaro Vitor, and João Gustavo: thank you for the lovely memories we share, for all your inspiration and important support! You are a special treasure to me!

To my grandmother Nazaré (in memory) and my aunt Marielena: thank you so much for opening the doors of your house and your hearts in different phases of my life, for all your essential encouragement and support!

A very special thank you to the awesome friends I made here in the past and current A^{2}SRL lab. Pardon for not citing names; I am afraid to forget someone, as the list is extensive. But I need to say I am so grateful to have met you guys and my stay here was so much better because of you. Thank you for the unforgettable moments we shared and for your valuable friendship.

Finally, I would like to thank the Department of Aerospace Engineering and the Conselho Nacional de Pesquisa (CNPq) for their financial support.

TABLE OF CONTENTS

LIST OF FIGURES

Figures

LIST OF TABLES

Tables

LIST OF APPENDICES

ABSTRACT

High Altitude Long Endurance aircraft are unmanned air vehicles with the capability of performing long-duration flights and can be used in many applications, such as communication and data relay, Earth observation, etc. To achieve their higher aerodynamic performance, HALE aircraft are typically high-aspect-ratio configurations, resulting in a very flexible structure. Moreover, the typical low cruise speeds often require a propeller-engine combination for propulsion.

Although many studies have been done in propeller-wing interaction, propeller effects on very flexible aircraft (VFA) have not received any attention. State-of-the-art nonlinear aeroelastic frameworks lack complete propeller modeling and, instead, use concentrated forces to model its thrust.

This work aims to fill this gap by incorporating propeller aerodynamics and inertial effects into a coupled nonlinear aeroelastic-flight dynamics framework. For that, the University of Michigan's Nonlinear Aeroelastic Simulation Toolbox (UM/NAST) is enhanced with an Unsteady Vortex Lattice for the lifting surfaces and a Lifting Line and Viscous Vortex Particle (LL/VVPM) methods to model the propeller aerodynamics. Furthermore, inertia effects associated with the rotating rigid blades are also incorporated.

Verification tests are performed for each of the new components added to the enhanced framework. Results for static and dynamic aeroelastic analysis with the coupled UM/NAST and Vortex Lattice model, isolated propeller, propeller-wing interaction, and the gyroscopic loads calculation are compared with results from other codes or published numerical and experimental data available in the literature.

Additionally, an approach based on system identification and proper orthogonal decomposition is introduced and verified for the stability analysis of VFA with propellers based on a time-series signal. The method is successfully verified against UM/NAST modal analysis for a purely structural case.

Based on the new developments, investigation of propeller effects on the aeroelastic response and stability of a VFA is presented and discussed. The results showed that the presence of propellers can influence the aeroelastic static and dynamic response of a VFA, as well as modify the aeroelastic modes and affect the flutter onset. Therefore, the additional propeller effects (besides just the incorporation of thrust) should be included in aeroelastic simulations.

Although focused on very flexible aircraft applications, it should be noted that the new enhanced framework can also be used to analyze new propeller-driven aircraft concepts such as the ones being proposed for Urban Air Mobility and distributed propulsion configurations.

CHAPTER 1

Introduction

The first chapter of this dissertation provides an introduction into the propeller-airframe interaction and how it has been modeled. First, motivation is provided in Section 1.1. Then, a literature review of selected topics relevant to this work is presented in Section 1.2, followed by a summary of the main conclusions. Finally, in Section 1.3, the dissertation objectives and a brief description of its organization are presented.

1.1 Motivation

In recent decades a new concept of aircraft has received increased attention due to its low energy consumption and promising applications: High-Altitude Long Endurance (HALE) aircraft. HALE aircraft is a class of unmanned air vehicle (UAV) with the capability of performing long-duration flights an can be used in satellite-like applications, for telecommunication and Earth observation.

In order to achieve their higher aerodynamic performance, HALE aircraft have high-aspect-ratio configurations and lightweight structures, reducing energy consumption. Therefore, they result in very flexible structural configurations characterized by large displacements. Such displacements cause a strong coupling between structural, aerodynamics, and flight dynamics and result in nonlinear behavior. For such very flexible configurations, conventional approaches for linear aeroelasticity are no longer applicable.

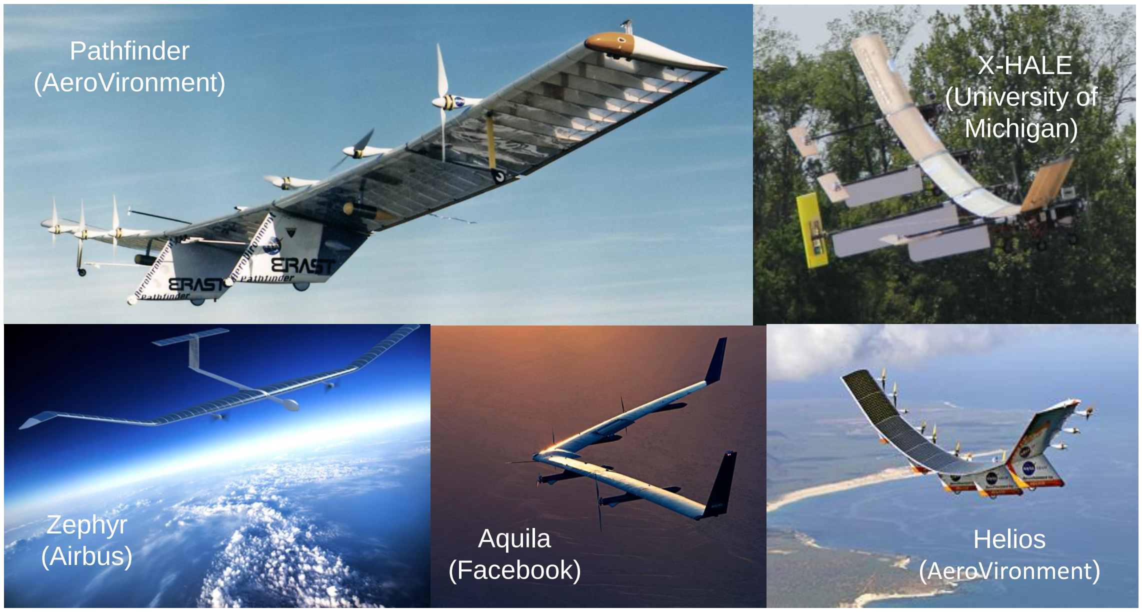


Figure 1.1: Examples of HALE aircraft: Pathfinder (AeroVironment), Zephyr (Airbus), Aquila (Facebook), Helios (AeroVironment) and X-HALE (University of Michigan). Propellers are usually a propulsion choice for such light and low speed configurations

Due to its high-efficiency goals and typical low cruise flight speeds, a propeller-engine combination is usually the propulsive solution of choice, as in the HALE aircraft illustrated in Figure 1.1. The presence of propellers results in different kinds of loads to an aircraft, as summarized in Figure 1.2:

i) aerodynamics loads, comprising of the resultant loads acting on the propeller hub (thrust, side and normal forces, torque, and yaw/pitching moments due to asymmetric blade loading, known as P-factor) and the influence of propeller slipstream on lift distribution of surfaces behind the propeller, and;

ii) inertial loads, which include the inertial effects due to propeller CG acceleration and the gyroscopic moments due to the variation of blades angular momentum when the aircraft performs a yaw or pitch motion.

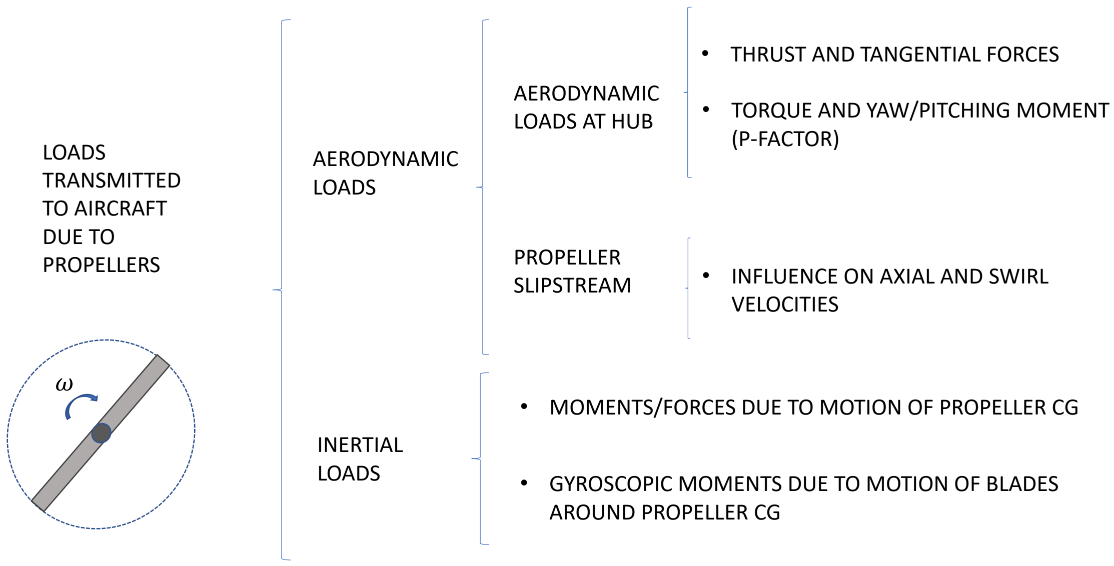

LOADS TRANSMITTED TO AIRCRAFT DUE TO PROPELLERS
ω
AERODYNAMIC LOADS
AERODYNAMIC LOADS AT HUB
THRUST AND TANGENTIAL FORCES
TORQUE AND YAW/PITCHING MOMENT (P-FACTOR)
PROPELLER SLIPSTREAM
INFLUENCE ON AXIAL AND SWIRL VELOCITIES
INERTIAL LOADS
MOMENTS/FORCES DUE TO MOTION OF PROPELLER CG
GYROSCOPIC MOMENTS DUE TO MOTION OF BLADES AROUND PROPELLER CG

Figure 1.2: Summary of propeller loads transmitted to aircraft

Despite the different loads contribution the propellers add, usually only thrust is incorporated in aeroelastic simulations of HALE aircraft, modeled as a point load, as in Jones and Cesnik [3], Ritter *et al.* [4], Hodges *et al.* [5], Shearer and Cesnik [6] and Changchuan *et al.* [7] and little has been investigated about the influence of the different propeller effects in the aeroelastic response and stability of such vehicles.

1.2 Literature Review

1.2.1 Propeller-Airframe Interaction for Rigid Configurations

1.2.1.1 Propeller-Wing Interaction

There are several studies that have considered propeller-wing interference for non-flexible wings. Early investigations on this topic started with the work of Prandtl [8]. Based on experiments exploring different relative positions between propeller and wing, Prandtl has identified two main kinds of propeller influences on the wing: an increase in the air velocity magnitude and a change of the velocity direction. Also, he noticed that the presence of a lifting surface influences the propeller, increasing or reducing thrust, depending on their relative position.

Other studies have also investigated the beneficial or detrimental outcomes of propeller-wing interaction and how different parameters, like geometry and the relative position between wing/propeller, determine the net effects. In Witkowski *et al.* [9] experimental and numerical investigations were performed focusing on the time-averaged performance of the propeller-wing combination in a tractor configuration. The results for an untapered, semispan wing and a Purdue 2-bladed propeller model indicated that the influence of the propeller can increase significantly the wing performance, causing drag reduction and lift augmentation. The numerical methods employed, a semi-empirical approach and a Vortex Lattice method (VLM) for both wing and propeller, showed a good ability to calculate performance trends, with more accurate results when using VLM.

Miranda and Brennan [10] studied how wing-tip-mounted propellers influence the wing performance. Using an analytical approach based on lifting line theory for wing and vorticity tube model to represent the propeller slipstream effects, it was identified that the wing-tip propeller can provide benefits in two ways: reducing the induced drag, for a propeller mounted in front of the wing (tractor configuration), or by increasing propeller propulsive efficiency when it is located behind the wing (pusher configuration). In both ways, the power necessary for a given flight condition is decreased. Also, it was observed that the direction of the propeller rotation determines if the interaction with the wing vortical structure is beneficial or detrimental. A performance benefit was achieved if the rotation of the wing-tip-mounted propeller is in opposite orientation to the vortices of the wing tip.

Kroo [11] used a generalized version of Munk's stagger theorem to assess propeller-wing combinations in inviscid incompressible flow. The lift distribution for maximum overall efficiency was found to be significantly different from elliptical loading. Also, due to swirl recovery caused by the presence of the wing, increments in net propeller efficiency were observed. This influence of wing on propeller performance was also studied by Marreta [12]. Using a free-wake analysis (FWA) approach for the propeller and Prandtl theory for the wing, different wing planforms with a propeller located in front were analyzed. The results showed a sensitivity of propeller thrust, power, and efficiency to the inflow condition, blade pitch, and wing planform. For example, an increase in angle of attack was found to cause an increase in thrust and power coefficient, with higher variations for a straight wing than for an elliptical one.

A comprehensive study considering the influence of propeller location on the wing aerodynamic performance is presented by Veldhuis [13]. For that, experimental investigation and also different numerical modeling approaches for propeller-wing aerodynamics were employed: empirical momentum theory, Vortex Lattice method coupled with blade-element propeller model, panel method (PM) with a slipstream envelope model, and RANS simulations. A swirl recovery factor (SRF) was considered for the VLM method to consider the reduction of the propeller slipstream velocity due to the presence of the wing. Also, the influence of both the propeller on the wing and the wing on the propeller were considered. This coupling was usually neglected by previous studies. Among the conclusions, it was noticed that propeller inboard-up rotation is beneficial to increase lift/drag ratio, especially when close to the wing tip. A negative propeller inclination with relation to the wing smaller than 15 degrees and high positioned propellers were also proved to be beneficial, although there is an increase in drag for the latter case.

Ananda *et al.* [14] studied the influence of wing aspect ratio on the propeller-wing problem motived by some discrepancies found in previous work in the literature. Based on the experimental investigation of rectangular flat-plate wings with different aspect ratios and at different flow conditions, it was observed a significant stall delay and maximum lift increase due to the induced effects of the propellers. Those improvements were found to be related to propeller diameter over wing span ratio.

In Mishra *et al.* [15] a study is presented comparing RANS-based computational fluid dynamics and a low fidelity model based on panel method for wings and an adaptation of extended blade element theory. The latter needs some parameter calibration that can be conducted based on CFD or experimental results. Comparisons for wing-vortex interaction on an isolated propeller and on a wing-mounted propeller showed good qualitative agreement, although some quantitative discrepancies were found with CFD and experimental data.

In Calabretta [16], a combination of PM for the airframe and actuator disk (a pseudo-steady model) and Vortex Particle to model the propeller and the propeller wake was implemented and proposed for early design optimizations. Singh and Friedmann [17,18], Thepvongs *et al.* [19], and He and Zhao [20] also employed Viscous Vortex Particles associated to some other model for the blades to model rotorcraft blades and/or propellers dynamics. As commented in Willis [21] and He and Zhao [20], some of the advantages of Viscous Vortex Particles are their formulation in Lagrangian approach (no need of grid), the possibility to model wake diffusion without problems with numerical dissipation, and a more natural development of the wake, avoiding singularities due to intersections between wake and lifting surfaces.

1.2.1.2 Distributed Propulsion Concepts

Based on the potential improvements identified in previous studies on propeller-wing interaction, recent ones have also investigated the use of distributed electric-driven propellers for the improvement of wing performance, as in Borer *et al.* [22], Borer and Moore [23], Ortun [24], and Fisher and Ortun [25].

In Borer *et al.* [22] a methodology was developed and used to explore the design space of a fixed-pitch propeller, focusing on lift augmentation and the ability to produce significant thrust at takeoff. This methodology was extended by Borer and Moore [23], allowing a significant increase in design freedom, and a probabilistic Compromise Ranking approach [26] was used to explore the high-lift propeller tradespace.

Recently, Ortun [24] developed a methodology to solve the flowfield of wings with multiple propellers in an intermediate way between low and high fidelity models. For that, a RANS solution for the airframe was coupled to a lifting line model for the propeller, whose effects were incorporated in the CFD simulations of the wing by adding source terms at selected cells inside the volume swept by the rotating blades. The method was applied in Fisher and Ortun [25] to investigate a wing-tip mounted propeller and also distributed propeller configurations. Propeller located at leading edge achieved up to 60% of lift increase and up to 107% for over the wing propellers. The use of a higher number of propellers showed to result in less power required for the same condition. Also, counter-rotating propellers were found to be more efficient.

Also recently, Alvarez and Ning [27], investigated the use of viscous vortex particle method (VVPM) for the interactional aerodynamics in distributed propeller concepts. They simulated multiple propellers configurations and demonstrated that the VVPM method is fully able to characterize induced velocities across the entire wake, from stable region, to instability transition and then unstable region. Propellers/wing interaction was not investigated.

1.2.1.3 Propeller-Aircraft Interaction

Jamison [28], Chen *et al.* [29], and He and Rajmohan [30] are examples of studies considering the effect of propeller in the whole aircraft.

In Jamison [28], flight tests were performed with the E-2C airplane for two different propeller models: Hamilton-Sundstrand model 54460-1 and model NP2000. The results showed that substituting one propeller by the other, with all other test conditions kept the same, influences the static longitudinal aircraft stability.

Using the method of Multiple Reference Frame for the calculation of propeller effects, Chen *et al.* [29] studied the slisptream interference of propellers on a rigid HALE aircraft. The investigation concluded that a higher propeller influence occurs at taking off, reducing at climbing, and becoming smallest at cruising. The influence increases with the increase of propeller thrust.

He and Rajmohan [30] studied the mutual interaction between a propeller and the compound rotorcraft. For that a hybrid method consisting of a CFD applied at regions near-body was coupled to a Viscous Vortex approch for modeling off-body wake. Among the conclusions based on the study of a compound rotorcraft, a reduction of wing lift due to propellers was observed. Also, fuselage blockage on the rotor wake was found to be significant in low speeds, while fuselage effect on the tail is less important than those of rotor and wing.

1.2.2 Propeller-Airframe Interaction for Flexible Configurations

Although the problem of propeller-wing interaction itself has been under investigation for some time, as shown above, the same is not true for how the propeller-wing interaction influences the static and dynamic response and stability of a very flexible aircraft.

In Jones and Cesnik [3], preliminary experimental results from the X-HALE flight test were compared with numerical results from the University of Michigan's Nonlinear Aeroelastic Simulation Toolbox (UM/NAST) framework [6,31,32], where thrust is also modeled as a concentrated follower force. As the results show, some significant differences were observed (as a soft Dutch-roll-like behavior instead of high amplitudes for roll and pitch predicted by numerical simulations). As pointed out by the authors, there are many possible explanations for the differences observed, and the lack of a propeller model may be responsible for some of them.

Among the works which have considered other propeller effects (besides thrust) in flexible aircraft one can find the experimental work of Gamble and Reeder [33], the numerical work of Agostinelli *et al.* [34], the experimental and numerical work of Cravana *et al.* [35], all of them applied for aeroelastic problems in the linear regime.

Gamble and Reeder [33] performed experiments with a flexible wing micro-air-vehicle and concluded that the relative position between the propeller and the wing influences yaw and pitch stabilities. In Cravana *et al.* [35] several configurations of flexible wing plus engine-propellers were considered in order to investigate the influence of motor position and spinning propellers on modal frequencies of the wing. Studies included cases of a clean wing, wing with pods only, and wing with pods plus spinning propellers (with different numbers of pods/propellers). The study demonstrated that edge-wise and torsion frequencies are significantly affected by mass/loads distribution, as well as the number of stores and their spanwise positions.

The study considering propellers used a small radius with a low thrust propeller. No measurable difference was noticed due to propellers, but as discussed by the authors, this may be caused due to the small amount of thrust produced by the propellers. A bending frequency increase was found for configurations of engines positioned at 20 cm from the wing root. The dynamic seemed to be also conditioned by the propeller's thrust, which could possibly cause a coupling effect between torsion and bending.

Agostinelli *et al.* [34] proposed a simplified approach for conceptual design using a pre-determined database (from CFD simulations or experimental results) of sectional wing aerodynamic coefficients coupled to a blade element theory to model propeller slipstream and a lifting line approach to propagate the effects along the wing spanwise direction. A beam finite element model with a linear force-displacement relation was employed for the structural model. The study focused on the effects of the propeller on velocity and lift distributions along the wing and the influence on the dynamic behavior was not investigated.

1.2.3 Propeller Influence on Flight Dynamics Stability

Early studies associated with the modeling and investigation of propeller effects in aircraft stability date from the first decades of the twentieth century. In Lanchester [36], from 1917, a whole appendix is dedicated to notes about the effects of propulsion on a flying machine. Harris [37] (1918) develops a mathematical formulation for side forces acting on a propeller as a response to yaw and Glauert [38], [39] (1919 and 1935) extends that formulation to derive other stability derivatives associated with the propeller.

Following those early investigations, in the work of Katzoff [40] experimental data obtained on the NACA full-scale wind tunnel for eight different airplane configurations were investigated in terms of the effects of the propeller in longitudinal stability and control. Among the conclusions, it was noticed that the loss of elevator effectiveness at high angles of attack can be largely eliminated by the inclusion of propeller. Also, the rate of increase of effective downwash angle with the angle of attack can be considerably increased due to the influence of propellers for certain aircraft configurations, as gull-wing and parasol-wing monoplanes. Ribner [41] extended some previous formulations to determine the side force of propeller due to yaw by incorporating induction effects and determines an expression for the side forces based on an analogy with fins, with the effective fin area taken as the lateral projected area of the propeller plane. It was observed that besides the side force, a single propeller in yaw also experiences a pitching moment. A dual-rotating propeller develops up to one third more side force than a single one. Also, the side forces due to angular velocities of pitch or yaw were found to be negligible for typical angular velocities that can be realized in maneuvers, with exception of spin.

In Butler *et al.* the slipstream effects in a v/stol aircraft performance and stability was studied. For that, an analytical investigation was proposed and correlated to experimental data. Following a performance investigation, a study on the slipstream influence on stability and control characteristics as well as the feasibility of use slipstream for stability augmentation. Preliminary results indicated that slipstream can be used as a potential solution for improving the dynamic stability of a tilt-wing aircraft in hovering flight.

Although one century has passed since the early studies, due to the complexity of the problem and new tools available, the investigation and modeling of propeller influence on aircraft stability are in continuous progress along recent years. Some examples of more recent efforts in this topic are the works of Jamison [28], Goraj and Cichocka [42] and Bouquet [43]. In Jamison [28], flight tests were performed with the E-2C airplane for two different propeller models, Hamilton-Sundstrand model 54460-1 and model NP2000. The results pointed that substituting one propeller by the other, with all other test conditions kept the same, influences significantly the static longitudinal aircraft stability. Goraj and Cichocka [42] studied the influence of gyroscopic effects on the stability of a light aircraft. Two types of gyroscopic effects were distinguished: weak gyroscopic effects, corresponding to maneuvers with small changes in pitch and yaw rates (classical dynamic stability) and strong gyroscopic effects, corresponding to rapid maneuvers, with substantial pitching and yaw rates. In this last case, it was found that coupling between lateral and longitudinal degrees of freedom can be significant, potentially causing loss of control.

The influence of propeller slipstream is investigated in the work of Bouquet [43], whose focus was developing, implement and validate a prediction method for the effects of the propeller on longitudinal stability. The implemented method is based on Obert's method, which was found to be computationally inexpensive and with relatively accurate results. It was observed four major effects caused by propeller slipstream: on the longitudinal stability: an additional normal force at propeller disk, the influence of slipstream on the wing, influence in tail-off pitching moment and change in tail contribution to the pitching moment. In the case of Fokker 50, It was found a decrease in tail effectiveness due to increase in downwash angle at the tail with the presence of propellers.

1.2.4 Propeller Influence on Aeroelastic Stability

In the case of very flexible aircraft, previous works like Hodges *et al.* [5], Feldt and Herrmann [44] and Quanlong *et al.* [45] have demonstrated that follower thrust has an important influence on the aeroelastic stability.

In Hodges *et al.* [5], the effects of thrust on the bending-torsion flutter of very flexible wings were investigated. For this purpose, thrust was modeled as a follower force with a prescribed magnitude. Propeller gyroscopic and slipstream effects were not included. A nonlinear mixed finite element method was used to the structural model of the wing, represented by beams, and a finite-state two-dimensional unsteady aerodynamic approach was used to model wing aerodynamics. Their results suggest that thrust has either a stabilizing or destabilizing effect depending on the ratio, λ, of bending stiffness to torsional stiffness. If $\lambda < 5$, it was observed that an increase in thrust, up to a certain value, increases the flutter speed. For $\lambda > 10$, thrust contributes to decrease flutter speed. Differences up to 11% in flutter speed were observed, pointing to the importance of considering engine thrust influence on wing flutter of very flexible aircraft. While the thrust as a follower force can be included in the analysis relatively easily, the lack of propeller aerodynamic and inertial effects on the free flight of very flexible aircraft may be a potential source of errors in simulations.

In Feldt and Herrmann [44], the bending-torsional flutter of a cantilevered wing with a lumped mass on its tip was investigated. It was found that the follower forces contributed to reducing the critical speed, while an increase in the tip mass demonstrated a stabilizing effect. Quanlong *et al.* [45] also concluded that the presence of thrust, modeled as a follower force, reduces the flutter limit. In the case considered, a reduction of flutter speed in more than 10% was noted.

Studies of other propeller effects besides thrust on the aeroelastic stability have been limited to problems involving displacements inside the linear regime, as in Rezaeian [46], Sui An [47] and Guruswamy [48].

In Rezaeian [46] a numerical investigation of the two main instability phenomena associated to a propeller-nacelle-wing system was performed using the software ZAERO. The influence of propeller on modal damping was studied, including slipstream and gyroscopic modeling, and it was observed that gyroscopic effects have influenced modal damping of the wind tunnel wing.

The development of a gradient-based aeroelastic optimization considering propeller influence is presented in Sui An [47]. The framework employs Double Lattice method for surfaces aerodynamics, actuator disk to model the average effects of the propeller (one-way coupling), and uses the Toolkit for the Analysis of Composite Structures (TACS) for the structural analysis. Results indicated that despite the improvement expected for aerodynamic efficiency, an increase in aspect ratio and in the number of propellers can make the structure more vulnerable to instabilities and structural failures. The effect back of wing on propellers, gyroscopic moments and nonlinearities for structure and aerodynamics were not included in that model. Also, aeroelastic static and dynamic considerations (in terms of elastic and rigid body displacements) were not the focus of that investigation.

In recent work, Guruswamy [48] investigated the influence of a wingtip mounted propeller on the aeroelastic stability of a flexible aircraft with aspect ratio five. The flow-field is simulated based on Navier-Stokes equations, while the structure is modeled using plate finite elements. Results for the wing model with and without the tip propeller showed that propeller caused destabilizing effects for the same freestream condition. However, increasing the propeller RPM, a stable response could be achieved.

1.2.5 Traditional Simplified Propeller Aerodynamics

Although CFD methods are capable of capturing complex phenomena, they are still associated with a high computational cost, besides the extra time associated with the mesh generation. In the context of VFA simulations, involving the simultaneous solution of multiple disciplines for a model with deforming surface and often multiple propellers, such high fidelity approaches can be prohibitive.

Alternatively to CFD, simplified approaches have been developed. Methods based on blade element model are certainly one of the most common. In Gur and Rosen [49], typical methods based on blade element model, using different approaches for the calculation of induced velocities, were compared for an isolated propeller: momentum, simplified momentum (traditional actuator disk), lifting-line (prescribed, semi-prescribed and free-wake) and vortex approach (McCormick and Theodorsen). The results indicated that for the axial flight of straight blades (or small sweep) propellers, the simplified-momentum model showed good quality results at a low computational cost. They also indicate that for cases such as static operation, the induced velocities due to propeller slipstream are higher than for typical cruise conditions and a more accurate model may be necessary, as, for example, lifting line model with free-wake.

Abedi *et al.* [50] also compared different aerodynamics methodologies for an isolated wind turbine: lifting line prescribed wake, lifting surface prescribed wake and panel method prescribed wake. The methods were compared against results from GENUVAP, an unsteady flow solver based on vortex blob approximations. Lifting Line showed to be an advantageous method due to its short computational time and good agreement with GENUVAP.

Alternatives consisting of variations of the cited methods or their combination are also available. Khan and Nahon [51], for example, combined analytical and semi-empirical equations, some of them based on marine propellers, to define a slipstream model which considers both acceleration and diffusion. The method showed good agreement with experiments but depended on pre-determined coefficients. Wang [52] *et al.* proposed a regionalized actuator disk approach, with parameters varying in radial and circumferential direction along the disk. The new model showed better performance than classical Actuator Disk [53].

Additionally, empirical methods are also available, as the empirical methods of Smelt and Davies [54] and Kuhn [55]. Although the low cost and capability of predict important effects, like viscosity and separation, they are restricted to the limited set and conditions of experimental data they are based on and do not account for mutual iterations between the propeller and lifting surfaces.

Table 1.1: Classical Aerodynamic Approaches for Propeller Blades Modelling

Method	**Main Characteristics**
Blade Element (BE) Model [49,56]	Blade is divided into segments, which behave like 2D wings; assumes absence of interaction between neighbor elements; aerodynamic coefficients are obtained from a database.
Lifting Line (LL) [57,58]	Blade divided into segments, replaced with a straight vortex filament with constant strength located at 1/4 chord; takes into account mutual influence between different sections and lift loss at blade tip.
Vortex Lattice method (VLM) [57,58]	Blades are modelled as a panelled surface (thickness neglected) and condition of zero normal velocity is imposed.
Panel method (PM) [57,58]	Similar to VLM, but thickness is taken into account, with panels defined over the actual surface.

As the most expensive part of the propeller aerodynamics approaches is usually the calculation of its wake induced velocities, it is a common practice to associate different approaches for the blade and the wake. Tables 1 and 2 present examples of

Table 1.2: Classical Aerodynamic Approaches for Propeller Slipstream

Method	Main Characteristics
General Momentum Model [39,49]	Based on actuator disk approach; flowfield is divided into concentric annuli control volumes.
Simplified Momentum Model [39,49]	Classical Actuator Disk; Additional assumptions: far wake pressure approximately equal to the ambient pressure; circumferential induced velocity right after disk is similar to the one in the far wake.
Lifting-Line Model [59]	Based on Prandtl Lifting Line Theory (prescribed, semi-prescribed or free-wake).
Vortex Models [60,61]	Based on optimal distribution of blade's circulation and Kutta-Joukowski theorem (e.g. Theodorsen, McCormick).
Empirical methods [54,55]	Based on a set of experimental data and accounts for effects like viscosity and separation; do not account for mutual aerodynamic iterations.

traditional methodologies for aerodynamic modeling of blades and slipstream.

1.2.6 Propeller Modeling on Aeroelastic Frameworks for VFA

Due to its large deflections under typical loads conditions, VFA is characterized by nonlinear behavior. However, traditional aeroelastic frameworks make use of linear theory for structural and aerodynamics formulations, and currently, commercial tools for the analysis of VFA aircraft are not available [62].

Separate research group efforts have developed aeroelastic framework capable of taking into account the nonlinear aspects of the VFA behavior. Examples of state of the art codes in this context cited on literature are UM/NAST [6, 31, 32], from University of Michigan, SHARPy [63–66] from London Imperial College, ASWING [67–69], from Massachusetts Institute of Technology, NATASHA [70,71], from Georgia Institute of Technology, and DLR toolbox [4, 72], from Deutsches Zentrum für Luft-

und Raumfahrt (DLR). Each of those codes has differences and similarities about the formulations employed, and the focus here is given on the propeller modeling.

Concerning the modeling of the propeller effects previously described (summarized in Figure 1.2), usually, the aeroelastic frameworks for VFA consider the thrust effect only, modeled as a follower punctual force. Exceptions are, from the best of this author's knowledge, for ASWING and the current version of UM/NAST (with the enhancements from the present work). ASWING employs an actuator disk model and can model thrust, torque and P-factor loads (loads due to asymmetric loading of blades) [68, 69].

UM/NAST originally also incorporated just thrust to model the propeller effects. However, developments made by the present work [73–75] made possible the complete inclusion the propeller effects, as summarized in Figure 1.2.

1.2.7 Summary and Main Conclusions

From the literature review on the topics selected, its possible to conclude that:

- The study of propellers and airframe interaction date from a long time, and many studies with rigid configurations have demonstrated that propellers can significantly influence an aircraft aerodynamic performance, free-flight response, and stability;

- Few studies have been conducted on the effects of propellers on very flexible aircraft, and usually, just thrust, modeled as a punctual force, is incorporated in the simulation of such very flexible configurations;

- Aeroelastic studies have indicated that thrust can have a significant impact on flutter, reducing the predicted critical velocities in about 10% when compared to a model without thrust. Other propeller effects were not investigated;

- Concerning the aerodynamic modeling for propellers, high fidelity tools (CFD simulations) are still too expensive for the required multidisciplinary analysis and simulation of VFA. Alternative simplified approaches offer quicker evaluation, but many of them assume a pre-determined wake shape, isolated propeller configuration, and usually capture just averaged effects. Then, their applicability to VFA aircraft with several interaction surfaces and propellers may not be adequate;

- Finally, it was observed a lack of propeller modeling among the state-of-the-art codes available for the aeroelastic analysis of VFA.

1.3 Dissertation Objectives and Outline

1.3.1 Dissertation Objectives

In view of the lack of complete propeller modeling in nonlinear coupled aeroelastic-flight dynamics frameworks and the consequently limited investigation of its effects on VFA, this thesis has the following main goals:

- Develop and verify a coupled nonlinear aeroelastic-flight dynamics framework capable of taking into account the propeller loads transmitted to a very flexible aircraft (Figure 1.2);

- Investigate the effects of propellers on the aeroelastic response and stability of a HALE aircraft, focusing on three aspects:

  i) Identify importance and impact of including propeller effects in VFA modeling;

  ii) Identify the isolated contribution of the different loads transmitted by the propeller;

iii) Get an insight about if/how flexibility may change the expected behavior of the propeller-wing interaction when compared to a rigid case.

In order to accomplish these objectives, the UM/NAST framework was enhanced by coupling it with an Unsteady Vortex Lattice code for the aircraft lifting surfaces and a Lifting Line/Viscous Vortex Particle Method for modeling the propeller aerodynamics. Gyroscopic effects were derived and incorporated. Each new piece added was verified against numerical or experimental data. Also, an alternative approach for stability analysis of very flexible aircraft was developed and verified. The framework was then applied for the investigations of propeller influence on a representative HALE aircraft.

1.3.2 Dissertation Outline

This dissertation is structured as follows. Chapter 2 presents details of the formulation of the new pieces added to the UM/NAST framework. First, an overview of the enhancements added to UM/NAST is presented. Next, a discussion about the motivation and description of theoretical aspects for each aerodynamic method selected (for lifting surfaces, blades, and propeller wake) is provided. The derivation of the inertial effects is described, and the interfaces between the different disciplines are discussed. Chapter 3 details the numerical implementation of the new developments in terms of architecture and workflow of its different parts. The objective is to clarify the solution process and provide essential information for future use and improvements. Chapter 4 presents the motivation and development of an alternative approach based on a combination of Proper Orthogonal Decomposition and System Identification for the extraction of dynamic information (frequency, damping, and modes) from aeroelastic simulations with propellers. Chapter 5 presents verification results for each new component added to the coupled aeroelastic-flight dynamics framework. Then, an investigation of the propeller effects on the aeroelastic response and stability of HALE aircraft is presented in Chapter 6. Finally, concluding remarks and key contributions of this work are summarized in Chapter 7, and ideas for future steps are provided.

CHAPTER 2

Coupled Nonlinear Aeroelastic-Flight Dynamics Framework with Propellers

The presence of propellers can affect the dynamics and stability of an aircraft in several ways. In the case of very flexible aircraft, most of these effects have been neglected in coupled aeroelastic-flight dynamic simulations, and only thrust was included to represent its effects. The exploration of their importance and impact on simulations has not been considered.

This chapter describes the theoretical foundations and developments towards a coupled nonlinear aeroelastic-flight dynamics framework capable of capturing the missing propeller effects. Section 2.1 introduces the University of Michigan's Nonlinear Aeroelastic Simulation Toolbox (UM/NAST) [6,31,32] used as the basis for this work and summarizes the enhancements added to it in order to include the propeller effects. Sections 2.2 to 2.6 describe in more details the various components added into the UM/NAST framework, describing the formulation for the lifting surface aerodynamics, propeller blades and wake aerodynamics, propeller inertial effects, and the interfacing between the different disciplines.

2.1 Framework Overview

The coupled nonlinear aeroelastic-flight dynamics framework is based on the University of Michigan's Nonlinear Aeroelastic Simulation Toolbox (UM/NAST), which was enhanced by this work to account for the additional propeller effects. A brief summary about key aspects of UM/NAST formulation and the enhancements added are described in the following subsections.

2.1.1 UM/NAST Framework

The University of Michigan's Nonlinear Aeroelastic Simulation Toolbox (UM/-NAST) [6, 31, 32] is a multidisciplinary tool developed for the simulation of very flexible aircraft. It features solutions for the modal characterization about different steady state conditions, static, trim and nonlinear transient simulations and aeroelastic stability analyses. For that, the equations for structural dynamics, nonlinear 6-DoF vehicle dynamics and aerodynamics are integrated simultaneously using three different numerical scheme options: trapezoidal, generalized alpha, and forward Euler schemes.

In order to enable the modeling, different frames of reference are defined [76]. A summary of the most important ones is illustrated in Figure 2.1. First, a global (inertial) frame G, fixed in the ground, is defined. Then, a body-attached frame B is built to describe the body position and orientation in inertial frame, with the unit vectors $B_x(t)$ pointing to the right wing, $B_y(t)$ pointing forward and $B_z(t)$ given by the cross product between $B_x(t)$ and $B_y(t)$. In order to describe the elastic deformations of each point on the aircraft with relation to body frame, a local beam frame W is attached to each node along the beam model, with the basis $W_x(s,t)$ pointing along reference beam axis towards right wing, $W_y(s,t)$ pointing towards leading edge, and $W_z(s,t)$ is normal to the beam surface. Here s refers to the curvilinear beam coordinate. Finally, a local 2D aerodynamic frame is also defined, with the A_y unit vector aligned with the chordwise direction of the airfoil, pointing upstream, and the A_z unit vector pointing vertical up. The local aerodynamic and beam frames are usually aligned, but they can become different in certain situations, as, for example, when the wing has a sweep angle.

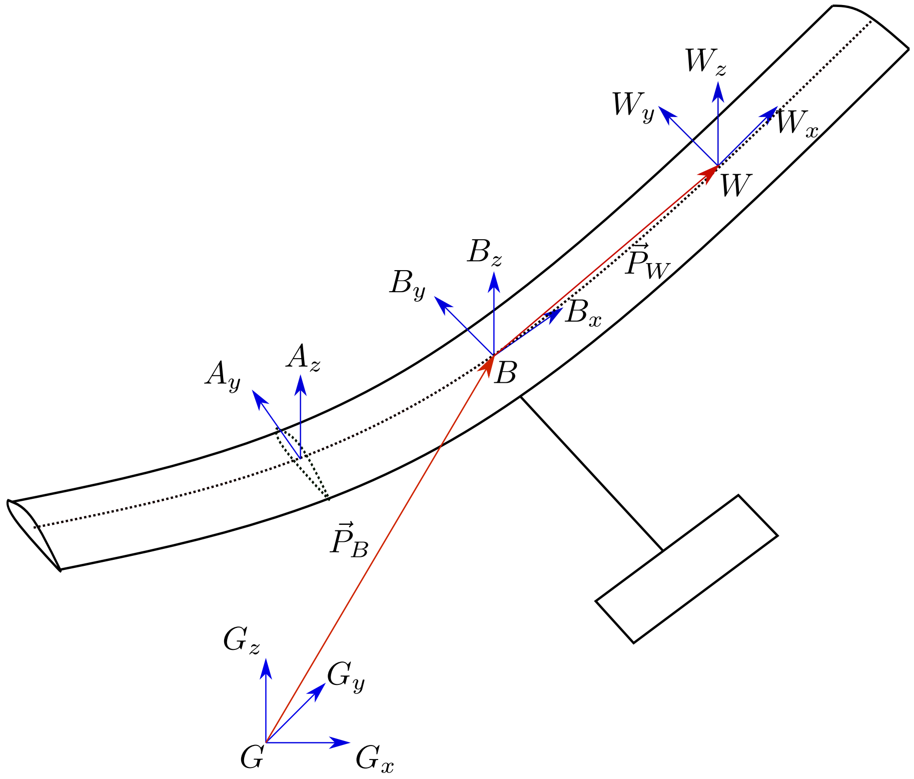


Figure 2.1: Basic UM/NAST reference frames

UM/NAST employs a geometrically nonlinear structural formulation using a strain-based finite element [77]. Within each element, constant strains in extension, twist, and in- and out-of-plane bending are assumed, resulting in a 4-degree-of-freedom element:

$$\epsilon_e = \{\epsilon_x, \kappa_x, \kappa_y, \kappa_z\} \tag{2.1}$$

The nonlinear equations of motion are solved in terms of those strain values, and the related displacements are post-processed. For each node along the beam, the position and orientation is defined by a vector h with 12 components: the first 3 corresponding to the position P_w of the node in body frame and the next 9 corresponding to the local unit vectors W_x, W_y and W_z also resolved in body frame.

$$h(s) = \left\{ P_w(s)^T, W_x(s)^T, W_y(s)^T, W_z(s)^T \right\} \tag{2.2}$$

The relation between the strains and displacements is provided by Cesnik and Brown [78] and given by the equation:

$$\partial h(s)/\partial s = A(\epsilon(s))h(s) \tag{2.3}$$

where A is a matrix which is a function of the strains.

Assuming constant strain for each element, the solution of Eq (2.3) is simplified to:

$$h(s) = e^{G(s)} h_0 \tag{2.4}$$

with h_0 being the vector containing position and orientation of a prescribed node at the root of the beam.

Originally, a potential-flow finite-state Peters' inflow aerodynamic theory [79, 80] with a correcting factor for 3D effects was implemented in UM/NAST [6, 31, 32]. That theory is based on the assumption of two-dimensional thin airfoil immersed in an inviscid and incompressible flow. The main advantage of strip theory is the minimal computational cost, even when compared to other potential methods, as Vortex Lattice. However, the dependency on an adequate reference lift distribution provided a priori (in order to correct for 3D effects), and the inability to take into account mutual influence between different lifting surfaces are the major drawbacks of the method.

The resulting system of equations, accounting for structural dynamics, nonlinear 6 DoF vehicle dynamics and aerodynamics, has the following format [6]:

$$M(\epsilon)\ddot{q} + C(\epsilon, \dot{\epsilon})\dot{q} + Kq = R \tag{2.5}$$

where M, C and K represents the generalized mass properties, damping and stiffness matrices, respectively; $q = (\epsilon, p_B, \Theta_B)$ is a set of generalized coordinates containing the strains associated with the flexible vehicle, ϵ, the body inertial position, p_B, and an arbitrary orientation vector, Θ_B, of the body frame of reference; and R represents generalized forces accounting by aerodynamics loads, gravity, applied loads, etc.

2.1.2 Enhancements to Include Propeller Effects

The original aerodynamic modeling in UM/NAST framework employed a corrected strip theory, whose solution accuracy depends on the proper choice of a lift distribution provided *a priori*. Besides that, the corrected strip theory method is not able to take into account the mutual aerodynamic influence of different lifting surfaces. However, in order to capture the interaction of propellers and aircraft lifting surfaces, a method capable of taking into account that mutual aerodynamic influence was necessary. For that purpose, the Unsteady Vortex Lattice method (UVLM) was selected and a UVLM code developed by Ritter [81] was adapted and integrated to UM/NAST framework, as an additional aerodynamic option. The UVLM code was originally written in Python with FORTRAN subroutines. In order for the framework to have a common language and more efficient integration of the different parts, it was later on re-written in C++, improving the performance and facilitating the integration with UM/NAST, which is written in C++.

As the structural model employs a 1D finite element representation and the UVLM uses 2D non-planar grid, an interface between the two solvers was implemented in order to transfer loads and displacements. As will be discussed in more details in a dedicated section, such mapping is based on the assumption of rigid behavior in the chordwise direction and on the principle of equivalence of virtual work between the forces at panels and forces transferred to beams.

After completing the integration between UVLM and UM/NAST, the aerodynamic modeling for the propellers, based on Lifting Line (LL) for the blades and Viscous Vortex Particle (VVPM) method for the wake, was implemented and integrated into the UVLM code. Again, as the UVLM was originally in Python, the propeller aerodynamic solver was first developed in Python and later on re-written to C++. A general C++ interface allowing the integration between different possibilities for the modeling of propeller and lifting surfaces aerodynamics was also developed, allowing future investigations with other combinations of methods.

Finally, the inertial effects of propellers were also included, incorporating the effects of propeller CG acceleration and the gyroscopic moments associated with the angular motion. For this, a derivation based on UM/NAST reference systems was developed, and the formulation was directly implemented into UM/NAST framework.

Figure 2.2 presents an overview of the enhanced coupled aeroelastic-flight dynamics framework, for the case of dynamic simulation. For each time step, based on current values of loads and control inputs, the new geometric configuration of the structure is determined. Beam coordinate deformations are then converted to panel grid point deformations, and aerodynamic loads are calculated with uVLM coupled to LL/VVPM. Those loads are then converted to concentrated loads at beam nodes, and the dynamic process continues until the predetermined simulation time is reached. It is important to note that the contribution of the elastic deformation of the body in the effective free-stream speed is considered in both uVLM and LL/VVPMs.

In the next sections, more details are provided about the theoretical aspects of each enhancement added. The numerical implementation, as well as a detailed verification

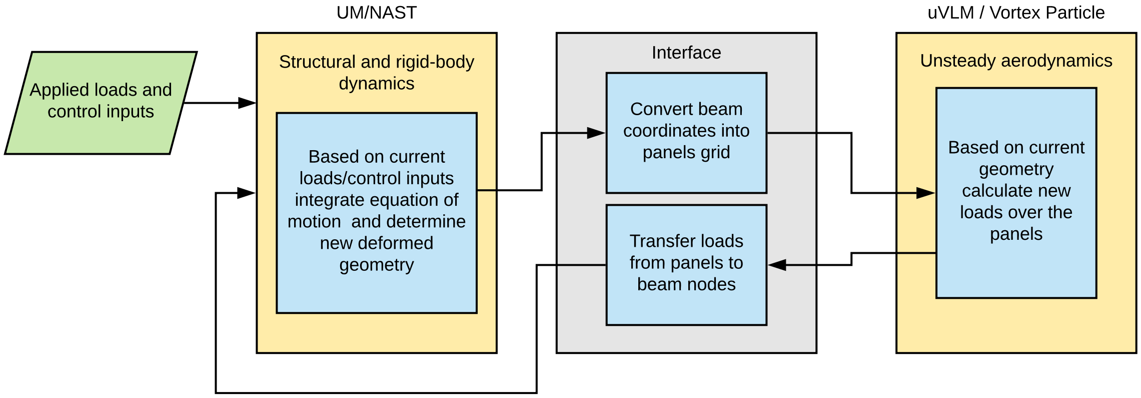

UM/NAST
uVLM / Vortex Particle
Applied loads and control inputs
Structural and rigid-body dynamics
Based on current loads/control inputs integrate equation of motion and determine new deformed geometry
Interface
Convert beam coordinates into panels grid
Transfer loads from panels to beam nodes
Unsteady aerodynamics
Based on current geometry calculate new loads over the panels

Figure 2.2: Enhanced coupled aeroelastic-flight dynamics framework with propeller effects

of the different parts of the enhanced framework, will be presented in the next chapter.

2.2 Lifting Surfaces Aerodynamics

2.2.1 Vortex Lattice vs. Strip Theory

As concluded in the investigation performed by Ritter *et al.* [82], although strip theory has been a popular method for the aeroelastic simulation of very flexible aircraft [5,6,31,83–86], its inability to take into account mutual lifting surfaces influence and other 3D effects limit its accuracy and the possible range of applications.

The reason for its extensive use may be related to the fact that earlier designs of VFA consisted mostly of high aspect ratio flying wings [82]. For such designs, 3D effects had smaller influence and, with the absence of other lifting surfaces, the lacking of modeling for mutual aerodynamic surfaces interaction was not affecting the results. Moreover, the implementation of strip theory is much simpler than other aerodynamic methods, strips are allowed to translate and rotate almost arbitrarily, and the computational cost is very low (one of the greatest advantages of strip theory). Such characteristics make it a very attractive model for the multidisciplinary coupled aeroelastic-flight dynamics framework.

For a more complex very flexible aircraft, however, with multiple lifting surfaces of varying aspect ratios moving with relation to each other, the limitations of strip theory become more critical. Although the rectangular distribution of lift provided by strip theory can be corrected by a reference lift distribution, this reference may not be representative along all the VFA maneuver considered. Also, the lacking of downwash modeling of the wing on tails, for example, plays an important role in the aeroelastic behavior of the whole aircraft and correction methods to include the varying downwash effect become very difficult to be applied in a general way.

Another class of potential method that could take into account the mutual influence between lifting surfaces and 3D effects are methods based on aerodynamic influence matrix (AIC), as Double Lattice and Vortex Lattice. While Double Lattice has been a standard tool for fixed-aircraft aeroelasticity, due to its faster calculation of the unsteady aerodynamic loads (compared to UVLM) , it is a linear method restricted to small out-of-plane motions with flat wake, being no longer valid for the novel highly flexible configurations [63]. Vortex Lattice has the advantage to be capable of modeling the aircraft undergoing large translations and rotations and has become a popular method in aeroelastic frameworks for the simulation of VFA [1,63,82,87–92]. Also, for conditions within the limitations and assumptions considered in its formulation, Vortex Lattice results demonstrate remarkable agreement with CFD and experimental data [93–95]. Its range of validity is typically the case of very flexible aircraft, whose usual velocities are within the incompressible regime, but with high enough Reynolds number to assume that the viscous effects are confined in a thin boundary layer, and whose typical sections shape can be considered as thin-airfoil configurations.

Based on the arguments presented above, and given the necessity to include mutual surfaces influence in order to take into account the propeller/lifting surfaces interaction, the Vortex Lattice method was chosen for the aerodynamic modeling of the aircraft surfaces, added as an additional option in UM/NAST. The major drawback of Vortex Lattice method is its quadratic dependence on the number of panels (over the surfaces and their wakes), resulting in a much slower computation than strip theory (but still much cheaper than unsteady RANS solution). Also, the original formulation does not consider viscous effects, but this can be added by an additional correction, as will be discussed soon. In the next section, more details about the formulation of the Vortex Lattice method used in this work are presented.

2.2.2 Vortex Lattice Method Formulation

The Vortex Lattice Method (VLM) is a mid-fidelity approach based on potential theory [57, 58] that allows the calculation of aerodynamic loads for steady and unsteady problems. It assumes thin-wing models under low speeds, but with high enough Reynolds Number to neglect viscous effects, and attached flow conditions [63].

Earlier formulations of the Vortex Lattice Method date from 1931, when Rosenhead [96] studied the replacement of a two-dimensional vortex-layer by a system of vortex filaments. In 1943, Falkner [97] introduced the term "Vortex Lattice." In 1965, Hedman [98] established the now classical steady Vortex Lattice, based on the placements of horseshoe singularities along a discretized wing. In a tentative to expand the method to unsteady cases, and assuming a flat wake, Albano and Rodden [99] replaced the vortex sheet by equivalent oscillating doublets, creating the Double-Lattice Method. Later it was demonstrated [100] that a panel with a piecewise constant double distribution is equivalent to a vortex ring around its periphery, opening the way for the extension of VLM to unsteady situations. Initial studies developing Unsteady Vortex Lattice (UVLM) were carried out by Belotserkovski [101], Rehbach [102] and groups at Virginia Tech [103, 104] and Technion [105, 106].

In the next subsections, a summary of the main aspects of the Vortex Lattice formulation that was applied to this work is presented. Although there are several references available in the literature, the reader is referred to the book of Katz and

Plotkin [57], which is one of the current most complete descriptions about the Vortex Lattice Method (named there as “Lifting-Surface Solution by Vortex Ring Elements”), for more details.

2.2.2.1 Main Assumptions and Basic Equations

Assuming potential theory, *i.e.*, low speeds such that the flow is incompressible ($Mach < 0.3$), but high enough Reynolds Number such that the fluid viscosity can be neglected and the fluid behaves as irrotational, it is possible to define a potential scalar function Φ whose gradient corresponds to the velocity field:

$$\vec{u} = \nabla\Phi \tag{2.6}$$

Also, for an incompressible fluid, the continuity equation can be written as:

$$\nabla \cdot \vec{u} = 0 \tag{2.7}$$

Then, substituting Eq. 2.6 into Eq. 2.8, will result in the Laplace equation:

$$\nabla \cdot \vec{u} = \nabla \cdot \nabla\Phi = \nabla^2\Phi = 0 \tag{2.8}$$

Laplace equation is an elliptic linear differential equation, and its solution can be written as a linear combination of functions that are individual solutions of that equation. Many possible singularity functions are a solution for the Laplace equation (*e.g.*, vortex, source, doublet). With the additional assumption of thin-lifting surfaces (such that thickness effects can be neglected), the idea of the Vortex Lattice method is to represent a thin-wing by its camber surface, discretizing that surface in quadrilateral elements (panels) and distributing singularity functions (which are solutions of Laplace equation) over those panels and over the wake. Such singularity elements model the effect of vorticity confined in the thin boundary layer near-surface and in the wake sheet.

In the classical, steady approach of Vortex Lattice, horseshoe vortex (composed by a straight bound vortex segment, which models the lifting properties, and two semi-infinite trailing vortex that model the wake) are typically used. For unsteady cases, however, the choice of vortex-ring, quadrilateral elements composed by discrete vortex segments in a closed-loop, is more appropriate, as both, lifting surfaces and wake can be discretized using those elements, making wake shedding and, consequently, time-domain simulation of Vortex Lattice possible [63, 107]. Additionally, the vortex-ring elements allow for the boundary conditions to be specified at the actual camber surface, which can have camber and several planform shapes [57]. Then, in this work, both steady and unsteady solutions are modeled using vortex-ring elements.

Once the solution is written in terms of the added effects of the vortex-ring elements it is necessary to satisfy the boundary conditions, which, for this case, is the zero normal flow across the surface (non-penetration condition) and velocity going to zero at infinity:

$$\frac{\partial \Phi}{\partial n} = -\vec{n} \cdot \vec{u_b}, \text{ on body surface} \tag{2.9}$$

$$\nabla \Phi \longrightarrow 0, \text{ at infinity} \tag{2.10}$$

It is important to note that Φ is a scalar potential associated with the velocity field induced by the singularity elements only. Then, in Eq. 2.9, $\partial\Phi/\partial n$ is the normal component of velocity induced by the vortex-ring at a given point while $\vec{n} \cdot \vec{u_b}$ is the sum of all other normal components due to other velocity field perturbations (free-stream, velocity due to surface deformation, influence of lifting surfaces wakes, velocity induced by external sources, as the presence of propellers, etc.). Then, in order to satisfy the non-penetration condition, the velocity induced by the vortex-ring distribution at a given point on the surface should be equal to negative the sum of all other normal velocity components.

Additionally, for the solution to be unique, it needs to satisfy Kutta-condition, which states that the flow should leave a sharp trailing edge of an airfoil smoothly and with finite velocity, and Kelvin's Theorem, which says that in order to conserve angular momentum, the time change of circulation around a curve surrounding both, airfoil and wake, should remain unchanged [57].

The zero velocity at infinity condition is automatically satisfied by the vortex-ring elements, as the velocity induced by them is inversely proportional to the square of the distance to the point considered and goes to zero when the distance goes to infinity. Also, as will be discussed along next sections, due to the way the vortex-ring are placed and the vorticity is convected to the wake, the vortex Lattice method also automatically satisfies the Kutta-condition and Kelvin's Theorem, without the necessity to add or modify equations. Then, the problem is reduced to find the circulations over the surface panels such that the non-penetration boundary condition is satisfied.

As a final note, it is worthy to observe that no linearization assumption of small perturbations was considered in the derivation of the method. Then, the Vortex Lattice method is applicable for arbitrary displacements of the wing, which is a desired characteristic for aeroelastic simulations of VFA.

2.2.2.2 Discretization in Vortex Elements and Solution Approach

Figure 2.3 illustrates how the vortex-ring elements are distributed over the wing and its wake. First, the corresponding camber surface is determined and discretized in a mesh of quadrilateral panels. Each vortex ring is assumed to have a constant circulation and is placed such that the leading edge of the vortex ring (bound vortex) is located at 1/4 chord of the panel, the chordwise sides of the vortex-ring (trailing vortex) are aligned with the sides of the panels and the remaining 4th vortex segment is aligned with the quarter-chord line of the adjacent panel in chordwise direction. The reason for positioning the vortex ring leading edge at the quarter chord line of the panel is that the 2D Kutta-condition is satisfied along the chord [57]. Collocation points, where the non-penetration condition will be applied, are located at 3/4 chords of the panel, which falls in the center of the vortex ring. Geometrical information from the surface grid such as panels area, panels normal unit vectors, coordinates of control points as well as quarter points (points in the center of each panel quarter chord line - corresponding to center of vortex-ring leading edge - where the aerodynamic loads of each panel are located) need to be determined and updated as the grid deforms.

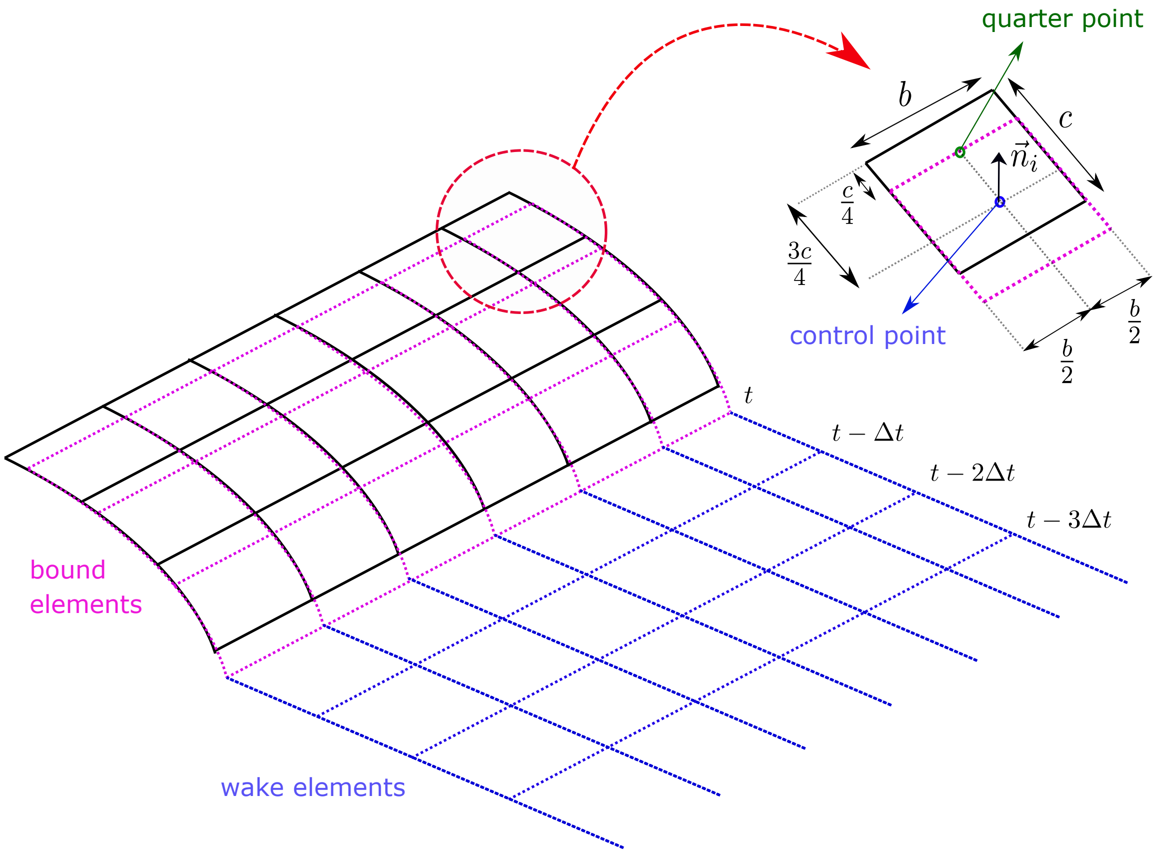


Figure 2.3: vortex-ring placement over a lifting surface and its wake

An aerodynamic coefficient matrix (AIC), where each entry a_{ij} correlates the influence of a given panel i on another panel j per unit of circulation, also needs to be determined based on the grid. The influence of one vortex ring at a given point in space is given by the added effect of each of its four vortex line segments and is calculated based on the Biot-Sarvart law. For each vortex segment, if it points from point 1 to point 2, as illustrated in Fig 2.4, the velocity u_P induced by that segment at an arbitrary point P is given by:

$$u_P = \frac{\Gamma}{4\pi} \frac{\vec{r}_1 \times \vec{r}_2}{|\vec{r}_1 \times \vec{r}_2|^2} \vec{r}_0 \cdot \left(\frac{\vec{r}_1}{r_1} - \frac{\vec{r}_2}{r_2} \right) \tag{2.11}$$

where $\vec{r}_1$ and $\vec{r}_2$ are the vectors from points 1 and 2 to point P respectively, $\vec{r}_0$ is the vector from point 1 to point 2, and Γ is the constant circulation of the vortex segment.

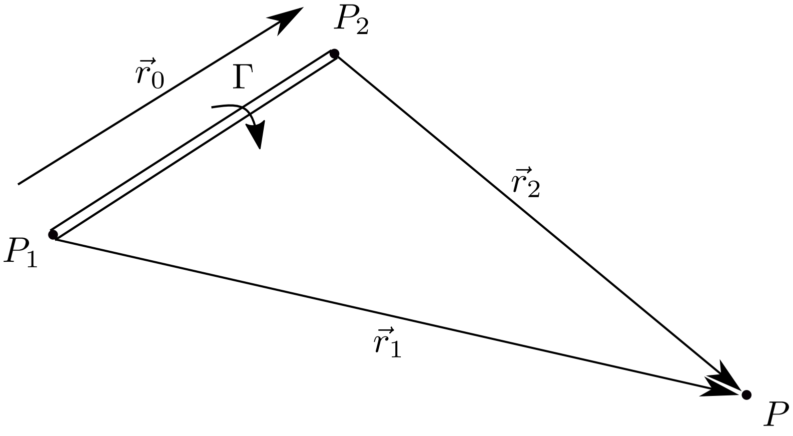


Figure 2.4: Influence of a straight vortex segment at a generic point P

Then, each coefficient a_{ij} of the AIC matrix is calculated by determining the effect of the four segments of the vortex ring associated with panel i in the control point of panel j, assuming unitary circulation. If the surfaces considered have N panels, the AIC matrix has a size of $N \times N$.

The next step is to determine, for the given static iteration or time step (if the problem is unsteady), a column matrix containing the normal components u_n of all other velocity contributions besides the panels induced velocity at the panels control points: kinematic velocity (due to onflow and/or rigid body motion of the wing), $\vec{u}_{kinematic}$, induced velocity due to the surfaces wakes, $\vec{u}_{wake}$, velocity due to body deformation, $\vec{u}_{elastic}$, gust perturbation, $\vec{u}_{gust}$, and other possible contributions (as presence of propellers, etc). This is, in fact, the way that time dependency is incorporated in the problem [57] and the normal components are determined by applying the scalar product of the calculated velocities and the normal unit vector $\vec{n}_i$ at each panel i:

$$u_{n,i} = (\vec{u}_{kinematic,i} + \vec{u}_{wake,i} + \vec{u}_{elastic,i} + \vec{u}_{gust,i}) \cdot \vec{n}_i \tag{2.12}$$

Finally, applying the non-penetration boundary condition, the following linear system is obtained:

$$[AIC]\ \Gamma = -u_n \tag{2.13}$$

where Γ is a column matrix containing the circulation of all panels, which are unknown, the product $[AIC]\ \Gamma$ is the velocity induced by the vortex-ring over the panels at the control points, and, as already commented, u_n is the sum of all normal components of other velocities contributions.

The system is solved for the circulation, Γ. Once the circulation of the panels is determined, other quantities like loads, pressure, induced velocities at a given set of points, etc., can be determined and the wake circulation, size, and position can be updated.

2.2.2.3 Aerodynamic Loads Calculation

The aerodynamic loads over the panels are calculated based on the method proposed by Mauermann [108]. The forces are divided into two contributions, steady and unsteady:

$$\vec{F}_{total} = \vec{F}_{steady} + \vec{F}_{unsteady} \tag{2.14}$$

The steady contribution is calculated based on the three-dimensional Kutta-Jowkowski therorem for vortex lift, resulting in the steady lift force normal to the panel inflow [108], which, for the i-th panel, is given by:

$$\vec{F}_{steady,i} = \rho_\infty \Gamma_{eff,i} \left(V_{\infty,eff_i} \times \vec{r}_i \right) \tag{2.15}$$

where ρ_∞ is the density of the fluid, $\vec{r}_i$ is the incremental vortex segment, corresponding to the quarter chord line of the i-th panel, $\Gamma_{eff,i}$ is the effective circulation of that vortex segment (the bound vortex) and corresponds to the vortex segment circulation itself if the panel is at the surface leading edge or the difference between the circulation of the i-th panel and the circulation of its neighbor panel in downstream direction, otherwise; V_{∞,eff_i} is the effective onflow velocity at the panel quarter point (middle of panel quarter-chord line and vortex ring leading edge) i, and corresponds to the contribution of kinematic velocity, elastic deformation, gust and other possible contributions not related to the influence of the vortex-ring. Therefore, the induced velocities due to the surfaces and their wakes singularity elements are not included in V_{∞,eff_i}:

$$V_{\infty,eff_i} = \vec{u}_{kinematic,i} + \vec{u}_{elastic,i} + \vec{u}_{gust,i} \tag{2.16}$$

For the transient cases, the unsteady contribution of the aerodynamic loads is related to the change of circulation with time and, for a panel i, is calculated as:

$$\vec{F}_{unsteady,i} = \rho_\infty A_i \frac{\partial \Gamma_i}{\partial t} \vec{n}_i \tag{2.17}$$

where A_i is the area of the panel i.

The derivative $\partial\Gamma_i/\partial t$ can be numerically calculated using some choice of backward scheme. In this work, a second-order backward scheme was used:

$$\frac{\partial\Gamma_i}{\partial t} \approx \frac{3\Gamma_i^n - 4\Gamma_i^{n-1} + \Gamma_i^{n-2}}{2\Delta t} \tag{2.18}$$

with n an index representing the current time and Δt the size of the time step.

It is worth to mention here that, as Γ is a function of the surface velocity, as it is given by the application of the no-penetration boundary condition, the differential $\partial\Gamma_i/\partial t$ will be a function of the panel acceleration and, therefore, the added mass effect is indirectly taken into account by the Vortex Lattice formulation.

In the previous load's expressions, the contribution of the velocities induced by the vortex-ring distributed over the surfaces, and their wakes were not taken into account. This implies that the induced drag was not included in those expressions. Following the approach proposed by Katz and Plotkin [57], the calculation of the induced drag is done separately, by first determining the contributions of the streamwise segments of the bound vortex-ring and all the segments of wake vortex-ring into the panels quarter points, and then applying the following expression:

$$\vec{F}_{induced\ drag,i} = \rho_\infty \Gamma_{eff,i} \left(\vec{u}_{bodystreamwise,i} + \vec{u}_{wake,i}\right) \times \vec{r}_i \tag{2.19}$$

Although Vortex Lattice method assumes inviscid flow, viscous drag estimation is also added and is briefly described in a following dedicated subsection.

2.2.2.4 Wake Modeling

The wake is updated using a time-stepping approach. At the first time step, no wake (and then no wake panels) exist. After one time step the wing has moved along its flight path and each panel at the surfaces trailing edge sheds a wake panel (vortex ring), whose circulation is equal the circulation of the corresponding trailing edge panel in the previous time step (this is because it was released at previous time step, canceling the trailing edge vortex segment circulation, and then was convected by the local flow velocity). Following this approach, Kelvin's condition (stating that total circulation surrounding the wing and the wake does not change), as well as the three-dimensional trailing edge condition (stating that the circulation at the surface trailing edge should vanish) are automatically satisfied by the vortex ring elements [57].

Also, from one time step to the next, the wake panels are convected with the local flow velocity, allowing for the free development of the wake. In this work, although any influence on the velocity flow field can be included, just the onflow velocity (velocity due to rigid body motion and atmospheric perturbation) was considered for the wake convection. This reduces the computational cost (and possible numerical instabilities related to wake roll-up) while still retaining enough information to capture wake-surfaces interaction. Also, it was observed in simulations of representative VFA that the effects of wake roll-up are usually not significant [109, 110].

For static simulations, all the wake panels in chordwise direction would virtually have the same circulation, due to the steady-state condition. This means that each column of wake panels (vortex-ring in chordwise direction) are equivalent to one long wake-panel (a long vortex ring or a horseshoe vortex). Using this information, it is possible to reduce the computational cost for the steady case by modeling the wake with fewer long vortex-ring instead of several small vortex-ring [57, 81].

Finally, as the influence of the wake is taken into account at the panel control points and quarter points at each time step and wake increase as time evolves, the cost of the computation of wake influence can become very high after some time. As the influence of the vortex-ring at some point decrease with the square of the distance to that point, it is possible to truncate the wake after it has achieved a long enough length, limiting the increase of computational cost [81].

2.2.2.5 Viscous Drag Estimation

In this work, a simple approach was implemented in order to estimate and add viscous drag effects into the UVLM calculations. First, the local lift coefficient, c_l, of each strip is calculated, based on the sum of the loads and areas of the panels composing each strip, and the mean dynamic pressure of the strip.

Then, the corresponding drag coefficient, c_d, is determined based on a pre-determined polar table (*e.g.*, XFOIL, CFD, experiment) for a representative Reynolds number, matching the local value of c_l with the corresponding value of c_d. Rotation matrices based on UM/NAST formulation are used to rotate forces from wind axis to body system for each deformed configuration.

In order to facilitate the transfer of the viscous loads from the surfaces to the structure, the viscous drag calculated for each strip is uniformly distributed among the strip panels and added to the previously calculated aerodynamic loads. Although that distribution may not be uniform, this has minimal effects on the results, as the moments caused by viscous drag, tangent to the lifting surfaces, are negligible.

2.3 Propeller-Blade Aerodynamics

2.3.1 Propeller Blade vs. Lifting Surface Aerodynamics

As the chord of a propeller blade is typically much smaller than other scales of a VFA, a detailed prediction of the aerodynamic load distribution over the area of each blade is not necessary. Instead, the knowledge of the resultant loads as well as their spanwise distribution over the propeller blades (which will determine the behavior of the propeller slipstream) are the outputs needed from the propeller blades aerodynamics model. This means that simpler and less costly methods capable of providing a good estimation of spanwise loads distribution over the propeller blades can be applied.

A typical method used to determine the spanwise load distribution over the propeller blades is the Blade Element (BE) model [49, 50], which is associated with another independent approach to model the propeller wake induced velocities. In this method, the blade is divided into segments, each one considered to behave as an independent wing, and local airfoil polar data is used to determine the loads. The assumption of independent behavior of each wing segment, however, limit the capability of the method to account for 3D-flow effects as well as the inter-influence between the different segments and may cause over-prediction of thrust and under-prediction of torque [111].

An alternative to solve those limitations, yet keeping the simplicity and small computational cost, is the use of the Lifting Line (LL) method. This method is similar to the Blade Element method in the sense that the blade is discretized in segments and local airfoil data is used, but now each segment is modeled as a vortex segment, and they can influence each other as well as the flow-field around them. Additionally, based on the spanwise distribution of the circulation, vorticity is shed from the blade into the wake (more details in a dedicated section) allowing a direct coupling (and mutual influence) between bound and wake vorticity, such that 3D effects are taken into account.

The Lifting Line approach was then chosen for the blades aerodynamics in this work due to its simplicity, small computational cost and good capability to represent unsteady behavior of blade circulation, as observed in Abedi *et al.* [50]. It should be noted that, in this context, LL is applied to model the bound vorticity only and will be coupled to another approach for the wake model, as will be discussed soon. More details about the Lifting Line formulation implemented in this work is presented in the next section.

2.3.2 Lifting Line Formulation

The Lifting Line method is also based on the potential theory assumption [57]. Following this method, the propeller blades are represented as a vortex line passing through the 1/4-chord location from leading edge at each airfoil section and discretized in N segments of constant circulation, Γ_i, with control points defined in the middle of each segment, as illustrated in Figure 2.5.

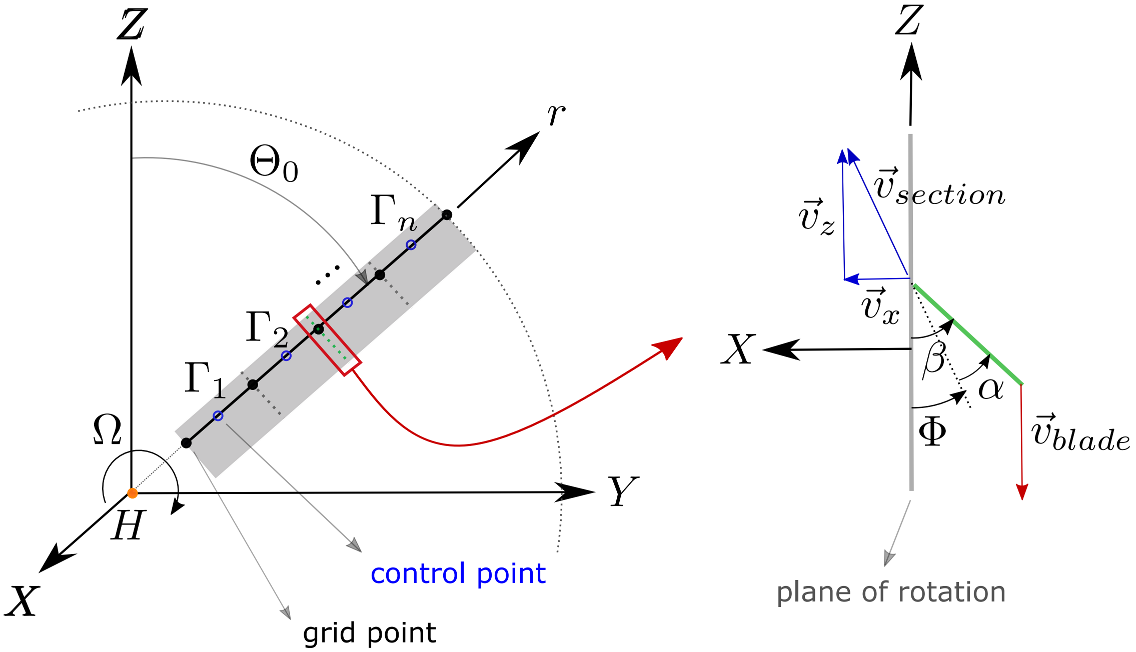


Figure 2.5: Blade representation using Lifting Line method

For each section, the local blade twist (β), chord, airfoil type and associated polar tables (for a range of Reynolds number and angle of attack) are provided. In this work, data from XFOIL was used to generate the polar tables, but any other data source, such as experiment or CFD, could be used as well. XFOIL employs the e^N method for transition prediction, which depends on the choice of the parameter N_{crit}, corresponding to the amplification factor of the most amplified frequency that causes the transition. The default value is 9, and it was applied in this work unless otherwise stated.

The grid defining the blade segments and control points, defined in body frame, moves with the propeller rotation, as well as elastic deformation, and therefore needs to be updated at each step, as below:

$$y = y_H + rcos(\Omega t + \theta_0) \tag{2.20}$$

$$z = z_H + rsin(\Omega t + \theta_0) \tag{2.21}$$

where y and z are coordinates defining the plane of rotation of the lifting line, the subscript H represents the propeller hub coordinates (in body frame), Ω is the angular propeller velocity, θ_0 is the initial angular blade position, and r is the radial position of each point. The values of hub coordinates are updated at each time step based on the elastic deformations calculated by the structural solver.

For the initial time no propeller wake is present, and the local velocity at each control point is determined based on the geometric characteristics of the section, effective freestream, $\vec{v}_{\infty,eff}$, (due to oncoming onflow velocity and propeller rotation) and other possible contributions to the velocity flow field, $\vec{v}_{ext}$, as the presence of other lifting surfaces. For the next time steps, the induced velocity due to the propeller wake is also included.

$$\vec{v}_{local} = \vec{v}_{\infty,eff} + \vec{v}_{propeller\ wake} + \vec{v}_{ext} \tag{2.22}$$

At each time step, based on the local flow velocity, the sectional Reynolds number and effective angle of attack, α, are calculated (see Figure 2.5) and used to interpolate the local aerodynamic coefficients, c_l, c_d and c_m from the polar tables provided. The loads at each section are then calculated (using the component of total velocity aligned with the section) and the resultant loads acting on the propeller hub can be determined. Those loads are then transferred to the structural nodes where the propellers are attached.

The bound vortex segments circulation is calculated from the combination of 2D Kutta-Jukowski theory and the definition of lift coefficient:

$$\Gamma = 1/2cV_{section}c_l \tag{2.23}$$

where Γ is the circulation at the considered control point, c is the local chord, $V_{section}$ is the component of total velocity at the control point that is in the plane of airfoil section considered, and c_l is the local lift coefficient.

Based on the new values of the segments circulation, the propeller wake circulation and position is updated (as will be discussed in a dedicated section) and the influence of the propeller wake and blades circulation in one another and at other points of interest, as the UVLM control points, can be determined for the next time step.

2.4 Propeller Wake Aerodynamics

In order to complete the propeller aerodynamics modeling, a propeller slipstream approach needs to be coupled to the Lifting Line method such that an estimate of the propeller wake induced velocity can be provided for the assessment of resultant loads over blades, as well as their spanwise circulation distribution.

Also, considering that the application goal is the aeroelastic simulation of VFA, where multiple surfaces (and possibly multiple propellers) can be present and moving with relation to one another, additional characteristics from the method are desired:

- Capability to take into account the mutual influence of propeller-wake and lifting surfaces or propellers that may be present, as well as their wake;

- Capability to easily accomodate wake shape to the dynamically changing aircraft configuration, avoiding singularity issues due to wakes crossing lifting surfaces;

- Be able to model wake unsteady behavior, instead of just its averaged effects;
- Have a satisfactory balance between computational cost and accuracy, given that at each time step (or static loop) the coupled aeroelastic-flight dynamics framework needs to solve a multidisciplinary problem, and it is highly desirable that each discipline demands a small computational time (yet providing sufficient accuracy), allowing design, control, and ideally, real-time applications.

There are many possibilities of propeller wake modeling available on literature, ranging from high-cost, high-fidelity solutions, as CFD approaches, to low-cost, low-fidelity solutions, as Momentum Theory, vortex filaments or panels (prescribed and free-wake), and semi-empirical approaches.

While CFD can be used to capture complex effects, as viscous effects, dynamic stall and flow separation, and has demonstrated good capability to reproduce experimental results, it is usually too expensive, in particular in the case of multiple surfaces and propellers. Besides that, due to numerical discretization, CFD methods inherently suffer from excessive dissipation, which may cause the tip vortices to appear with much less intensity than in physical reality. Also, meshing requirements add an additional layer of complexity to the solution process, and the use of numerical overset meshes (typical for this kind of application) can introduce numerical errors in the calculations.

On the other hand, although the lower-fidelity approaches provide a much quicker evaluation of propeller slipstream velocities, their accuracy is limited: many of them assume a pre-determined wake shape, isolated propeller configuration, and usually capture just averaged effects. Although those methods can be enough for some applications, this is not the case of a dynamically deforming VFA, which may contain multiple surfaces and propellers. In the case of vortex filaments (or vortex panels), although there is the possibility to capture unsteady behavior and account for the mutual interaction with other surfaces, those vortex elements have the disadvantage of requiring wake connectivity (making more difficult, for example, the modeling of propeller wake over a wing or wakes mixing cases) and, due to the potential flow assumption (therefore no viscous effects on the wake is included), they are just able to capture first-order effects, having to rely on empirical methods (*e.g.*, vortex decay factor, vortex core size) to a more realistic representation of the wake [20].

A promising alternative to those methods is the Viscous Vortex Particle method (VVPM), a mid-fidelity approach that is able to capture unsteady wake behavior, viscous diffusion, vortex mixing, and decay, as well as complex wake-wake and wake-lifting surfaces interactions, showing a good correlation with CFD and experimental results [17,20,112], but at a much smaller cost than CFD, yet higher than low-fidelity solutions. Instead of applying numerical discretization over the entire flow-field, as is typically done in CFD formulations, the method directly solves the vorticity-velocity form of incompressible Navier-Stokes equations with a Lagrangian formulation, resulting in a grid free modeling of wake diffusion without problems of artificial numerical dissipation. Also, due to the vorticity representation of the flow by free vortex particles, they can move freely with local flow velocity, allowing a natural development of the wake and avoiding singularities due to intersections between wake and lifting surfaces.

As with the UVLM method, the VVPM has the disadvantage of being a N-body problem, such that the computational cost is proportional to the square of the number N of particles, O(N^2). Since the number of particles increases at each step the calculations may become too slow (yet usually less costly than CFD). However there are approaches to overcome this problem, by limiting the number of particles or significantly reducing this cost (*e.g* Tree-code and Fast Multipole Method (FMM)), such that it can be proportional to O($Nlog(N)$) or O(N).

Therefore, the VVPM satisfies the characteristics desired for the wake modeling in coupled aeroelastic-flight dynamics framework context. The method is an attractive option for the simulation of a very flexible aircraft, involving multiple moving lifting surfaces and propellers. Therefore, it was selected as the preferred for this work. Additionally, Viscous Vortex Particle method has been successfully applied in many studies involving rotors and propellers and their interaction with other lifting surfaces or propellers [16–21, 27].

In this work, the VVPM was implemented and integrated into the UM/NAST framework, allowing the investigation of its use in the context of VFA applications, as well as the assessment of propeller slipstream effects. In the next sections, a summary of the formulation implemented is presented. For more in-depth details about the VVPM the reader is referred to Winckelmans and Leonard [113].

2.4.1 Viscous Vortex Particle Fundamentals

The basis for the VVPM dates from 1930 when Rosenhead proposed an expression for desingularised vortex particle [114] and demonstrated a dynamic simulation using singular point vortex [96]. For a long time, however, the method was limited to 2D applications. Attempts to extend to three dimensions were done in 1980 [115, 116], using vortex filament approximations, which had difficulties related to the requirement of retaining elements connectivity. The first proposition of a 3D numerical method with disconnected spherical particles was presented by Beale and Majda [117], in 1982. A proof of convergence was provided by Beale [118], given the requirement that the vortex core radius of a particle is larger than the interparticle spacing. The first implementation of the 3D method, however, took more years, with the studies of Beale [119] and Winckelmans and Leonard [120], in 1988 and 1989 respectively. Since then, it has been applied in several wake dominated contexts, in particular the cases of coaxial rotors [17, 112] and rotorcraft forward flight [121–123].

2.4.1.1 Basic Equation and Assumptions

In the VVPM, both viscosity and rotational effects are taken into account. The assumption of incompressible flow, however, is retained.

From the classical Navier-Stokes equations for incompressible flow, the momentum conservation for a differential fluid element is:

$$\frac{\partial \vec{u}}{\partial t} + \left(\vec{u} \cdot \nabla\right) \vec{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{u} \tag{2.24}$$

where $\vec{u}(\vec{x}, t)$ is the velocity field, $\omega(\vec{x}, t)$ is the vorticity field, $p(\vec{x}, t)$ is the pressure field and ρ and ν are the fluid density and kinematic viscosity, respectively.

Taking the curl of Eq. 2.24, using the fact that the fluid is incompressible (then, from continuity equation, $\nabla \cdot \vec{u} = 0$) and remembering the definition of vorticity, ($\omega = \nabla \times \vec{u}$), Eq. 2.24 can be re-written as:

$$\frac{D\vec{\omega}}{dt} = [\vec{\omega} \cdot \nabla]\vec{u} + \nu \nabla^2 \vec{\omega} \tag{2.25}$$

where $D()/dt = \partial()/\partial t + \vec{u} \cdot \nabla()$ is the material derivative.

Equation 2.26 is the vorticity-velocity in Lagrangian form and it is the fundamental equation for solving the transport of vorticity in the VVPM. The Lagrangian approach allows for a grid-free representation of vorticity, greatly simplifying the solution process compared to grid-based methods. Also, different from finite difference approaches, the convection term is not treated explicitly, avoiding problems of nonphysical numerical dissipation.

2.4.1.2 Vorticity Field Discretization

The key idea behind the VVPM is that the vorticity field can be discretized into vortex particles, corresponding to influencing elements characterized by a volume, a position, and strength. The global vorticity field is then approximated by the sum of each particle's individual vorticity field:

$$\vec{\omega}(\vec{x},t) \approx \sum_{i=1}^{N} f(\vec{x}-\vec{x}_i)\vec{\alpha}_i \tag{2.26}$$

where $\vec{x}_i$ and $\vec{\alpha}_i$ are the position and strength (vorticity times volume) of particle i, and $f(\vec{x}-\vec{x}_i)$ is the particle vorticity distribution function. Depending on how particle vorticity is distributed, they are classified as singular or regularized particles.

The particles can be interpreted as a small section of a vortex tube, with the advantage that the particles are independent in the sense that they do not necessarily belong to the same vortex filament for all times [113]. This absence of particles connectivity provides the VVPM with an inherent adaptivity to the flow-field, avoiding intersecting vortex tubes and capability to take into account wake viscous diffusion. Such capabilities allow the modeling of complex interactions, as the interaction and mixing of two propellers wakes, illustrated in Figure 2.6, where the blue dots represent the positions of the particles.

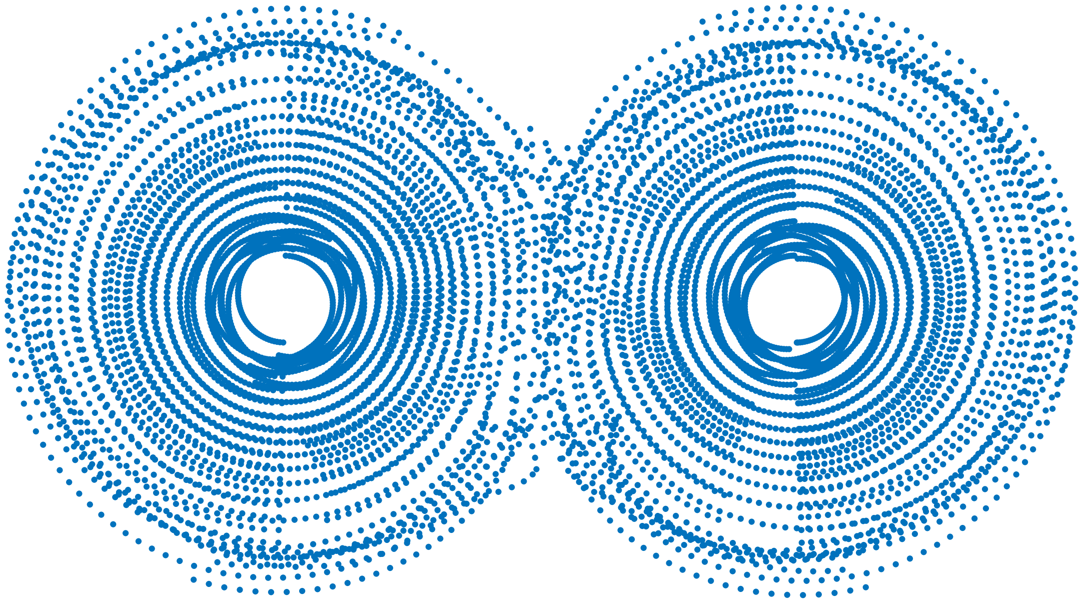


Figure 2.6: Frontal view of two adjacent propeller wakes, illustrating capability of VVPM to model complex interaction phenomena, like wake mixing

Based on the vortex particles positions and strengths in a given time, the induced velocity at any point in the domain can be determined. The particles can translate with the local flow velocity and change their strength due to the interaction with other particles and lifting surfaces present.

2.4.1.3 Combining VVPM with Potential Methods

In order to take into account the mutual aerodynamic influence between the aircraft lifting surfaces and the propellers, we need to combine a potential approach (UVLM) with an approach that includes both, viscous and rotational effects (VVPM). For that, as in the studies of Willis [21] and Martin [124], who also combined a potential approach with VVPM, we make use of the Helmholtz decomposition [125], which assumes that any sufficiently smooth, rapidly decaying vector field, can be decomposed in the sum of an irrotational (curl-free) vector field and a rotational (divergence-free) vector field. In the present context, this means that the velocity field can be represented by the superposition of an irrotational velocity field, $\vec{u}_\Phi$, associated with the UVLM solution and represented by the scalar potential function Φ and a rotational velocity field, $\vec{u}_\Psi$, associated with the LL/VVPM solutions and a vector potential function $\vec{\Psi}$ and the velocity field can be written as:

$$\vec{u} = \vec{u}_\Phi + \vec{u}_\Psi = \nabla\Phi + \nabla \times \vec{\Psi} \tag{2.27}$$

In fact, if one applies the divergence operator at the Eq. 2.27, as the divergence of the rotational velocity is zero, it will result in the Laplace equation, $\nabla^2\Phi = 0$ which is the basis for the solution approach in UVLM. By the other hand, if one applies the curl operator at Eq. 2.27, the vorticity field is obtained, $\vec{\omega} = -\nabla^2\vec{\Psi}$, which, as expected, does not depend on the potential flow solution.

Lastly, the boundary condition in the UVLM method needs to be adapted to account for the rotational velocity field. This is done simply by adding the propeller induced velocities in the non-penetration boundary condition. The boundary condition in the far-field is automatically satisfied, as the induced velocities due to the particles decrease quadratically with the distance of a given point to them, going to zero in the infinity.

2.4.2 Singular Particle Representation

In the classical proposition of the Vortex Particle method, the vorticity field is assumed to be concentrated at discrete points, called singular particle or point vortex [113, 126] and can be written as:

$$\vec{\omega}(\vec{x},t) = \sum_{i=1}^{N} \delta(\vec{x}-\vec{x}_i)\vec{\alpha}_i(t) \tag{2.28}$$

where $\delta(x)$ is the 3D δ-function, and $\vec{x}_i$ is the particle position and $\vec{\alpha}_i$ corresponds to its strength, which is given by the particle volume times the vorticity, $\vec{\alpha}_i = \vec{\omega}_i vol_p$. Then, in this approach, the region outside of the vortex particles is irrotational. Detailed mathematical approximation for this choice of particle representation can be found in [127–129].

As previously discussed, the rotational velocity field is determined from the vorticity field as the curl of a streamfunction (vector potential) $\vec{\Psi}$ which solves:

$$\vec{\omega} = -\nabla^2\vec{\Psi} \tag{2.29}$$

The streamfunction can be found in terms of the Green's s solutions for $-\nabla^2$, which in unbounded domain is given by $G(\vec{x}) = 1/(4\pi|\vec{x}|)$ [113], resulting in:

$$\vec{\Psi}(\vec{x},t) = G(\vec{x}) \circledast \vec{\omega}(\vec{x},t) = \sum_{i=1}^{N} G(\vec{x}-\vec{x}_i)\vec{\alpha}_i(t) = \frac{1}{4\pi}\sum_{i=1}^{N}\frac{\vec{\alpha}_i(t)}{|\vec{x}-\vec{x}_i|} \tag{2.30}$$

where $\circledast$ stands for the convolution product.

And finally the rotational velocity field can be then obtained by:

$$\vec{u}_{\Psi}(\vec{x},t) = \nabla \times \vec{\Psi}(\vec{x},t) = \sum_{i=1}^{N} \vec{K}(\vec{x}-\vec{x}_i) \times \vec{\alpha}_i(t) \tag{2.31}$$

where $\vec{K}(\vec{x}-\vec{x}_i)$ is the Biot-Savart kernel, given by:

$$\vec{K}(\vec{x}-\vec{x}_i) = -\frac{1}{4\pi}\frac{\vec{x}-\vec{x}_i}{|\vec{x}-\vec{x}_i|^3} \tag{2.32}$$

From a quick examination of the Biot-Savart kernel, one can conclude that it is a singular function, leading to numerical instabilities when particles move toward each other. In order to overcome this problem, the concept of "regularized particles" also known as "vortex blob" was introduced [117] and is widely applied by studies using VVPM.

2.4.3 Regularized Particle Representation

As illustrated in Figure 2.7, the basic idea of regularized particles is to substitute the 3D δ-function, $\delta(\vec{x}-\vec{x}_i)$, in the vorticity field representation (Eq. 2.28) by a distribution function (known as regularization function, regularized smoothing kernel or also cutoff function), $\xi_\sigma(\vec{x}-\vec{x}_i)$, such that the point vortex particles are transformed into vortex "blobs" with a finite core σ. Using this regularization function, the vorticity field is now written as:

$$\vec{\omega}_\sigma(\vec{x},t) = \xi_\sigma(\vec{x}-\vec{x}_i) \circledast \vec{\omega}(\vec{x},t) = \sum_{i=1}^{N} \xi_\sigma(\vec{x}-\vec{x}_i)\vec{\alpha}_i(t) \tag{2.33}$$

where $\vec{\omega}_\sigma(\vec{x},t)$ is the regularized representation of the vorticity field and $\vec{\omega}(\vec{x},t)$ is the vorticity field represented by the singular particles (Eq. 2.28).

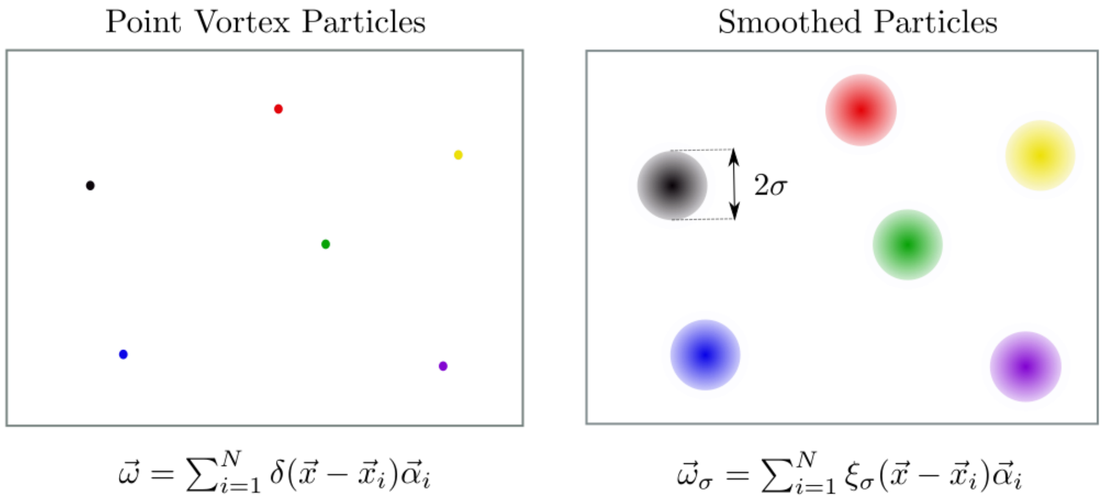

Point Vortex Particles
Smoothed Particles
2σ
$\vec{\omega} = \sum_{i=1}^{N} \delta(\vec{x} - \vec{x}_i)\vec{\alpha}_i$
$\vec{\omega}_\sigma = \sum_{i=1}^{N} \xi_\sigma(\vec{x} - \vec{x}_i)\vec{\alpha}_i$

Figure 2.7: Vorticity representation for singular versus regularized particles

2.4.3.1 Regularization Functions

A function is called smoothing kernel if it is a continuously differentiable C^∞ function, $\xi(\rho)$, and satisfies the normalization constraint, which, for three-dimensional radially symmetric functions, is given by [126]:

$$\int_0^\infty \xi(\rho)\rho^2 d\rho = \frac{1}{4\pi} \tag{2.34}$$

A smoothing kernel is said to be of order $r \epsilon \mathbb{N}_0$ if the following moment conditions are satisfied [126]:

$$\int_0^\infty \xi(\rho)\rho^{2+s} d\rho = 0, \text{ for all } 2 \leq s \leq r-1, \text{ s even} \tag{2.35}$$

$$\int_0^\infty |\xi(\rho)|\rho^{2+r} d\rho < \infty \tag{2.36}$$

Finally, a function with the format:

$$\xi_\sigma(\rho) = \frac{1}{\sigma^3}\xi(\frac{\rho}{\sigma}), \ \sigma > 0 \tag{2.37}$$

is called regularized smoothing kernel with with core size σ, where $\xi(\rho)$ a smoothing kernel function.

The normalization condition ensures the conservation of the total vorticity and can be seen as a moment condition of order zero, while the other moment conditions ensure the conservation of higher momenta, as linear and angular momentum [126]. The order of a kernel is related to the convergence rate (higher order implying faster convergence rate). This conservation of higher momenta, however, brings a loss of positivity, causing opposite signs of the smoothed vorticity field, and for positive smoothing kernels, the order needs to be $r \leq 2$ [126].

There are many possibilities for the choice of the regularized function (regularized smoothing kernel). Examples of two and three-dimensional kernels can be found in [113, 130–132]. In this work, Gaussian smoothing kernel, with order $r = 2$, was used.

$$\xi(\rho) = \frac{1}{(2\pi)^{3/2}} e^{-\rho^2/2} \tag{2.38}$$

Resulting in the regularised smoothing kernel:

$$\xi_\sigma(\rho) = \frac{1}{\sigma^3 (2\pi)^{3/2}} e^{-\rho^2/(2\sigma^2)} \tag{2.39}$$

Gaussian smoothing regularization function is a common choice for regularized vortex particle, as in [17, 20, 133–135].

2.4.3.2 Regularized Velocity Field

Similarly as in Eq. 2.29 for the singular particle representation, the regularized streamfunction, Ψ_σ is related to the regularized vorticity field ω_σ by:

$$\vec{\omega_\sigma} = -\nabla^2 \vec{\Psi_\sigma} \tag{2.40}$$

Defining $G(\rho)$ in terms of the selected smoothing kernel ξ such that [113]:

$$-\xi(\rho) = \nabla^2 G(\rho) = \frac{1}{\rho}\frac{d^2}{d\rho^2}((\rho)) \tag{2.41}$$

The regularized stream function can be determined by:

$$\Psi_\sigma(\vec{x}, t) = G(\vec{x}) \circledast \omega_\sigma(\vec{x}, t) = G_\sigma(\vec{x}) \circledast \omega(\vec{x}, t) = \sum_{i=1}^{N} G_\sigma(\vec{x} - \vec{x}_i)\vec{\alpha}_i(t) \tag{2.42}$$

where $G_\sigma(\vec{x})$ is defined as $G_\sigma(\vec{x}) = (1/\sigma)G(|\vec{x}|/\sigma)$.

Finally, the regularized rotational velocity field can be obtained:

$$\vec{u}_{\Psi\sigma} = \nabla \times \Psi_\sigma(\vec{x}, t) = \sum_{i=1}^{N} \nabla G_\sigma(\vec{x} - \vec{x}_i) \times \vec{\alpha}_i = \sum_{i=1}^{N} K_\sigma(\vec{x} - \vec{x}_i) \times \vec{\alpha}_i(t) \tag{2.43}$$

Noting that $K = \nabla G$, a formula for the direct calculation of k_σ as a function of the selected smoothing kernel is [126]:

$$K_\sigma(\vec{x}) = \frac{\vec{x}}{|\vec{x}|^3} f\left(\frac{|\vec{x}|}{\sigma}\right) \tag{2.44}$$

with

$$f(\rho) = \int_0^\rho \xi(s)s^2 ds \tag{2.45}$$

For the Gaussian smoothing kernel selected for this work, Eq. 2.38, the derivation of the regularized Biot-Savart Kernel results in:

$$K_\sigma(\vec{x}) = -\vec{x}\left[\frac{1}{4\pi|\vec{x}|^3} erf\left(\frac{|\vec{x}|}{\sqrt{2}\sigma}\right) - \frac{1}{(2\pi)^{3/2}\sigma|\vec{x}|^2} e^{-|\vec{x}|^2/(2\sigma^2)}\right] \tag{2.46}$$

Figure 2.8 shows a comparison between the singular Biot-Savart velocity kernel and its regularized version using the selected Gaussian smoothing kernel (Eq. 2.38), for different values of the core radius σ. While the Biot-Savart function goes to infinity as the distance of the point of interest tot he particle goes to zero, in the regularized version the velocity reaches a maximum at a distance about the core radius and then goes to zero as the distance to the particle further reduces. Also, as the distance increases (outside the core radius) the regularized and the singular Biot-Savart Kernel become equal.

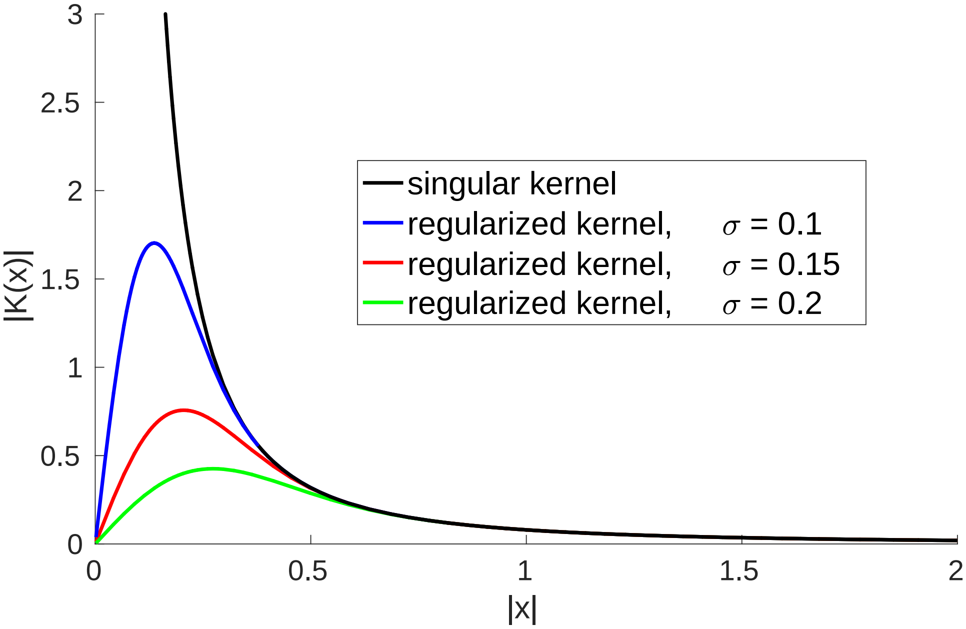


Figure 2.8: Vorticity representation for singular versus regularized particles

2.4.3.3 Overlap Criteria

Detailed mathematical demonstration for the convergence and stability of the vorticity field represented by regularized vortex particles has been developed in many studies [117,118,131,136,137]. It has been shown that, for stability and accuracy, the particles need to satisfy the overlap criteria, which couples inter-particle distances and core sizes:

$$\frac{\sigma}{h} > 1 \tag{2.47}$$

where h is the typical distance between neighbor particles.

This is illustrated in Figure 2.9, which shows a simple example considering three smoothed particles with core size σ and distance h. By satisfying the overlap criteria, the particles core region are forced to overlap, allowing particles to "communicate" with each other [126, 127].

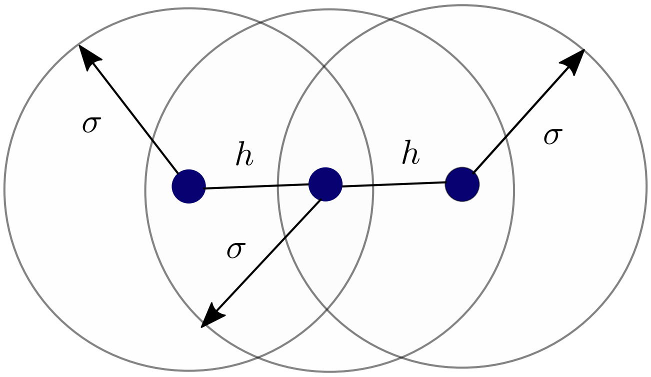


Figure 2.9: Overlap Criteria: $\sigma/h > 1$.

This overlap criteria is essential, as the error associated with the representation of the vorticity field using regularized particles has been demonstrated to be proportional to $(h/\sigma)^m$ [16, 113], where m is related to the number of derivatives that exist for the chosen smoothing kernel and is usually much greater than one.

Also, as one can observe from Figure 2.8, as the value of the particle core radius σ increases, the regularised velocity field becomes more and more smooth, and eventually will not be representative of the continuous vorticity field anymore. Then, the core radius, σ, needs to be as small as possible but satisfying the overlap criteria.

The exact relation between σ and h that would provide an ideal representation of the vorticity field was the object of many studies, and explicit analytical relations were proposed for some two-dimensional cases [130, 138–140]. However, three-dimensional cases are often more complicated, and the relation between σ and h depends on the geometric considerations for each case [126, 141, 142].

In this work, and unless otherwise stated, the value of the parameter σ is constant for all the particles and calculated by multiplying a reference length (related to the distance between neighbor particles), h_0, by a factor c, chosen based on the convergence of C_T. It should be noted that the convergence of the vortex particle method was proven for both a uniform smoothing parameter and a varying-size smoothing parameter [20].

2.4.4 Evolution Equations

Substituting the vorticity representation, given by Eq. 2.33, into the incompressible vorticity-velocity equation in Lagrangian form, Eq. 2.26, results in the evolution equation for the strength of each particle i, as given by Eq.2.48. Also, vortex particles are considered fluid elements and, as so, travel with the local flow velocity (taking into account free-stream, velocity induced by particles, velocity induced by blades and any other influences in the velocity field), leading to the particle convection equation (Eq. 2.49).

$$\frac{D\vec{\alpha}_i}{dt} = [\vec{\alpha}_i \cdot \nabla]\vec{u} + \nu \nabla^2 \vec{\alpha}_i \tag{2.48}$$

$$\frac{d\vec{x}_i}{dt} = \vec{u}(\vec{x}_i, t) \tag{2.49}$$

Equations 2.48 and 2.49 are the viscous vortex particle evolution equations, governing the update of particles strength and positions at each time step. The first term in the right hand side of Eq. 2.48, $[\vec{\alpha}_i \cdot \nabla]\vec{u}$, corresponds to the stretching effect, corresponding to vortex stretching and rotation due to the velocity field gradient. The second term in the right hand side of that equation, $\nu \nabla^2 \vec{\alpha}_i$, corresponds to the viscous diffusion, corresponding to the vorticity diffusion due to the viscous effects. More details about the calculation of each of those terms will be provided on the next sections.

Another important aspect to comment about Eq. 2.48 is about its mathematical representation, which can have different implications for the numerical implementation [102,113,126]. The way the strength evolution equation is written in Eq. 2.48 is called “classical scheme” or “direct scheme.” Making use of $\nabla \cdot \vec{u} = 0$ and $\vec{\omega} = \nabla \times \vec{u}$, Eq. 2.48 can be re-written as:

$$\frac{D\vec{\alpha}_i}{dt} = [\vec{\alpha}_i \cdot \nabla^T]\vec{u} + \nu \nabla^2 \vec{\alpha}_i \tag{2.50}$$

which is called “transposed scheme” or, alternatively, as:

$$\frac{D\vec{\alpha}_i}{dt} = [\vec{\alpha}_i \cdot (\nabla + \nabla^T)]\vec{u} + \nu \nabla^2 \vec{\alpha}_i \tag{2.51}$$

which is called “mixed scheme.” It should be noticed that only the stretching term changes from one form to another.

The three schemes given by equations 2.48, 2.50 and 2.51 would be identical if we could guarantee that the vorticity field defined by Eq. 2.33 is equal to the velocity field calculated in Eq. 2.43, $i.e.$, $\vec{\omega}_\sigma = \nabla \times \vec{u}_\sigma$ at all times. However, the vorticity field discretized by vortex particles (both singular or regularized) cannot be guaranteed to be divergent free for all times [113,126]. This is an inherent drawback of the VVPM formulation, and while there are still ongoing efforts to address this issue and some existent approaches to tackle this limitation [127], it is not in the scope of the present work.

From the curl of the regularised velocity field:

$$\nabla \times \vec{u}_\sigma = \nabla \times (\nabla \times \vec{\Psi}_\sigma) = -\nabla^2 \vec{\Psi}_\sigma + \nabla(\nabla \cdot \vec{\Psi}_\sigma) = \vec{\omega}_\sigma + \nabla(\nabla \cdot \vec{\Psi}_\sigma) \tag{2.52}$$

and, as the vorticity field is not guaranteed to be divergent free, this means that when $(\nabla \cdot \vec{\Psi}_\sigma \neq 0)$, then $\vec{\omega}_\sigma \neq \nabla \times \vec{u}_\sigma$ and, consequently, the three different formats of the strength evolution equation will not be equivalent and may lead to different numerical results [113].

The transpose scheme was favored by Rehbach [102], as the symmetry of the deformation tensor would reduce the computational cost. By the other hand, the transposed scheme was shown by Choquin and Cottet [143] to conserve the total vorticity, a characteristic that the other two schemes do not share. However, as noticed by Winckelmans and Leonard [113], for regularized particles, the transposed scheme does not show superior performance in terms of linear impulse and kinetic energy conservation when compared to the classical scheme.

Although it is not conservative in terms of total vorticity, the classic scheme was noted to have the property of not amplifying initial disturbances of the divergence-free constraint [126, 127], such that the following equation holds for the divergence of the discretized vorticity, $\nabla \cdot \vec{\omega}_\sigma$ [127]:

$$\frac{\partial(\nabla \cdot \vec{\omega})}{\partial t} + \nabla \cdot (\vec{u}(\nabla \cdot \vec{\omega})) = 0 \tag{2.53}$$

The transposed and mixed schemes do not share this property. Considering this advantage over the other schemes, the classical form was chosen for this work. Finally, in order to calculate the right hand side of Eq. 2.48, each term on the right hand side will be determined separately, as summarized in the next sections.

2.4.4.1 Vortex Stretching Effect

The isolated contribution of the stretching effect to the evolution of the particle strength is given by:

$$\left(\frac{D\vec{\alpha}_i}{dt}\right)_{stretching} = [\nabla\vec{u}(\vec{x}_i, t)] \cdot \vec{\alpha}_i \tag{2.54}$$

In this work, the gradient of the velocity field $\vec{u} = \vec{u}_\phi + \vec{u}_\Psi$ was approximated to the gradient of the rotational velocity field only:

$$\nabla\vec{u} \approx \nabla\vec{u}_\Psi \tag{2.55}$$

which means that the stretching effect due to Vortex Lattice panels was not taken into account. As will be shown later, this approximation resulted in good agreement for the propeller-wing verification case.

The rotational velocity field is given by Eq. 2.43 and can be re-written in a matrix form as:

$$\vec{u}_\Psi(\vec{x}_i, t) = \sum_{j=1}^{N} [\tilde{\alpha}_j][K_\sigma(\vec{x}_i - \vec{x}_j)] \tag{2.56}$$

where $[\tilde{\alpha}_j]$ is the skew-matrix associated with the vector $\vec{\alpha}_j$.

The gradient of $\vec{u}_\Psi(\vec{x}_i, t)$ can be, then, written as:

$$\nabla\vec{u}_\Psi(\vec{x}_i, t) = \sum_{j=1}^{N} [\tilde{\alpha}_j][\nabla K_\sigma(\vec{x}_i - \vec{x}_j)] \tag{2.57}$$

And, for the Gaussian smoothing kernel selected, the kl element of the matrix $[\nabla K_\sigma(\vec{x}_i - \vec{x}_j)]$ (with k and l the row and column index, respectively) is given by:

$$[\nabla K_\sigma(\vec{x}_i - \vec{x}_j)]_{kl} = \delta_{kl} H(\rho) - \frac{1}{\sigma^5} F(\rho)(x_{ik} - x_{jk})(x_{il} - x_{jl}) \tag{2.58}$$

where $H(\rho)$ and $F(\rho)$ are defined as:

$$H(\rho) = \frac{1}{\sigma^3 \rho^2} \left(\frac{1}{4\pi\rho} erf(\frac{\rho}{\sqrt{2}}) - \frac{1}{(2\pi)^{3/2}} e^{-\rho^2/2} \right) \tag{2.59}$$

$$F(\rho) = \frac{3H(\rho) - \xi(\rho)}{\rho^2} \tag{2.60}$$

with $\rho = |\vec{x}_i - \vec{x}_j|/\sigma$ and $\xi(\rho)$ corresponding to the Gaussian smoothing kernel defined by Eq. 2.38.

Finally, substituting Eq. 2.57 into Eq.2.54, the contribution of the stretching effect in the evolution of particle strength can be determined.

2.4.4.2 Viscous Diffusion Effect

The isolated contribution of the viscous diffusion effects to the evolution of particles strengths is given by:

$$\left(\frac{D\vec{\alpha}_i}{dt} \right)_{viscous} = \nu \nabla^2 \vec{\alpha}_i \tag{2.61}$$

There are many existent approaches for the numerical treatment of the Laplacian term ∇^2, and a comprehensive overview of those approaches applied to viscous vortex methods can be found in Barba [144] and Barba *et. al.* [140]. In this work, the particle strength exchange (PSE) method [127, 145, 146], a widely used scheme with a consistent convergence statement [126], was applied.

The basic idea of the PSE is to approximate the Laplacian operator by an integral operator, avoiding the numerical differentiation (which has a poor accuracy compared to integral operation). The integral operator is then discretized by making use of the particles representation of the vorticity field.

First, the approximation of the Laplacian of the vorticity by an integral is given by:

$$\nabla^2 \vec{\omega}(\vec{x}) \approx \frac{2}{\sigma^2} \int \eta_\sigma(\vec{x} - \vec{y})[\vec{\omega}(\vec{y}) - \vec{\omega}(\vec{x})] d\vec{y} \tag{2.62}$$

where $\eta_\sigma(\vec{x})$ is a regularized smoothing kernel satisfying moment conditions very similar to the ones defined by equations 2.35 and 2.37 (more details in [126]). In this work this kernel was chosen as the Gaussian distribution defined in Eq.2.38 and also used for the particles regularization.

The second step is to discretize the integral by applying the midpoint quadrature over all particles positions, resulting in:

$$\nabla^2 \vec{\omega}(\vec{x}) \approx \frac{2}{\sigma^2} \sum_{j=1}^{N} \eta_\sigma(\vec{x} - \vec{y_j})[\vec{\omega}_j(\vec{y_j}) - \vec{\omega}(\vec{x})] V_j \tag{2.63}$$

where V_j is the volume associated with particle j. Now, in order to evaluate the viscous diffusion term at a particle position $\vec{x}_i$, Eq.2.63 is integrated over the volume V_i of a particle i:

$$\int_{V_i} \nabla^2 \vec{\omega}(\vec{x}) d\vec{x} \approx \frac{2}{\sigma^2} \sum_{j=1}^{N} \int_{V_i} \eta_\sigma(\vec{x} - \vec{y_j})[\vec{\omega}_j(\vec{y_j}) - \vec{\omega}(\vec{x})] V_j d\vec{x} \tag{2.64}$$

And applying again the midpoint quadrature and substituting $\vec{\alpha}_i = \vec{\omega}_i V_i$ and $\vec{\alpha}_j = \vec{\omega}_j V_j$:

$$\int_{V_i} \nabla^2 \vec{\omega}(\vec{x}) d\vec{x} \approx \frac{2}{\sigma^2} \sum_{j=1}^{N} \eta_\sigma(\vec{x}_i - \vec{y_j})(V_i \vec{\alpha}_j - V_j \vec{\alpha}_i) \tag{2.65}$$

Finally, from Eq.2.61, the contribution of the viscous diffusion on the particles strength evolution is given by:

$$\left(\frac{D\vec{\alpha}_i}{dt}\right)_{viscous} = \frac{2\nu}{\sigma^2} \sum_{j=1}^{N} \eta_\sigma(\vec{x}_i - \vec{y_j})(V_i \vec{\alpha}_j - V_j \vec{\alpha}_i) \tag{2.66}$$

2.4.5 Generating New Particles

After each time step, new particles are generated satisfying the conservation of vorticity.

$$\vec{\Gamma}_{wake} = -\frac{d\vec{\Gamma}}{dt} + \vec{u}_b(\nabla \cdot \vec{\Gamma}) \tag{2.67}$$

where $\vec{\Gamma}_{wake}$ is the circulation shed into the wake, $\vec{\Gamma}$ is the bound circulation and $\vec{u}_b$ is the resultant relative velocity between the air flow and the blade.

In the preceding equation, the first term corresponds to the vorticity shed into the wake due to the time-varying circulation, called shed vorticity (associated with shedding particles). The second term corresponds to the vorticity shed into the wake due to the gradient of circulation (in this case, the spanwise distribution of circulation), and is called trailing vorticity (associated to trailing particles).

In this work, where lifting line is used to model the blades, the shed particle strength from each segment is calculated as:

$$(\vec{\alpha}_{shed}) = \frac{d\vec{\Gamma}}{dt}\ dt\ s \tag{2.68}$$

where $\vec{\Gamma}$ is the circulation at the given blade segment, dt is the size of the time step, and s is the length of the blade segment (as shed particles represent vorticity aligned with blade segment). The derivative of the circulation, $d\vec{\Gamma}/dt$, is calculated numerically by a second-order backward scheme, similarly as done in the Vortex Lattice by using Eq.2.18. The positions of new shed particles are located in the centroid of the area that each blade segment travels in one time step.

The trailing particle strength is related to the spatial change of circulation from one segment to the next and is given by:

$$\vec{\alpha}_{trailing} = (\Gamma_{i-1} - \Gamma_i)\ \vec{v_b}\ dt \tag{2.69}$$

if the propeller generates positive thrust rotating in counterclockwise direction (for one looking in front of the propeller) and,

$$\vec{\alpha}_{trailing} = -(\Gamma_{i-1} - \Gamma_i)\ \vec{v_b}\ dt \tag{2.70}$$

if the propeller is rotating in clockwise direction, where Γ_i and Γ_{i-1} are the scalar values of circulations at adjacent blade segments $i-1$ and i with i increasing from root to tip (for tip it means tip circulation minus zero), and $\vec{v_b}$ is the blade velocity with respect to the flow. Similarly as for shed particles, the position of new trailing particles are located in the middle of the line given by $\vec{v_b}\ dt$, representing the line traveled by each grid points in one time step. The direction of the trailing particles strength is such that the resultant induced velocity created by the trailing particles is along the opposite direction of thrust. Then, depending on what is the direction of rotation of propeller that generates positive thrust, the strength direction may have the same or opposite orientation to the local flow velocity.

2.4.6 Particle Refinement

Due to the different velocity gradients inside the domain, as the simulation evolves particles may concentrate in areas of negative gradient and spread in areas with positive gradient. This may lead to the non-satisfaction of the core overlap condition, resulting in unphysical behavior and numerical instabilities for longer simulation times. In the simulation of a 2D axisymmetric inviscid vortex patch example presented by Speck [126], for instance, non-negligible distortions of the vorticity field are noticed to start at about 10 seconds of simulation.

Then, in order to make the VVPM simulation more robust for longer simulation times, a method acting on restoring the vortex overlap condition is necessary. This may be done by refining the locations where the vortex stretching is intense [113].

While this is still an ongoing research area, there are many approaches available on literature that aim to avoid the numerical instabilities due to the non-satisfaction of overlap condition. Among them, it is common the use of grid-based approaches, as the methods summarized in [127]. Such grid-based approaches create new particles satisfying the overlap condition located at the grid nodes. However, the use of grid brings complexity and remove one distinctive advantage of the VVPM Lagrangian formulation, which is to be a meshless approach.

In this work a simple meshless approach proposed by Winckelmans and Leonard [113] is used to split particles whenever a particle strength doubles its initial value $|\vec{\alpha}_i(t)| \geq 2|\vec{\alpha}_i(0)|$, restoring the overlap initially established. If that condition occurs, then the particle is split into two particles with vorticity equally distributed:

$$\vec{\alpha}_{new} = \frac{1}{2}\vec{\alpha}_i \tag{2.71}$$

And located at positions:

$$\vec{x}_{new} = \vec{x}_i \pm c_r \sigma \frac{\vec{\alpha}_i}{|\vec{\alpha}_i|} \tag{2.72}$$

where c_r is a factor chosen such that the new particles smoothly replace the previous ones (*e.g.* $c_r\sigma = h$, where h is a typical distance between neighbor particles). In this work, a value of $1/4$ is used for c_r, as also done by Singh and Friedmann [17].

As argumented by Winckelmans and Leonard [113] this choice of strength split and new particles positions ensures the conservation of total vorticity and linear impulse.

2.4.7 Computational Cost

As commented previously, the VVPM behaves as an N-body problem, with a characteristic speed of $O(N^2)$ for a number N of particles. Then, as the number of particles increases with time evolution, the cost may increase significantly (yet less costly than CFD) and some approach to accelerate or at least limit this cost is necessary, especially when considering the application in aeroelastic simulations.

In this work, a cut-off distance is applied when particles are sufficiently far away from the region of interest. The sensitivity of the Viscous Vortex Particle simulation to the choice of cut-off distance was investigated by He and Zhao [20] for a rotor model. For that case, it was concluded that the influence of the wake cut-off distance (r_{cut}) on the simulation results is not significant provided that $r_{cut} \geq 2R$. Also, code parallelization was implemented on the most expensive portions in order to reduce the computational cost of the remaining particles inside the region of interest.

Although it has not been currently implemented in this work, it is worth to mention that there are acceleration algorithms developed for N-body problems that could further accelerate the calculations, being a desirable future step. Two commonly used methods in the context of Vortex Particle simulations are the TreeCode method [147, 148] and the Fast Multipole Method (FMM) [149, 150]. Both methods require the generation of a data structure (usually in the form of an Oct-Tree) and differ mainly in the way the tree is traveled and the velocities are evaluated [20]. In the TreeCode the action of a group of particles on a target particle is approximated through a multipole expansion if the distance between the center of the particles cluster and the target is larger than a specified critical distance. In the FMM, if two groups of particles are far away, the influence of the source group at the center of the target group is first determined by a multipole series expansion, and then the velocity at each particle inside the target group is determined by a local Taylor expansion. In general, the TreeCode algorithm can reduce the cost to $O(NlogN)$ while the FMM can reduce the cost to $O(NlogN)$ or even to $O(N)$, being, however, more complex in the conceptual and implementation level [27].

2.5 Propeller Inertial Effects

To derive the inertial loads that a rotating propeller will transfer to the body, consider the reference system (as presented in Figure 2.10): i) Inertial frame (I): as the name implies, a fixed inertial global frame; ii) Body frame (B): a frame attached to the body, that may translate and rotate with relation to the inertial frame; iii) Local beam frame (W): a frame attached to a structural beam node and it may translate and rotate in relation to the body frame due to elastic deformations; iv) Propeller frame (P): a frame whose origin is attached at the propeller hub position and has no rotation or translation with relation to local frame W; v) Blade frame (b): a frame that is attached to one of the propeller blades, has the same origin as the propeller frame, and rotates with relation to it with the propeller angular velocity.

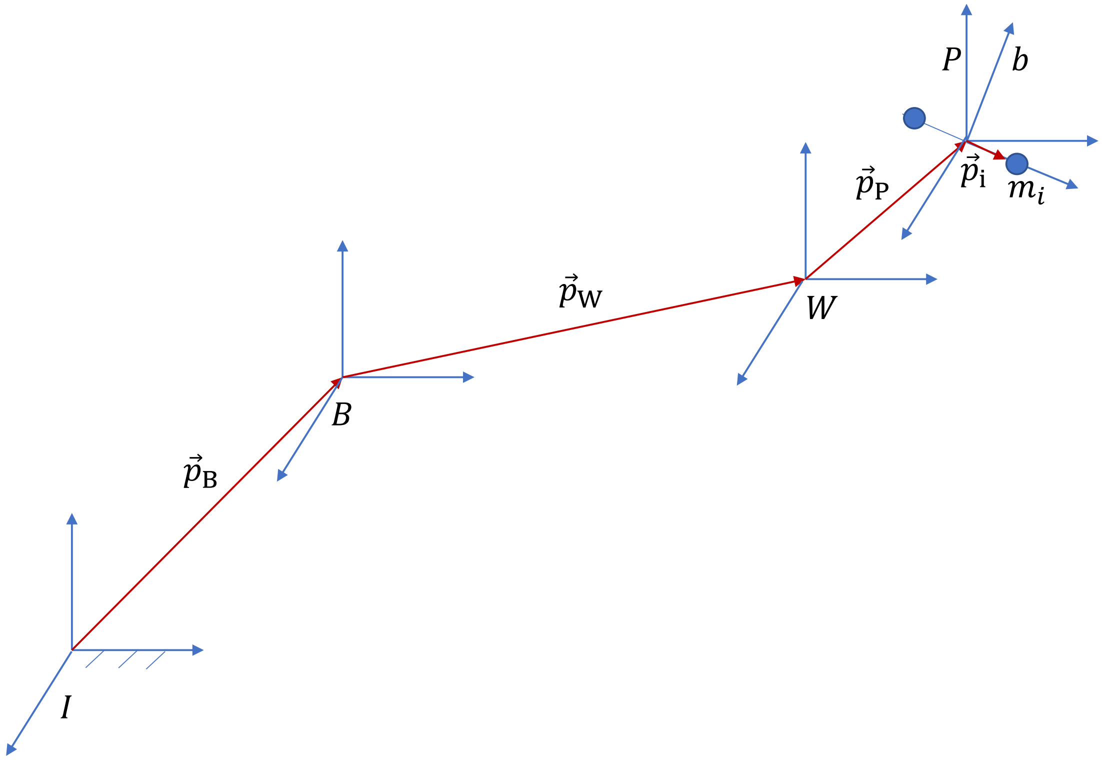


Figure 2.10: Representation of various reference systems used on modeling blade inertial effects

It is assumed the propeller configuration is such that: i) all blades have the same geometry and mass distribution, and ii) the same angle between adjacent blades. Note that from i) and ii) one has:

$$\sum_{i=1}^{N_{blades}} \vec{r}_{blade\ CGi} = \vec{0} \tag{2.73}$$

where N_{blades} is the number of blades and $\vec{r}_{blade\ CG,i}$ is the CG position of the i-th blade relative to the propeller hub, written in any desired frame. Both assumptions above are reasonable for most practical propellers.

2.5.1 Propeller Inertial Forces

Consider a structural model for the propeller in which each blade is represented by a rigid, massless rod with a concentrated mass (equal to total blade mass) located at the blade CG (the distribution of mass along blades is included below). Then, the position of one of the concentrated masses i defined with relation to the inertial frame (I) can be written in terms of the body frame (B) as:

$$\vec{p}_{m_i} = \vec{p}_B + \vec{p}_W + \vec{p}_P + \vec{p}_i \tag{2.74}$$

where, as illustrated in Figure 2.10, $\vec{p}_B$ is the position of the origin of the body frame with relation to the inertial frame, $\vec{p}_W$ is defined as the position of the origin of the local frame with relation to the body frame, $\vec{p}_P$ is the position of the origin of the propeller frame with relation to the local frame, and $\vec{p}_i$ is the position of the point mass i with relation to the propeller frame. While this is a vector sum, it is convenient to write all the quantities in the body frame.

Recalling rigid-body dynamics theory, the time derivatives of a vector $\vec{r}$ expressed with respect to two different frames, (I) and (B), rotating with respect to each other is given by:

$$ {}^{I}\dot{\vec{r}} = {}^{B}\dot{\vec{r}} + {}^{I}\vec{\omega}^{B} \times \vec{r} \tag{2.75} $$

where ${}^{I}\dot{\vec{r}}$ is the time derivative of $\vec{r}$ in frame I, ${}^{B}\dot{\vec{r}}$ is the time derivative of $\vec{r}$ with respect to B, and ${}^{I}\vec{\omega}^{B}$ is the angular velocity vector of B with relation to I.

Applying Eq. 2.75, the first and second time derivatives of the vector $\vec{p}_{mi}$ can be expressed as:

$$ {}^{I}\dot{\vec{p}}_{m_i} = {}^{B}\dot{\vec{p}}_B + {}^{I}\vec{\omega}^{B} \times \vec{p}_B + {}^{B}\dot{\vec{p}}_W + {}^{I}\vec{\omega}^{B} \times \vec{p}_W + {}^{B}\dot{\vec{p}}_P + {}^{I}\vec{\omega}^{B} \times \vec{p}_P + {}^{B}\dot{\vec{p}}_i + {}^{I}\vec{\omega}^{B} \times \vec{p}_i \tag{2.76} $$

$$ \begin{aligned} {}^{I}\ddot{\vec{p}}_{m_i} = {} & {}^{B}\ddot{\vec{p}}_B + {}^{B}\ddot{\vec{p}}_W + {}^{B}\ddot{\vec{p}}_P + 2\,{}^{I}\vec{\omega}^{B} \times \left({}^{B}\dot{\vec{p}}_B + {}^{B}\dot{\vec{p}}_W + {}^{B}\dot{\vec{p}}_P\right) \\ & + {}^{B}\,{}^{I}\dot{\vec{\omega}}^{B} \times \left(\vec{p}_B + \vec{p}_W + \vec{p}_P\right) + {}^{I}\vec{\omega}^{B} \times {}^{I}\vec{\omega}^{B} \times \left(\vec{p}_B + \vec{p}_W + \vec{p}_P\right) \\ & + \left({}^{B}\ddot{\vec{p}}_i + 2\,{}^{I}\vec{\omega}^{B} \times {}^{B}\dot{\vec{p}}_i + {}^{B}\,{}^{I}\dot{\vec{\omega}}^{B} \times \vec{p}_i + {}^{I}\vec{\omega}^{B} \times {}^{I}\vec{\omega}^{B} \times \vec{p}_i\right) \end{aligned} \tag{2.77} $$

Note that the superindex B is omitted from this point on since all derivatives are defined in the B frame.

The inertial force that each mass m_i transfers to the structure is:

$$ \vec{F}_{m_i} = -m_i\, {}^{I}\ddot{\vec{p}}_{m_i} \tag{2.78} $$

With the assumption that each blade has the same mass, the total force acting on the propeller hub due to inertial effects is:

$$
\begin{aligned}
\vec{F}_{hub} =& m_p(\ddot{\vec{p}}_B + \ddot{\vec{p}}_W + \ddot{\vec{p}}_P + 2{}^I\vec{\omega}^B \times \ (\dot{\vec{p}}_B + \dot{\vec{p}}_W + \dot{\vec{p}}_P) \\
&+ {}^I\dot{\vec{\omega}}^B \times \ (\vec{p}_B + \vec{p}_W + \vec{p}_P) + {}^I\vec{\omega}^B \times \ {}^I\vec{\omega}^B \times \ (\vec{p}_B + \vec{p}_W + \vec{p}_P)) \\
&+ m_p/N_{blades} \sum_{i=1}^{N_{blades}} [\ ((\ddot{\vec{p}}_i + 2{}^I\vec{\omega}^B \times \ \dot{\vec{p}}_i + {}^I\dot{\vec{\omega}}^B \times \ \vec{p}_i + {}^I\vec{\omega}^B \times \ {}^I\vec{\omega}^B \times \ \vec{p}_i))\ ]
\end{aligned}
\tag{2.79}
$$

where m_p represents total propeller mass.

Also, from Eq. 2.73, the third line in Eq. 2.79 goes to zero, resulting in:

$$
\begin{aligned}
\vec{F}_{hub} =& m_p\ (\ddot{\vec{p}}_B + \ddot{\vec{p}}_W + \ddot{\vec{p}}_P + 2\ {}^I\vec{\omega}^B \times \ (\dot{\vec{p}}_B + \dot{\vec{p}}_W + \dot{\vec{p}}_P) \\
&+^I \dot{\vec{\omega}}^B \times \ (\vec{p}_B + \vec{p}_W + \vec{p}_P) +^I \vec{\omega}^B \times \ {}^I\vec{\omega}^B \times \ (\vec{p}_B + \vec{p}_W + \vec{p}_P)\ )
\end{aligned}
\tag{2.80}
$$

Note that if one concentrates the mass m_p at the propeller hub location and follows a similar derivation, the same Eq. 2.80 would be obtained. Then, as expected, the inertial forces acting on the propeller hub are equivalent to the force acting in a punctual mass m_p located at that location.

2.5.2 Propeller Inertial Moments

Now, consider the propeller inertial moments transferred to the body. The objective is to determine the moment due to the inertial forces acting on each rotating punctual mass transferred to the local node position, where the origin of the W frame is situated. The moment of those forces in relation to the origin of the W frame can be determined by:

$$
\vec{M}_W = \sum_{i=1}^{N_{blades}} [\ (\vec{p}_P + \vec{p}_i) \times \vec{F}_{mi}\ ]
\tag{2.81}
$$

where F_{mi} is giving by Eq. 2.78.

As $\vec{p}_P$ is independent of the index i and remembering that $\vec{F}_{hub} = \sum_{i=1}^{N_{blades}} \vec{F}_{mi}$, one can re-write Eq. 2.81 as:

$$\vec{M}_W = \vec{p}_P \times \vec{F}_{hub} + \sum_{i=1}^{N_{blades}} [\ \vec{p}_i \times \vec{F}_{mi}\ ] \tag{2.82}$$

Then, the moment transferred to the origin of the W frame due to the inertial forces from the rotating masses is equivalent to the moment with relation to the origin of the W frame due to a concentrated mass m_P at the propeller hub plus a contribution due to the fact that masses are rotating around the hub. In the next section, this additional term is further addressed.

2.5.3 Contribution due to Rotating Masses around the Hub

Notice from Eq. 2.77 that ${}^I\ddot{\vec{p}}_{mi}$ can be written as a contribution of the hub acceleration, $\vec{a}_{hub}$, which does not depend on the indivudual blade (index i), and a term due to the rotation of the mass i around the hub. Then, the second term on the right hand side of Eq. 2.82, $\vec{M}_{rot}$, can be expressed as:

$$\vec{M}_{rot} = \sum_{i=1}^{N_{blades}} -m_i[\ \vec{p}_i \times (\ \vec{a}_{hub} + (\ddot{\vec{p}}_i + 2^I\vec{\omega}^B \times \dot{\vec{p}}_i + {}^I\dot{\vec{\omega}}^B \times \vec{p}_i + {}^I\vec{\omega}^B \times {}^I\vec{\omega}^B \times \vec{p}_i))] \tag{2.83}$$

which yields:

$$\vec{M}_{rot} = -m_i(\sum_{i=1}^{N_{blades}} \vec{p}_i) \times \vec{a}_{hub} + \sum_{i=1}^{N_{blades}} -m_i[\ \vec{p}_i \times (\ddot{\vec{p}}_i + 2^I\vec{\omega}^B \times \dot{\vec{p}}_i + {}^I\dot{\vec{\omega}}^B \times \vec{p}_i + {}^I\vec{\omega}^B \times {}^I\vec{\omega}^B \times \vec{p}_i)] \tag{2.84}$$

Invoking Eq. 2.73, the moment equation simplifies to:

$$\vec{M}_{rot} = \sum_{i=1}^{N_{blades}} -m_i[\ \vec{p}_i \times (\ddot{\vec{p}}_i + 2^I\vec{\omega}^B \times\ \dot{\vec{p}}_i +^I \dot{\vec{\omega}}^B \times\ \vec{p}_i +^I \vec{\omega}^B \times\ {}^I\vec{\omega}^B \times\ \vec{p}_i)] \tag{2.85}$$

The derivatives $\dot{\vec{p}_i}$ and $\ddot{\vec{p}_i}$ are defined in the body frame, ${}^B\dot{\vec{r}} = \sum_{i=1}^{3} dr_i/dt\ \hat{u}_{Bi}$, where $\hat{u}_{Bi}$ is the set of unit vectors in the B frame, and they also depend on the propeller angular velocity (Eq. 2.75 applied between the body and the blade frames). Also, notice that until here no assumption was made about the angular velocity of propeller.

If one considers constant angular velocity, the derivatives $\ddot{\vec{p}_i}$ and $\dot{\vec{p}_i}$ can be written in terms of angular velocity of the propeller frame with relation to the body frame, ${}^B\vec{\omega}^P$, and angular velocity of blades in relation to the propeller frame, ${}^B\vec{\omega}^b$, as:

$$\dot{\vec{p}_i} = ({}^B\vec{\omega}^P +^P \vec{\omega}^b) \times \vec{p}_i \tag{2.86}$$

$$\ddot{\vec{p}_i} =^B \dot{\vec{\omega}}^P\ {}^b\vec{p}_i + ({}^B\vec{\omega}^P +^P \vec{\omega}^b) \times ({}^B\vec{\omega}^P +^P \vec{\omega}^b) \times \vec{p}_i \tag{2.87}$$

which clearly shows that $\vec{M}_{rot}$ accounts for gyroscopic effects of the rotating blades.

2.5.4 Blade Representation with Discrete Masses

Modeling propeller blades as a concentrated mass at its CG will neglect the contribution of inertia due to the distributed blade mass in relation to its CG. Since this term may not be negligible, the formulation presented above can be extended by modeling the propeller blade in N_{seg} rigid segments with a concentrated mass at the middle of each segment, whose distribution is approximated as a function of its radial position:

$$m_k = m_{blade} c_k^2 / (\sum_{k=1}^{N_{seg}} c_k^2) \tag{2.88}$$

where the index k is related to the radial position along the blade, $m_{blade} = m_p/N_{blades}$ is the mass of one blade, and c_k is the chord at radial position k. Notice that the propeller mass m_P and the distribution of chord sizes along the radial position are data typically available in propeller databases. This is not the case for the blade thickness distribution. Then, for simplicity, the distribution of concentrated masses was calculated just in terms of the chord distribution.

Equation 2.85 is derived considering each blade is modeled as a concentrated mass at the same distance from the hub. Then, it holds for each set of masses at the same radial position k along the blade. Therefore, based on this, one can write the additional moment $\vec{M}_{rot}$ as:

$$\vec{M}_{rot} = \sum_{k=1}^{N_{seg}} \sum_{i=1}^{N_{blades}} -m_k[\,\vec{p}_{i,k} \times (\ddot{\vec{p}}_{i,k} + 2^I\vec{\omega}^B \times \dot{\vec{p}}_{i,k} + {}^I\dot{\vec{\omega}}^B \times \vec{p}_{i,k} + {}^I\vec{\omega}^B \times {}^I\vec{\omega}^B \times \vec{p}_{i,k})] \quad (2.89)$$

2.5.5 Extending UM/NAST Formulation

From the previous formulation (see Eqs. 2.80 and 2.82), it is evident that the inertial loads acting on the propeller are equivalent to the inertial loads acting on a point mass m_p located at the propeller hub plus a moment contribution due to the fact that the masses are actually rotating around the hub. Based on this, the inertial propeller loads were modeled by defining a concentrated mass m_p at the propeller hub (which was already available in UM/NAST) and an additional moment term given by Eqs. 2.85, 2.86, 2.87, and 2.89.

2.6 Interfacing Disciplines

2.6.1 Interfacing Structure and Aerodynamics

Since the structural model employs a 1D finite element representation and the uVLM uses 2D non-planar grid, an interface between the two solvers is required to transfer loads and displacements, as summarized in Figure 2.11. Assuming rigid behavior in the chordwise direction (no change in local camber), the local beam coordinates and orientation given by the UM/NAST structural solver are converted to a non-planar panel grid by re-writing the camber line coordinates of each section along the span from airfoil coordinates to body coordinates.

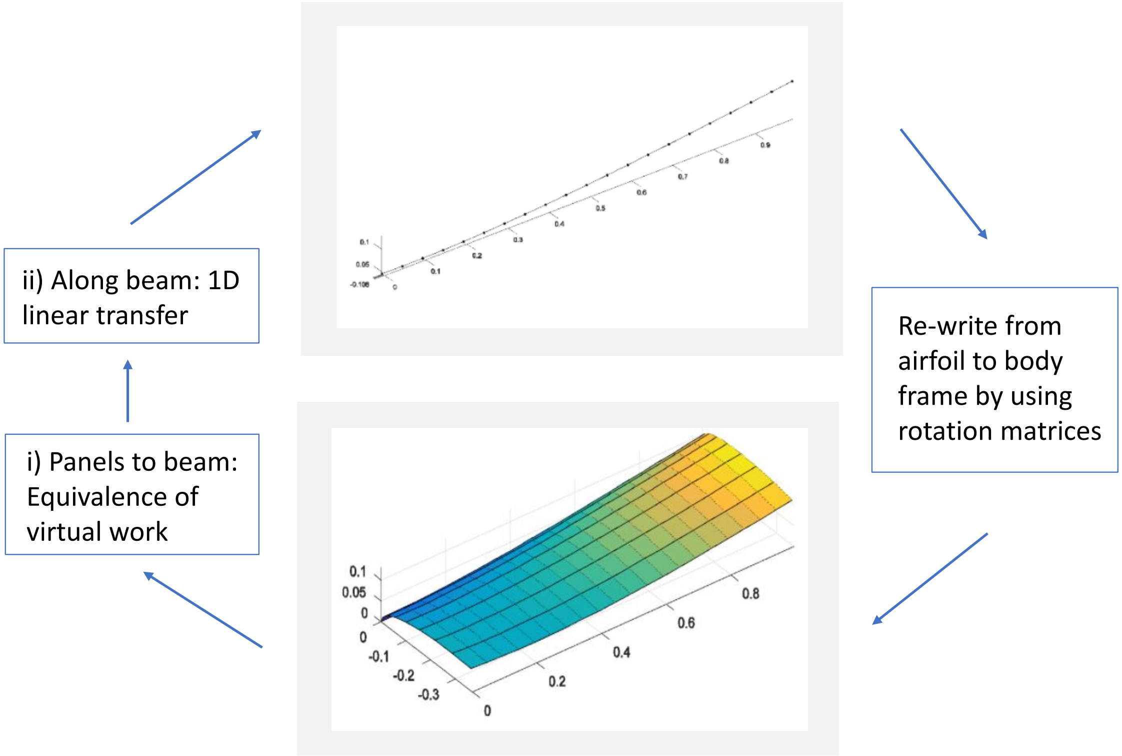

ii) Along beam: 1D linear transfer
i) Panels to beam: Equivalence of virtual work
Re-write from airfoil to body frame by using rotation matrices
0.1
0.05
0
0
-0.1
-0.2
-0.3
0
0.2
0.4
0.6
0.8

Figure 2.11: Interface between structural and aerodynamic solvers

The transfer of loads from panels to beam structural nodes is accomplished in two steps:

i) energy-consistent transfer of loads from the panels to the neighboring points along the beam given by panel discretization:

   Assuming negligible deformations in chordwise direction, the displacements of each panel quarter point (points located at 1/4 of panel chord from panel leading edge and half-panel span) can be written as a function of the points given by spanwise discretization along the beam. Then, in order to have the virtual work provided by the distribution of generalized forces on aerodynamic grid points equivalent to the virtual work performed by the distribution of generalized forces over the beam nodes, one has:

$$\{\delta u_a\}^T \{F_a\} = \{\delta u_s\}^T \{F_s\} \tag{2.90}$$

where $\{F_a\}$ and $\{F_s\}$ are column vectors containing the generalized aerodynamic and structural forces on each point (over the aerodynamic grid and structural nodes, respectively), and $\{\delta u_a\}$ and $\{\delta u_s\}$ are the corresponding virtual displacements. Then, as the displacements of the quarter-chord points and nodes along the beam are related by the assumption of a rigid chord-wise direction, an energy-consistent relation to transfer forces from panels to beam nodes can be determined.

ii) linear transfer of loads from those points to the structural nodes:

The loads at beam nodes given by span-wise discretization may need to be transferred to the structural nodes, as aerodynamic and structural span-wise discretizations can be different. For this case, a linear approach to transfer loads to neighboring structural nodes is employed. This approach is in accordance with the UM/NAST assumption of constant strains along each beam element.

2.6.2 Interfacing Lifting Surfaces and Propeller Aerodynamics

To integrate the uVLM for lifting surfaces with propeller modeling, the mutual influence between them needs to be taken into account. Figure 2.12 illustrates how the process of integration occurs.

At a given time step, the wing panel and wake panel circulations are updated by

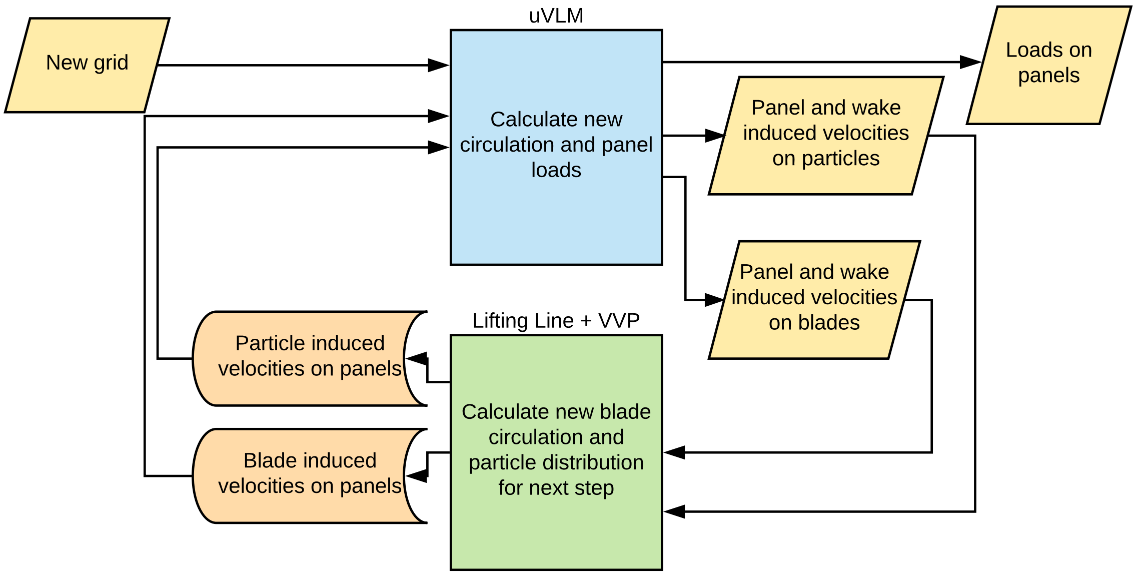

New grid
uVLM
Calculate new circulation and panel loads
Loads on panels
Panel and wake induced velocities on particles
Panel and wake induced velocities on blades
Lifting Line + VVP
Particle induced velocities on panels
Calculate new blade circulation and particle distribution for next step
Blade induced velocities on panels

Figure 2.12: uVLM and propeller model integration approach

uVLM considering the induced velocity due to current particle distribution (calculated in the previous time step) and blade bound circulation. Then, based on the new wing and wake panel circulations, the new circulation of the blades are calculated, the existing particles have positions and strength updated, and new particles are generated, defining new particle distribution for the next time step. It is important to notice that the time step used in the propeller solver should be smaller than the propeller period divided by the number of blades. In order to not restrict the time step of the dynamic solver by the propeller solver requirements, one should consider the propeller time step smaller than the global dynamic solver time step. In this work, and unless otherwise stated, the propeller time step was half of the time step on the dynamic solver.

CHAPTER 3

Numerical Implementation

This chapter offers an overview of the numerical implementation of the new components related to the propeller development, how they relate to each other, and how they integrate with the UM/NAST framework. Several block diagrams illustrate the architecture and workflow of each piece and brief explanations about the steps and involved functions are provided. The goal is to clarify the solution process and provide basic information in order to facilitate the usage and future improvements of the code.

3.1 Numerical Framework Evolution

As mentioned previously, the University of Michigan's Nonlinear Aeroelastic Simulation Toolbox (UM/NAST) [6,31,32] was used as the basis for this coupled nonlinear aeroelastic-flight dynamics framework, and enhancements were added to capture the different propeller effects. The numerical framework of UM/NAST has been evolving by a collaborative effort of several researchers under the guidance of Professor Carlos E. S. Cesnik. In Su [76], a summary of the initial steps of this framework development is presented. The first UM/NAST version was written in MATLAB by Brown [151], followed by major improvements implemented by Shearer [152] and Su [76]. In order to improve computational efficiency and extensibility, the code was re-written in C++ (Pang [153]) and has continued evolving until the present.

Due to this dynamic characteristic of the UM/NAST numerical framework, different implementations and integrations were performed along this dissertation in order to incorporate the formulation presented in the previous chapter, and are briefly summarized as follow:

- The propeller effects were added in the first UM/NAST C++ version (UM/-NAST version 2) by i) adapting and integrating an existent Vortex Lattice code developed by Ritter [81] (writing a corresponding wrapper to communicate the different codes, which were in C++ and Python/FORTRAN, respectively); ii) developing an interface between UM/NAST structural solver and the Vortex Lattice solver; iii) developing and integrating the LL/VVPM propeller solver with the Vortex Lattice and, finally, iv) implementing the derived gyroscopic loads formulation, as a function call inside UM/NAST loads calculation. Detailed verification of this enhanced framework was performed, with comparisons for each piece of the integration, as will be presented later in this work.

- To improve the performance and facilitate the integration with UM/NAST, the Python and FORTRAN portions of the codes were re-written in C++, with i) a C++ library for the propeller solver, ii) a C++ library for the UVLM solver, iii) a C++ interface class between a general propeller and lifting surface aerodynamics, and iv) an interface class between UM/NAST and the UVLM solver (which, by its turn, was coupled to the propeller). As by the time UM/NAST version 3 was already developed, those new developments were integrated into that version and new verification, now against the previous integration, was performed.

- Finally, recent improvements have been done to UM/NAST and its organization, as well as the way aerodynamic solutions are integrated, resulting in a completely different version, UM/NAST 4.0. Recent integration of the developed C++ codes has been performed with this new UM/NAST structure and it is currently in its final phase of tests.

3.2 Integration with UM/NAST 4.0

In UM/NAST 4.0 the aerodynamic solvers are no more an intrinsic part of UM/-NAST core structure, but they are treated as external libraries and linked by an interface class with the UM/NAST kernel. The UM/NAST itself is a library, and both libraries, UM/NAST and aerodynamics, are linked and called by an external driver. This process is illustrated in Figure 3.1, where a high-level overview of the UVLM/Propeller module is also presented.

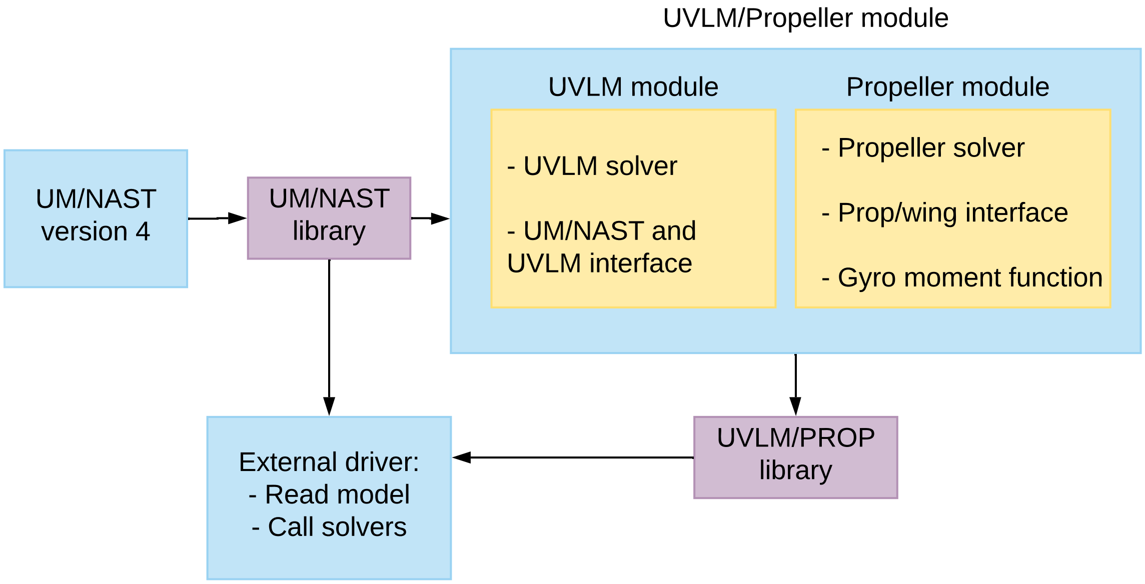


Figure 3.1: Overview integration UVLM/propeller module and UM/NAST 4.0

First, UM/NAST is installed, and a library is generated. Then, the UM/NAST library is called inside the external aerodynamic module, in this case, the UVLM/Propeller module. The link between the codes is done by inheriting from an interface class defined inside UM/NAST (class `AeroSolver`). This plugin class has pre-defined virtual functions which should be defined with the same name in the corresponding external aerodynamic solver, according to UM/NAST 4.0 documentation [154]. The external solver is also installed and generates a UVLM/PROP library. Finally, both libraries are directly invoked by an external driver, which reads the model geometry, the case conditions, and call the respective solvers for the desired simulation.

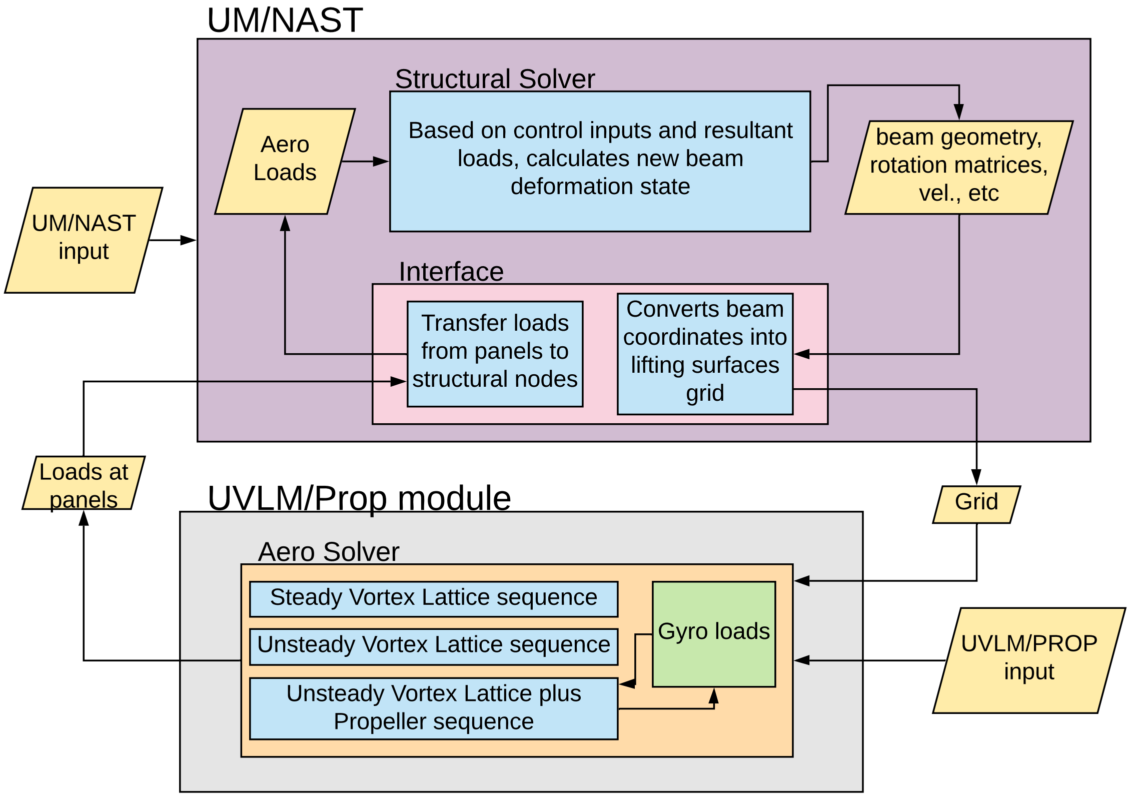

UM/NAST
Structural Solver
Aero Loads
Based on control inputs and resultant loads, calculates new beam deformation state
beam geometry, rotation matrices, vel., etc
UM/NAST input
Interface
Transfer loads from panels to structural nodes
Converts beam coordinates into lifting surfaces grid
Loads at panels
UVLM/Prop module
Grid
Aero Solver
Steady Vortex Lattice sequence
Unsteady Vortex Lattice sequence
Unsteady Vortex Lattice plus Propeller sequence
Gyro loads
UVLM/PROP input

Figure 3.2: Overview of integrated UM/NAST and UVLM/Propeller module

As shown in Figure 3.1, the UVLM/Propeller module is organized into two submodules: i) UVLM module, containing the UVLM solver and the UM/NAST and UVLM class interface, and ii) the propeller module, containing propeller solvers (currently LL/VVPM solver is the only option, but other propeller solvers can be added, as will be discussed soon), the propeller and lifting surfaces interface and a function for the gyroscopic moment implementation. An overview of the workflow of the integrated UM/NAST and UVLM/Propeller module is presented in Figure 3.2. For each time step (or static iteration) the current beam coordinates are converted by the UM/NAST and UVLM interface into an UVLM grid. The selected solver is called (static or dynamic, with or without propeller) and the loads from panels (as well as loads from the gyroscopic moments at propeller hubs) are transferred again by the interface to the nodes of the UVLM structural beam model. Based on the new loads, the new geometry is determined, and the process continues.

In the next sections, each piece of the UVLM/Propeller module is presented.

3.3 New C++ Developments

3.3.1 Steady/Unsteady Vortex Lattice Library

The vortex Lattice library was written based on the Python/FORTRAN Vortex Lattice code developed by Ritter [81]. It is a C++ class and can be used for isolated aerodynamic analysis as well as the coupling with the UM/NAST framework. Also, propeller influence was incorporated in the calculation of the panels' loads and circulation and a static Vortex Lattice solver including averaged propeller effects was developed. More details about the code are presented in the next subsections.

3.3.1.1 Architecture Overview

Figure 3.3 presents an overview of the Vortex Lattice library architecture. It consists of a C++ class containing private variables and functions, for internal use of the Vortex Lattice code only, and public functions and variables, accessible from external code. The public variables are divided into argument variables and output variables, allowing efficient and quick communication between different codes by directly accessing and modifying those variables.

For the aeroelastic coupling, two input files are expected: i) an input file called `input_uvlm.xml` containing several Vortex Lattice parameters, as presented in Table 3.1; and ii) a txt input file (with a user-defined name) containing the coordinates of the airfoils involved, given in the format provided by XFOIL. If the solver is run-

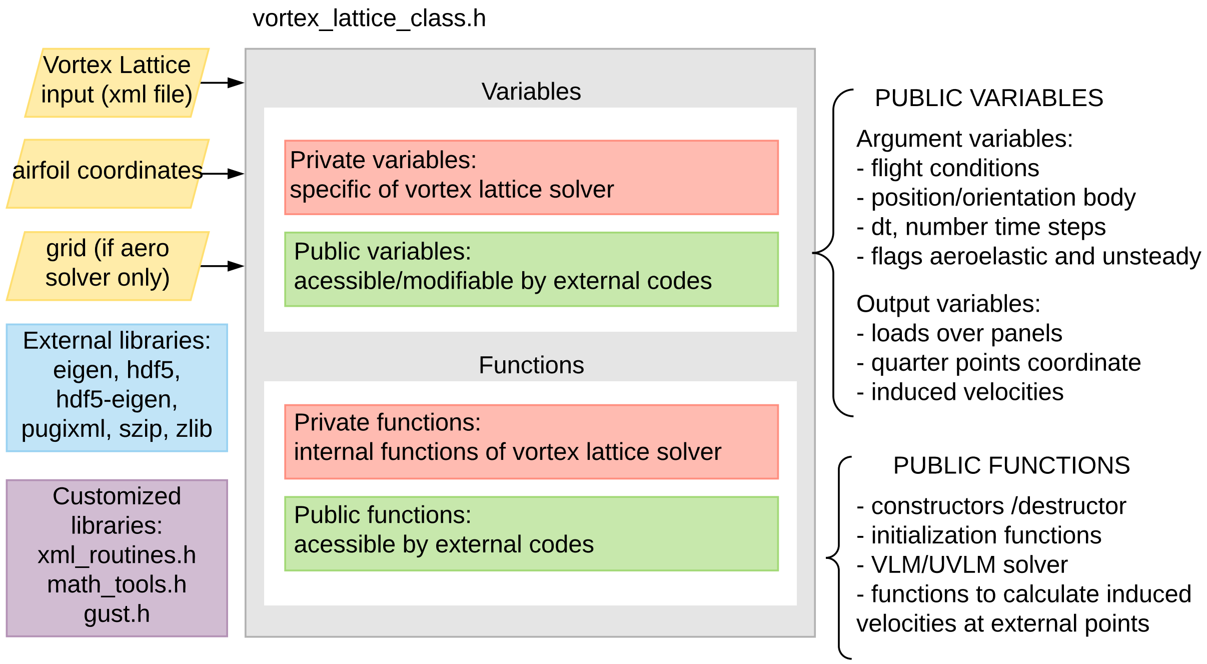

vortex_lattice_class.h
Vortex Lattice input (xml file)
airfoil coordinates
grid (if aero solver only)
External libraries: eigen, hdf5, hdf5-eigen, pugixml, szip, zlib
Customized libraries: xml_routines.h math_tools.h gust.h
Variables
Private variables: specific of vortex lattice solver
Public variables: acessible/modifiable by external codes
Functions
Private functions: internal functions of vortex lattice solver
Public functions: acessible by external codes
PUBLIC VARIABLES
Argument variables:
- flight conditions
- position/orientation body
- dt, number time steps
- flags aeroelastic and unsteady
Output variables:
- loads over panels
- quarter points coordinate
- induced velocities
PUBLIC FUNCTIONS
- constructors /destructor
- initialization functions
- VLM/UVLM solver
- functions to calculate induced velocities at external points

Figure 3.3: Vortex Lattice library architecture overview

ning isolated for pure aerodynamic solutions, then a third file called `grid.xml` and containing the model grid should be provided (points provided from root to tip, from trailing edge to leading edge).

Besides the input files, there are input quantities that may need to be updated at each step for the aeroelastic coupling or directly determined by an external structural solver. They are defined as public variables (argument variables) and can be modified by an external code (in this case, the interface between UM/NAST and the Vortex Lattice solver). Additionally, they may be provided as arguments of the public functions calls.

The outputs are also provided by using public variables. In the Vortex Lattice class, they correspond to the loads over the panels, quarter points coordinates and, if propellers are present, the induced velocities at propellers control points (blades and particles). Additionally, if desired, there is the option for saving an output .h5 file containing several quantities like lift distribution, panels loads, delta C_p for each member and global aerodynamic coefficients (in wind and body frame).

Table 3.1: Inputs defined in Vortex Lattice XML input file

Parameters for each lifting surfaces	
chordwise discretization	number of panels in chordwise direction
spanwise discretization	number of panels in spanwise direction
type of chordwise discret.	linear (0) or sine spacing (1)
type of spanwise discret.	linear (0) or sine spacing (1)
flag constant airfoil	flag to specify if member has constant (1) or variable (0) airfoil
theta_0 spanwise	intial angle interval if spanwise sine spacing is selected
theta_end spanwise	final angle interval if spanwise sine spacing is selected
theta_0 chordwise	intial angle interval if chordwise sine spacing is selected
theta_end chordwise	final angle interval if chordwise sine spacing is selected
flapping panels chordwise	If control surface exist: initial and final index of chodwise panels
flapping panels spanwise	If control surface exist: initial and final index of spanwise panels
airfoil at each section	airfoils names for each section (just 1 string for cte airfoil)
General parameters	
flag control surface mode	turn on (1) or turn off (0) control surface
flag set symmetry	if half of a symmetric model is provided, (1); otherwise, (0)
flag flapping panels	indicate existence of control surfaces; (1) if CS exist, (0) otherwise
max number of wake rows	maximum number of unsteady wake rows
steady aero residual	tolerance used for the steady vortex lattice solver
steady wake extension	extension of the panels of the steady wake
flag second order circ diff	(1) if 2nd is desired for circulation differentiation, (0) if 1rs order
flag save output h5	(1) to save .h5 output, (0) otherwise
Aerodynamic coefficients parameter	
reference area	reference area for calculation of aerodynamic coeff.
reference length	refence length for calculation of longitudinal moment coeff.
reference length lateral	refence length for calculation of lateral moment coeff.
Drag parameters	
flag for induced drag	(1) to calculate induced drag, (0) otherwise
flag for viscous drag	(1) to calculate estimate viscous drag, (0) otherwise
Gust parameters	
flag gust on	(1) if gust is active, (0) otherwise
gust switch	if = 0, gust is on/off based on t_start/end; if = 1, based on position pts
flag gust moving	(1) if gust is moving with freestream, (0) otherwise
z	y’ for lateral gust; ’z’ for vertical gust
string gust type	currently: ’1-cos’ or ’1-cos_time’ or ’sharp_edge’ or ’DARPA’
gust u_ds	design velocity (max magnitude) of the gust
gust h	gust wave length
gust x_0	chordwise position of end of gust region
DARPA gust lambda	parameter related to the spanwise wave length of DARPA gust
DARPA gust phi	Param. of DARPA gust to vary the disturbance in spanwise direction
t start	time gust starts (for ’1-cos_time’ gust or gust switch = 0)
t stop	time gust ends (for ’1-cos_time’ gust or gust switch = 0)
General propeller parameters	
flag propeller on	(1) if propeller exists, (0) otherwise
propeller solver name	Currently one option: ’VVP_propeller’
propeller sub steps	number of propeller sub time steps per Vortex Lattice time step
Parameters defined for each propeller (if UM/NAST coupling)	
reference member	index of member to which propeller is attached (reference member)
reference node	index of node (inside reference member) where propeller is attached
hub local y	y coord. of hub position wrt UM/NAST local coord. at reference node
hub local z	z coord. of hub position wrt UM/NAST local coord. at reference node
prop. inclination with local x	angle by which the prop frame may be rotated wrt local ref node x-axis
prop. inclination with local y	angle by which the prop frame may be rotated wrt local ref node y-axis
prop. inclination with local z	angle by which the prop frame may be rotated wrt local ref node z-axis
prop inclination axis order	int of 3 digits (1 for x, 2 for y or 3 for z) with order of the rotations

Finally, external and customized libraries are also used, as indicated in Figure 3.3. With the exception of Eigen library, all other third party libraries are already included in the UVLM/Propeller module for convenience and robustness of installation. The libraries hdf5, hdf5-eigen, szip and zlib are used for the output process of .h5 files just described, Eigen is used for matrix operations and general linear algebra involved, and the pugiXML library is necessary for reading the XML input files.

The customized libraries are: i) `xml_routines.h`: used for the reading of XML input files; it was ported from UM/NAST and slightly adapted to remove dependencies and allow general use for other solvers; ii) `math_tools.h`: contains additional math tools not available on Eigen. It was also ported from UM/NAST, removing dependencies for general use, and new functions were added. iii) `gust.h`: contains simple functions to calculate the effect of different gust types on a given point and can be used by different solvers.

3.3.1.2 Initialization Functions

Figure 3.5 illustrates the initialization of the Vortex Lattice solver. A summary of the steps is given in the execution order:

- Build grid metrics (function `BuildGridMetrics`): this step calculates and stores in internal variables metrics associated with the grid, like control points and quarter points coordinates, panels' quarter chord line and normal vectors, panels' areas, among other quantities. Those grid-related quantities are essential for the calculation of induced velocities and panels' loads;

- Initialize panels circulation: initializes with zeros the panels' circulation and previous circulations (time $t-dt$, and $t-2dt$) matrices. The previous circulations are updated as the simulation evolves and are used in the calculation of unsteady panels' loads;

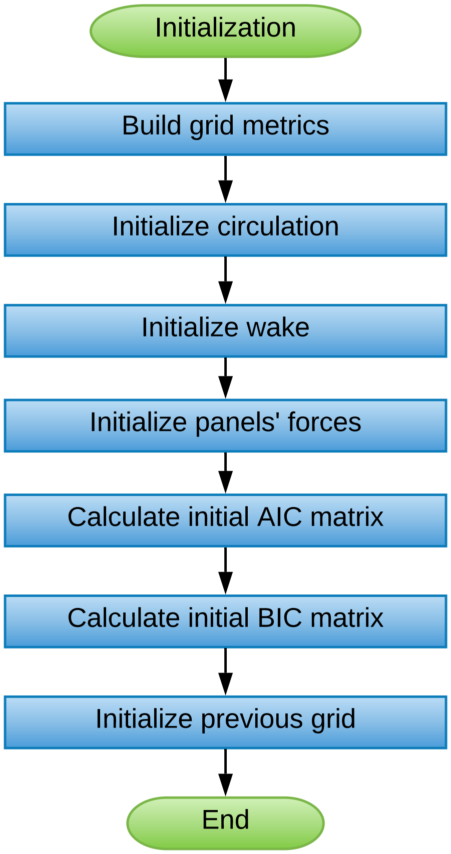


Figure 3.4: Flowchart illustrating Vortex Lattice solver Initialization

- Initialize wake (function `InitializeWake`): initializes matrices containing connectivity information of the points defining each vortex ring element of the wake, initialize long wake panels if the simulation is steady and initialize wake circulation with zeros;

- Initialize panels' forces: initializes matrices containing panels' loads and previous panels' loads (previous static iteration) with zeros; the previous panels' loads are used for the convergence criteria of the steady Vortex Lattice solver;

- Calculate initial AIC matrix (function `AICMatrix`): calculates the aerodynamic influence matrix (AIC) used to solve the non-penetration boundary condition for the panels' circulations;

- Calculate initial BIC matrix (function `BICMatrix`): calculates the BIC matrix, used to determine the contribution of streamwise segments of bound vortex rings in the induced drag calculation;

- Initialize previous grids: initializes with zeros matrices that contain previous grid coordinates (at times $t - dt$, and $t - 2dt$). Those matrices will be updated and used in the case the solver is unsteady for the calculation the deformation induced velocity on the panels' control points.

If the aeroelastic coupling is considered, a similar, however shorter version of this initialization is called at each new time step (or static iterations): function `UpdateMetricsAeroelastic`. This function updates the grid metrics, as well as AIC and BIC matrices, based on the current grid configuration.

3.3.1.3 Steady Vortex Lattice Solver

As discussed in the formulation part, the solution for steady cases makes use of long wake panels modeling. This is justified by the fact that, for steady conditions, all the wake panels in chordwise direction would virtually have the same circulation, behaving as one long panel. Then, this approach provides equivalent results as a discretized wake model, at a much smaller cost. The workflow of the steady Vortex Lattice solver is illustrated in Figure 3.5.

First, if the model has control surfaces and they are active, a function to rotate the normal vectors of the panels corresponding to the control surface region is called (`RotateNormalVectorsControlSurfaces`). A loop is then initiated until the panels' loads converge and the steps described as follow are performed:

- Calculate rigid body velocity contribution (function `RigidBodyInducedVelocity`): takes into account contribution of body translational and rotational velocity on the effective velocity at the panels' control points;

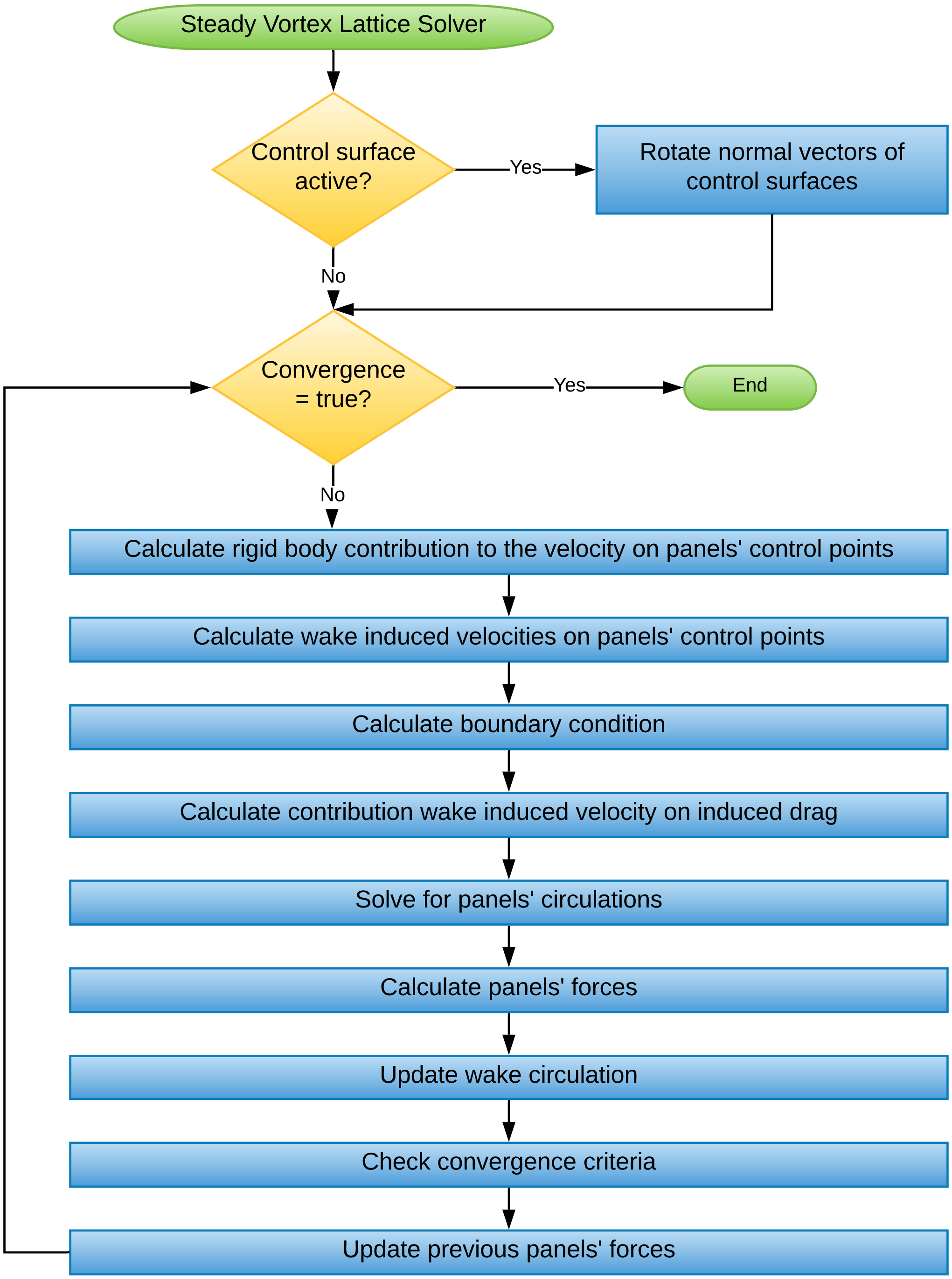

Steady Vortex Lattice Solver
Control surface active?
Yes
Rotate normal vectors of control surfaces
No
Convergence = true?
Yes
End
No
Calculate rigid body contribution to the velocity on panels' control points
Calculate wake induced velocities on panels' control points
Calculate boundary condition
Calculate contribution wake induced velocity on induced drag
Solve for panels' circulations
Calculate panels' forces
Update wake circulation
Check convergence criteria
Update previous panels' forces

Figure 3.5: Flowchart illustrating steady Vortex Lattice solver

- Calculate wake induced velocities (function `WakeInducedVelocity`): based on the current wake circulation and control points positions, calculates induced velocity of wake at the panels' control points.

- Calculate wake boundary condition (function `CalculatingBoundaryCondition`): calculates and adds normal components of the velocities due to rigid body motion and induced by the wake, in order to be used by the non-penetration condition system;

- Calculate contribution of wake induced velocity on induced drag (function `WakeInducedVelocityInducedDrag`): calculates the wake induced velocities at panels quarter points. This will be added to the induced velocity of streamwise bound vortex rings segments in order to determine the induced drag;

- Solve for panels circulations (`SolveCirculation`): solves the linear system given corresponding to the non-penetration condition;

- Calculate panel forces (`CalculatePanelForcesSteady`): calculates the aerodynamic loads due to panels circulations, and, if desired, the induced drag and an estimation of the viscous drag;

- Update wake circulation (`UpdateWakeCirculation`): for the steady solver, corresponds to the shedding of circulation from the trailing edge panels of all surfaces to the long wake panels;

- Check convergence criteria (`CheckingSteadyAeroConvergence`): checks if differences between current panels' loads and panels' loads on previous iterations are below a user-defined tolerance. If this is the case, the converged flag will be set to true, and the loop terminates.

- Update previous panels' forces: update current panels' forces for the convergence criteria checking in the next step;

For the aeroelastic coupled solution, the loop described in Figure 3.5 is called as many times as necessary for the convergence of the deformed geometry (equilibrium state).

3.3.1.4 Unsteady Vortex Lattice Solver

The unsteady loop has many similarities with the steady one, but some additional steps and peculiarities are present. Here, the focus is given to the main differences.

Figure 3.6 presents an overview of the unsteady loop. Now the loop is determined by the required number of time steps. For the aeroelastic coupling, each call of the unsteady solver will perform one step (with time provided as an argument for the loop).

As in the steady solution, if there are active control surfaces in the model, the corresponding panels' normals are rotated by the desired angle using the function `RotateNormalVectorsControlSurfaces`. The contribution of rigid body motion and wake induced velocity on the effective velocity at the control points are them determined by the functions `RigidBodyInducedVelocity` and `WakeInducedVelocity`.

Different from the steady solution, however, new velocity contributions are now considered: the contribution of gust, if active (function `GustInducedVelocity`) and the contribution of the deformation of the lifting surfaces, if coupled to the structural solver (function `DeformationInducedVelocity`). Also, although not apparent in the flow chart, propellers induced velocities are taken into account by updating public variables containing propeller velocity influence. This is done outside the loop, by the interface between aerodynamic and propellers, as will be presented soon.

Now, function `CalculatingBoundaryCondition` also incorporates the normal components of the additional velocities mentioned. The contribution of the wake panels induced velocity on induced drag (`WakeInducedVelocityInducedDrag`) and the solution of the linear system (`SolveCirculation`) is performed similarly as for the steady loop. The next steps have more differences and are listed below:

- Calculate panels loads (function `CalculatePanelForces`): besides the steady contribution (calculated based on Kutta-Jowkowski condition, in the same way as for the steady case) and the possible drag additions, the unsteady contri-

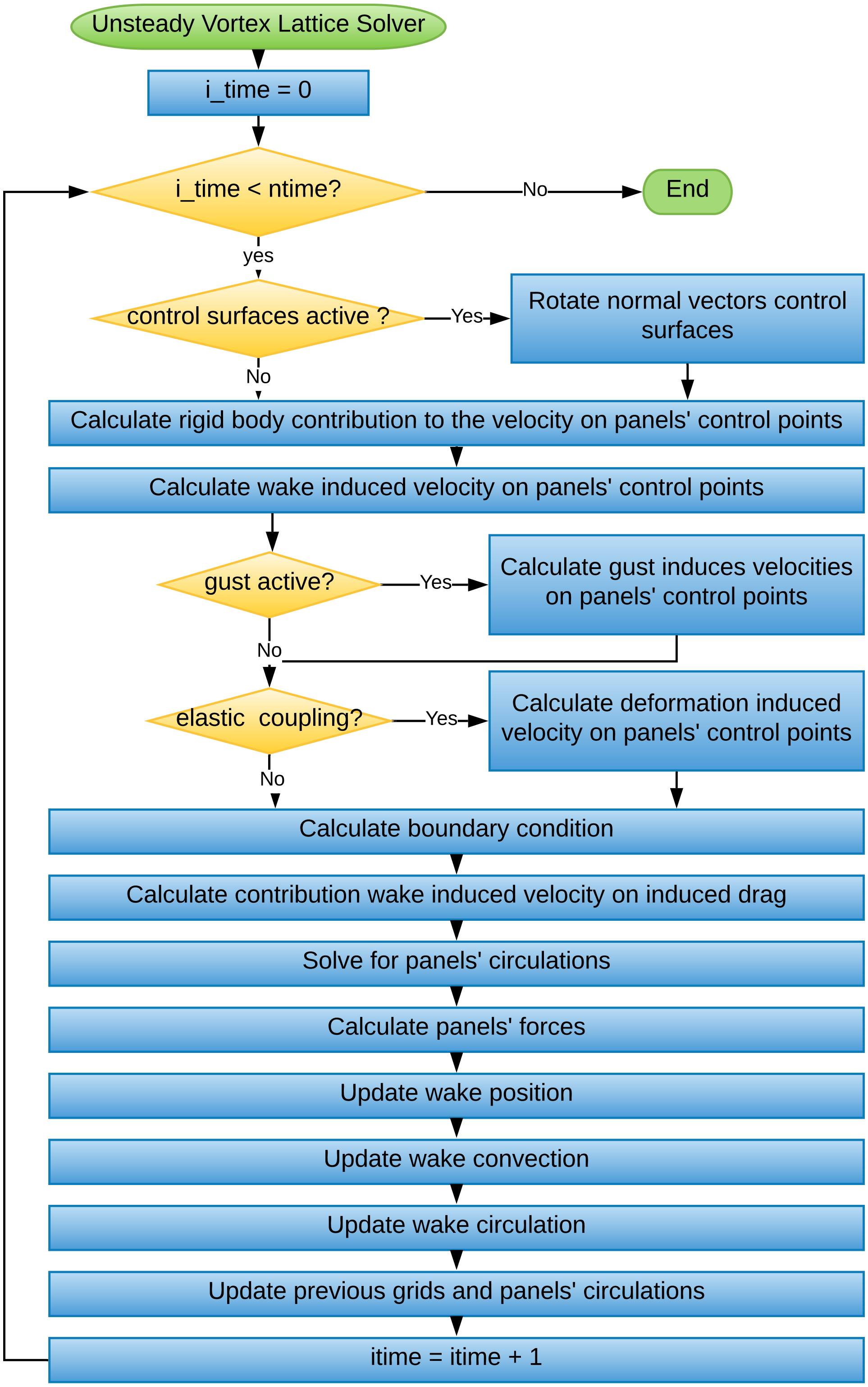

Unsteady Vortex Lattice Solver
i_time = 0
i_time < ntime?
No
End
yes
control surfaces active ?
Yes
Rotate normal vectors control surfaces
No
Calculate rigid body contribution to the velocity on panels' control points
Calculate wake induced velocity on panels' control points
gust active?
Yes
Calculate gust induces velocities on panels' control points
No
elastic coupling?
Yes
Calculate deformation induced velocity on panels' control points
No
Calculate boundary condition
Calculate contribution wake induced velocity on induced drag
Solve for panels' circulations
Calculate panels' forces
Update wake position
Update wake convection
Update wake circulation
Update previous grids and panels' circulations
itime = itime + 1

Figure 3.6: Flowchart illustrating unsteady Vortex Lattice solver

bution due to variation of circulation with time is taken into account. Also, the new velocity contributions (gust, deformation velocity, and propeller) are included in the loads' calculation, if they are present;

- Update wake position (function `UpdateWakePosition`): shed new panels into the wake, updating wake vortex corner points and the wake connectivity matrices;

- Update wake convection (`UpdateWakeConvection`): wake is convected with the onflow velocity (contribution due to body translational and rotational velocities and gust perturbations). Although not incorporated in the current C++ version, other influences on wake convection can be included as well, allowing wake relaxation. As previously commented, it was observed in simulations of representative VFA that the effects of wake relaxation are usually not significant [109,110] (while it increases the computational cost considerably);

- Update wake circulation (`UpdateWakeCirculation`): new entries, corresponding to the circulation shedded from the trailing edge panels into the new wake panels, are added to the wake circulation matrix;

- Update previous grids and panels' circulation: previous values of grid and circulation (at $t - dt$ and $t - 2dt$) are updated for next time step.

3.3.1.5 Steady Vortex Lattice Solver with Propeller

In order to consider the averaged effects of the propellers on the static simulation, a modified steady solver, based on the unsteady solution with propellers, was developed, as illustrated in Figure 3.7. Although it is a steady Vortex Lattice loop, this process is actually located inside the interface between lifting surfaces and propellers (more details soon) and makes use of the Unsteady Vortex Lattice loop just described. The steps are:

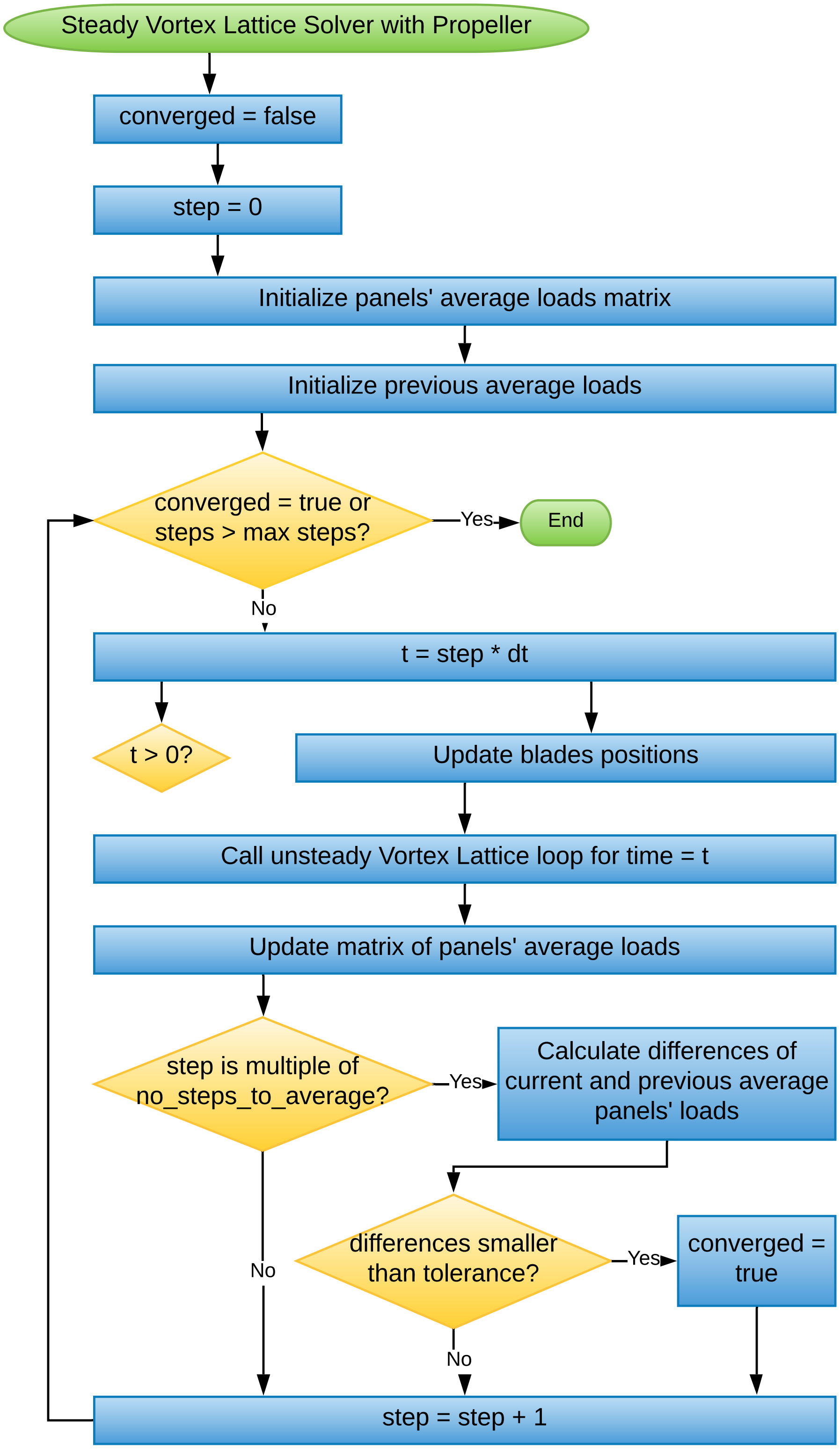

Steady Vortex Lattice Solver with Propeller
converged = false
step = 0
Initialize panels' average loads matrix
Initialize previous average loads
converged = true or steps > max steps?
Yes
End
No
t = step * dt
t > 0?
Update blades positions
Call unsteady Vortex Lattice loop for time = t
Update matrix of panels' average loads
step is multiple of no_steps_to_average?
Yes
Calculate differences of current and previous average panels' loads
No
differences smaller than tolerance?
Yes
converged = true
No
step = step + 1

Figure 3.7: Flowchart illustrating steady Vortex Lattice solver with propeller

- Set flag converged to false and number steps to zero;

- Initialize panels' average loads and previous average loads matrices to zero;

- then, while converged is false, or number of steps is smaller than the maximum, execute the steps described as follow;

- Update pseudo time step;

- Update blades positions (function `UpdateBladesPosition`): update blades' radial unit vector, grids and control points positions based on the current pseudo time step;

- Call unsteady Vortex Lattice loop described in Figure 3.6 (`UnsteadyVortexLattice`) for the current pseudo time step (just one execution of the loop);

- Update matrix of panels' average loads: the average is calculated along a user-defined number of steps, and the arithmetic mean is calculated;

- If the number of steps for averaged is achieved, check convergence: check if the difference of average loads on panels is smaller than a user-defined tolerance;

- Update number steps and continue until convergence or maximum number os time steps is achieved;

3.3.1.6 Additional Public Functions

In order to allow the coupling with propellers, additional public functions are defined and called appropriately inside the interface between lifting surfaces and propeller aerodynamics:

- `InducedVelocitiesPanelsOnSpecifiedPoints`: calculates the panels induced velocities over a given set of points (in this case, propeller blades control points and particles positions);

- `WakeInducedVelocityOnSpecifiedPoints`: calculates the lifting surfaces wake induced velocities over specified points (again, for this case, propeller blades control points and particles positions);

Also, as previously commented, there is an optional function (`SaveOutputH5`) that allows saving several aerodynamic related quantities in a .h5 file, (*e.g.* lift distribution, panels loads, delta C_p, global aerodynamic coefficients).

3.3.2 Lifting Line + Viscous Vortex Particle Propeller Library

The Lifting Line plus Viscous Vortex Particle Propeller library consists of a C++ class and can be used for isolated propellers aerodynamic analysis (single or multiple propellers) as well as for the coupling with Vortex Lattice (which may also be coupled to the aeroelastic solver). Its main goal is to calculate the resultant aerodynamic loads at propeller hub (that is transferred to the structural solver, for the aeroelastic coupling) and the influence of propeller blades and wake on the velocity flow-field, affecting the lifting distribution in the surfaces around. More details about the code are presented in the next subsections.

3.3.2.1 Architecture Overview

An overview of the Lifting Line plus Viscous Vortex Particle Propeller library architecture is provided in Figure 3.8. Similarly as for the Vortex Lattice library, it contains private variables and functions, for internal use of this class only, and public variables and functions, accessible from external code. The use of public variables (categorized as argument and output variables) allows efficient exchange of information between this class and external code.

Two input files are expected: i) a file named `input_VVP.xml`, described in Ta-

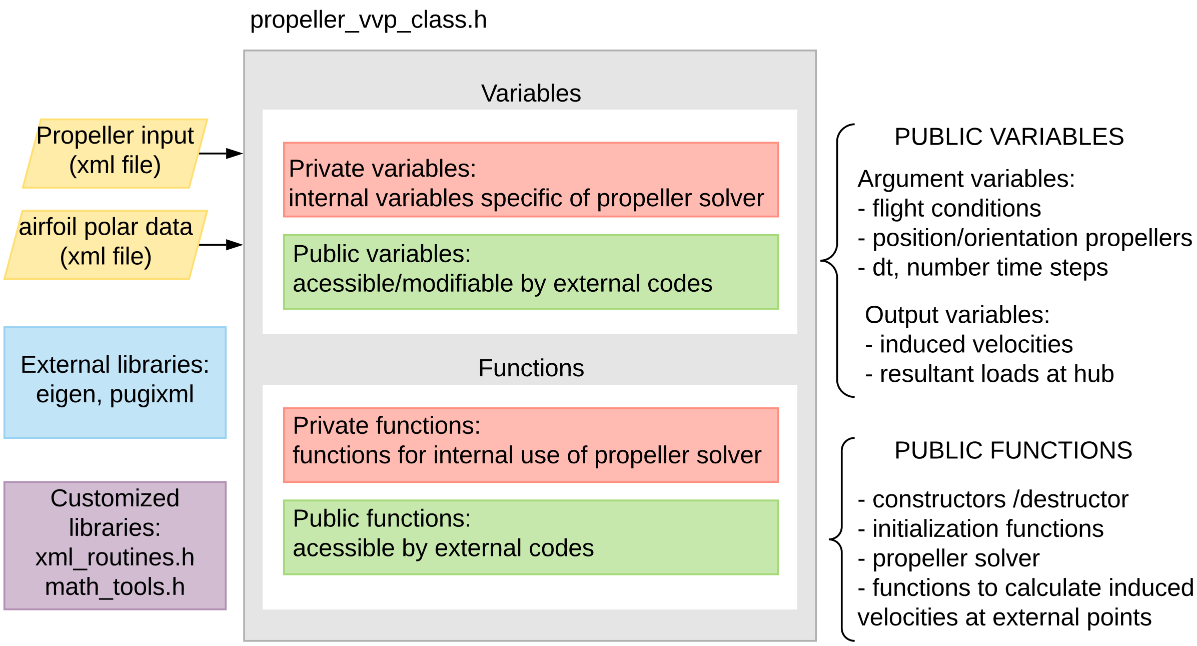


Figure 3.8: Lifting Line plus Viscous Vortex Particle Propeller library architecture overview

ble 3.2, containing information about propeller geometry and parameters required for the LL/VVPM, and ii) an XML file (with a user-defined name) containing polar data of the airfoils used along the blade, for a range of Reynolds number and angle of attack. If the vectors containing the values of Reynolds numbers (Re) and angle of attack (AoA) have sizes N_{Re} and N_{AoA}, respectively, then matrices with size $N_{Re} \times N_{AoA}$ need to be provided for the drag, lift and moment aerodynamic coefficients.

As for the Vortex Lattice class some input quantities may vary along the simulation steps (as propeller positions and flight conditions) or are defined in another code, (.g. time step), and are provided by the use of public argument variables, being directly accessed and modified by an external code. Additionally, some inputs are provided as arguments of the public functions calls.

Outputs to other solvers are provided by public output variables, being directly accessed and read by external code. In this case, they correspond to the induced velocities at the surfaces and the resultant loads at propeller hub.

Table 3.2: Inputs defined in propeller XML input file

Parameters for each propeller	
hub position	initial position of propeller hub in body frame
rotation (RPM)	(+) sign if counterclockwise and (-) if clockwise (looking in front of prop)
rotation direction	clockwise’ or ’counterclockwise’ (looking in front of prop)
number segments per blades	spanwise discretization of blades
number blades	quantity blades
blades initial ang. position	in degree, defined wrt positive prop frame spanwise direction
properties blades	user defined string labelling the blades properties associated
flag constant airfoil	flag to specify if blades have constant (1) or variable (0) airfoil
radius	propeller radius
mass	mass of propeller blades (for gyroscopic function)
Blades properties	
twist distribution (beta)	angle between blades and plane of rotation for each section (degree)
r_R ratio	(radial position)/(propeller radius) for each section along the blade
c_R ratio	(local chord)/(propeller radius) for each section along blade
airfoil table name	vector with names of the associated airfoil XML table for each section
General parameters	
number sub time steps	number of propeller time steps per external aero time step (e.g. uvlm)
flag shed particles included	(1) to include shed particles, (0) otherwise
resolution	characteristic length associated to typical distance among particles
factor c_sigma	factor that multiplied by resolution provides particle core radius, sigma
number source seg per blade	number of blade segments (from tip to root) that will release particles
cut off distance	cut off distance to eliminate particles from calculations
Optional parameters to measure velocity profile behind propeller	
flag measurements on	(1) to enable measurement of velocity profile, (0) otherwise
line x	coordinate x (chordwise) of line of measurement wrt propeller frame
line z	coordinate z (vertical) of line of measurement wrt propeller frame
line y min	minimum y coordinate of line wrt propeller frame
line y max	maximum y coordinate of line wrt propeller frame
number measurements	number of points along line of measurements

Again, external and customized libraries are used. The external libraries Eigen and pugixml are used for matrix operations/linear algebra and reading the XML files, respectively; Eigen is supposed to be already installed in the system. The customized libraries are `xml_routines.h` and `math_tools.h`, as previously described. In the current C++ version, gust disturbances are not included in the propeller library, but this can be done similarly as for the Vortex Lattice class.

3.3.2.2 Initialization Functions

There are three public functions involved in the propeller initialization:

i) `SetParticleVortexPropeller`: this function is used to read the input data provided in the XML file in the respective class variables;

ii) `ProcessParticleVortexPropeller`: this function process the XML data read into other input variables, as the total numbers of propellers, grid points and propeller segments; the number of particles per time step; the values of blade twist and chord at control points; the radial position of grid points and length of segments;

iii) `Init`: after the previous functions attribute the corresponding values to the input variables, this function initializes the propeller solver, as illustrated in Figure 3.9. It initializes the blades angular position, the grids coordinates and blades control points in propeller and body frame, and the blade radial unit vector in propeller frame.

Also, there is a public function named `UpdateBladesPosition`, that allows an external solver to update the blades grids, control points and radial unit vectors (useful, for example, for the interface with Vortex Lattice).

3.3.2.3 Propeller Solver

Figure 3.10 illustrated the workflow of the propeller solver. The following steps are executed:

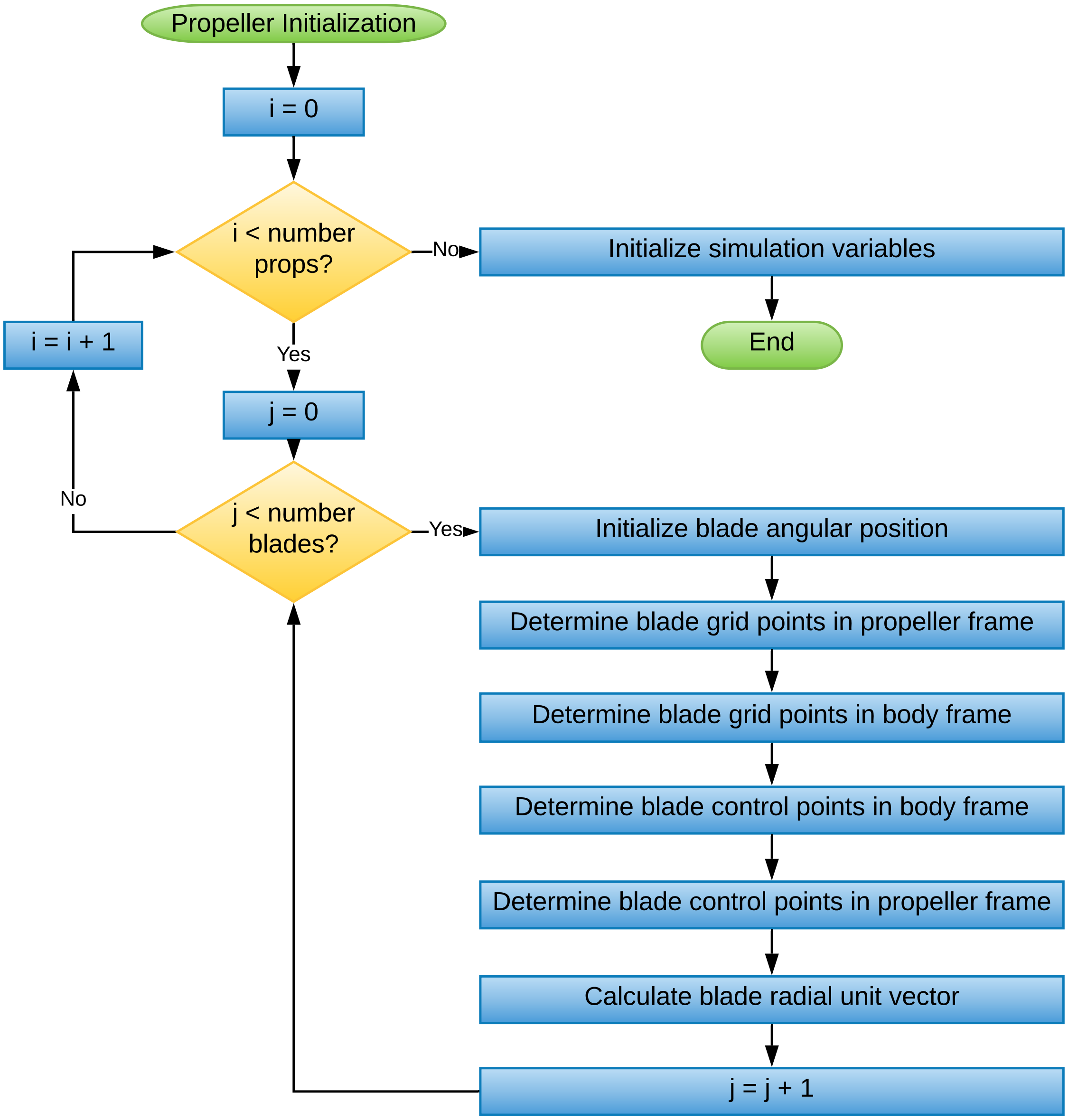

Propeller Initialization
i = 0
i < number props?
No
Initialize simulation variables
End
i = i + 1
Yes
j = 0
No
j < number blades?
Yes
Initialize blade angular position
Determine blade grid points in propeller frame
Determine blade grid points in body frame
Determine blade control points in body frame
Determine blade control points in propeller frame
Calculate blade radial unit vector
j = j + 1

Figure 3.9: Flowchart illustrating propeller initialization function

- While time step is smaller than the user-defined number of iterations, a loop with the steps described in the following items is executed. For the coupling with lifting surfaces aerodynamics (and the coupled aeroelastic-flight dynamics framework), each call of the propeller solver will repeat the loop as many times as the user-defined number of sub time steps (with updated initial time provided as an argument);

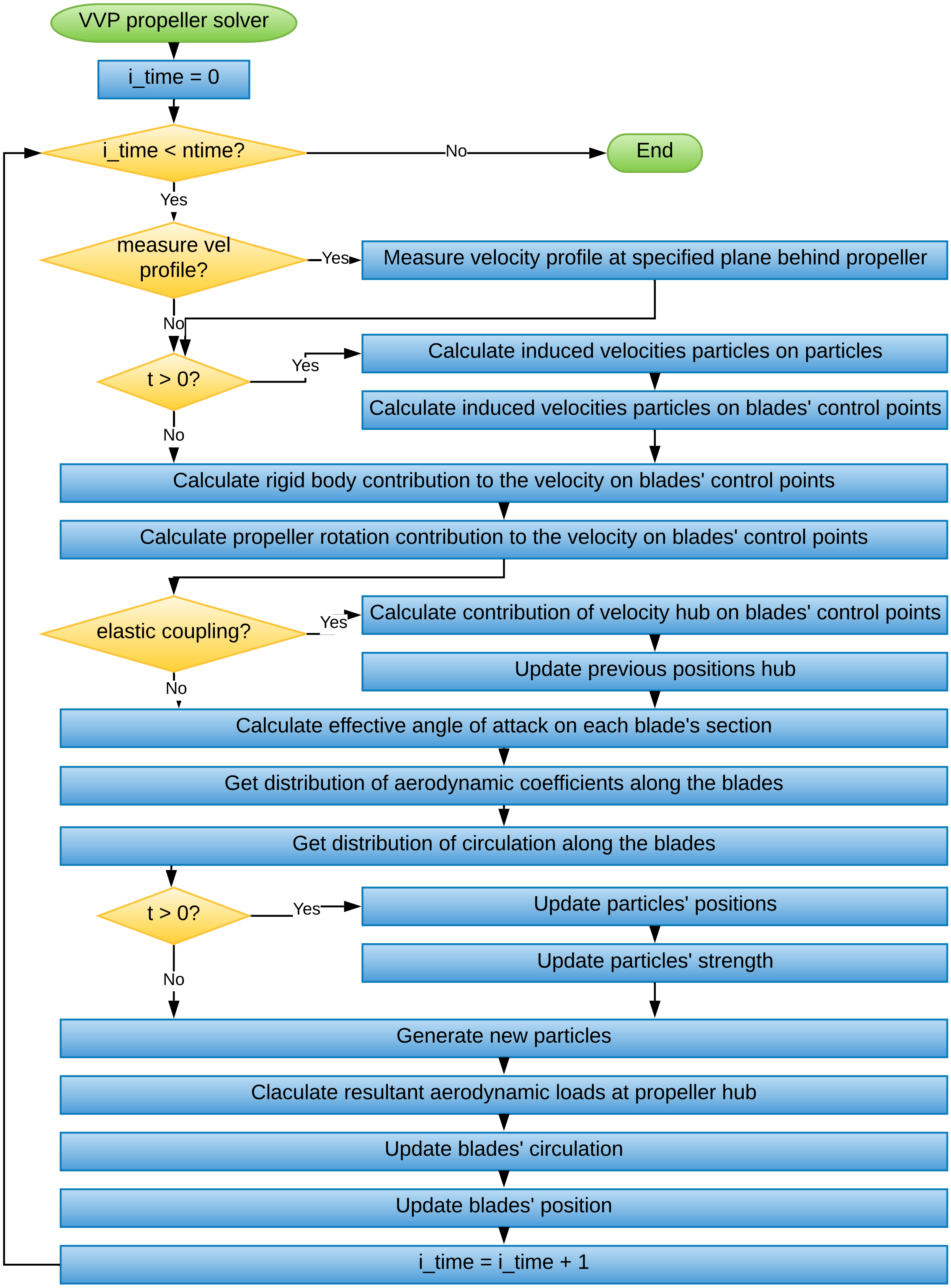

VVP propeller solver
i_time = 0
i_time < ntime?
No
End
Yes
measure vel profile?
Yes
Measure velocity profile at specified plane behind propeller
No
t > 0?
Yes
Calculate induced velocities particles on particles
Calculate induced velocities particles on blades' control points
No
Calculate rigid body contribution to the velocity on blades' control points
Calculate propeller rotation contribution to the velocity on blades' control points
elastic coupling?
Yes
Calculate contribution of velocity hub on blades' control points
Update previous positions hub
No
Calculate effective angle of attack on each blade's section
Get distribution of aerodynamic coefficients along the blades
Get distribution of circulation along the blades
t > 0?
Yes
Update particles' positions
Update particles' strength
No
Generate new particles
Claculate resultant aerodynamic loads at propeller hub
Update blades' circulation
Update blades' position
i_time = i_time + 1

Figure 3.10: Flowchart illustrating VVPM propeller solver

- If velocity profile measurement is desired, measure velocity behind propeller by the function call `MeasurementVelocitiesPropellerWake`; this function measures the components of velocity due to propeller along a straight line at a user-defined distance behind the propeller; although not currently ported to C++, in the Python version it was also implemented an alternative function to measure the velocity profile along a circumference behind the propeller;

- If time is greater than zero (and particles already exist), calculate induced velocities of the particles on themselves (`InducedVelocitiesParticlesonParticles`). This function was implemented such that the cost, without parallelization, is $O(N(N+1)/2)$, where N is the number of particles; Also, calculate particles induced velocities on blades `InducedVelocitiesParticlesonControlPoints` at control points positions;

- calculate the rigid body motion (translation and rotation) contribution to the velocity on blades' control points (function `CalculateRigidBodyInducedVelocity`); for the isolated propeller, this corresponds to the freestream velocity;

- Calculate the contribution of the propeller rotation on the velocity of blades' control points (function `CalculatePropellerRotationInducedVelocity`);

- If the code is coupled to external elastic solver, calculate the contribution of hub motion (function `CalculateVelocityHubWithRelationBody`) to the blades control points velocity and update previous hubs positions at $t - dt$ and $t - 2dt$ for the next step. The hub velocity is calculated by the numerical derivative of its position in body frame, using a second-order backward scheme;

- Calculate effective angle of attack (function `CalculateAoaEffective`) for each section: the total velocity at the blades control points due to due to rigid body motion (or freestream), propeller rotation, particles influence, elastic deformation (if aeroelastic coupling) and external induced velocities due to the presence of lifting surfaces are first calculated. For the latter, a public variable containing the external induced velocities on blades is updated by an external solver (propeller and lifting surfaces interface), in this case accounting for the influence of surfaces panels and wake panels. Then, taking into account the local blade twist, the local angle of attack at each control point section is determined;

- Get distribution of the aerodynamic coefficients along the blades (function `GetAeroCoeficientsDistribution`): calculates lift, drag and moment coefficients at each blade section based on effective local angle of attack, local Reynolds number and the respective airfoil polar table provided;

- Get circulation distribution (`function GetCirculationDistribution`): gets circulation distribution based on local lift coefficient c_l and by applying 2D Kutta-Joukowski theory for each control point;

- If time is greater than zero (particles exist), update particles positions with function `UpdateParticlesPositions`: updates particles positions with relation to body frame for the next step by convecting particles by the local flow velocity. Takes into account velocity contributions due to body motion (or freestream), self-influence of particles, the influence of blades and possible external induced velocities due to the presence of lifting surfaces (again updating the correspondent public variable in an external interface code);

- Also, if particles exist, update particles strength (`UpdateParticlesStrength`): updates the particles' strength for next step taking into account stretching and viscous effects as described in the formulation;

- Generate new particles (`GenerateNewParticles`): following the conservation of vorticity, new particles are generated for the next step, shed particles (due to circulation change with time) and trailing particles (due to gradient of circulation along blade spanwise direction). In order to reduce the computational cost, the following options are available in this function: i) neglecting shed particles (this can be a choice when shed particles strength are much smaller than trailing particles strength), ii) controlling number or particles per blade by defining the of 'number of source segments' (number of blade segments, from tip to root, which release particles. Particle at hub is always included), and, finally, iii) applying a cut off distance criteria to remove particles far away from the region of interest;

- Calculate resultant aerodynamic loads at hub (`CalculatePropellerLoads`): calculate and integrate loads at each blade segment based on the sectional component of total local velocity and the distribution of aerodynamics coefficients; thrust and power coefficients associated to each propeller are also determined;

- Update blades circulation `UpdateCirculation`: update previous values of blade circulation at $t - dt$ and $t - 2dt$ for the next step;

- Update blades position (`UpdateBladesPosition`): updates blades angular positions, grids, control points and radial unit vectors for the next step;

- Increment time step and continue the process until the number of steps is achieved (for aeroelastic coupling with aerodynamics or coupled aeroelastic-flight dynamics framework, this number corresponds to the number of sub time steps).

3.3.2.4 Additional Public Functions

Similarly as for the Vortex Lattice class, additional public functions are defined to allow the coupling between the LL/VVPM propeller and another approach for the lifting surfaces aerodynamics:

- `InducedVelocitiesParticlesonEvaluationPoints`: calculates particles induced velocity at a set of points provided as function arguments;
- `CalculateInducedVelocitiesBladesonEvaluationPoints`: calculate the induced velocity due to blades in a specified set of points.

3.3.3 Gyroscopic Loads Function

The gyroscopic loads function is implemented in C++ in the UM/NAST version 2 and is currently in process of being integrated with UM/NAST 4.0. It was initially implemented inside UM/NAST code, taking the propeller inputs from the UM/NAST input itself. In the new configuration, it will be placed in the propeller module, taking advantage of the input file for the propeller solver, as the propeller data are shared for both codes. Figure 3.11 presents an overview of the gyroscopic loads function.

For each propeller and each blade, the following steps are performed:

- For this step, and based on Eq. 2.75, the following relation between the transformation matrix from body to local frame M_{B2W} is applied:

  $$\frac{d(M_{B2W})}{dt} =^{B} \tilde{\omega}^{W} M_{B2W} \tag{3.1}$$

  where ${}^{B}\omega^{W}$ is the angular velocity of the local frame with relation to the body frame, and ${}^{B}\tilde{\omega}^{W}$ is its associated skew-matrix.

  Then, the numerical derivative of the transformation matrix from local to the body is first determined. Based on the known transformation matrix from local to body M_{B2W} and its derivative calculated numerically $d(M_{B2W})/dt$, the skew-matrix ${}^{B}\tilde{\omega}^{W}$ is calculated and its associated vector, the angular velocity of the local frame with relation to the body frame, is obtained;

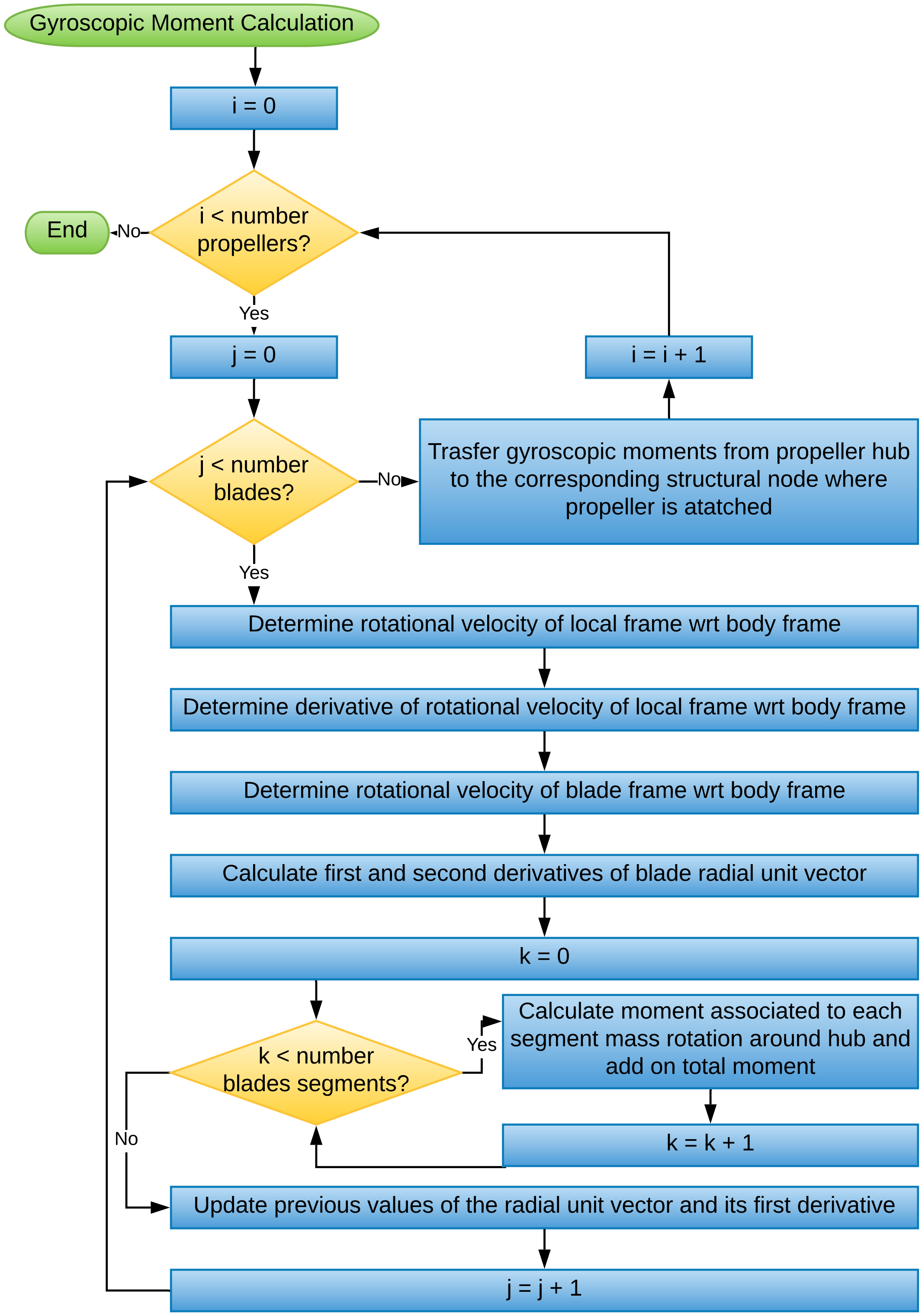

Gyroscopic Moment Calculation
i = 0
i < number propellers?
No
End
Yes
j = 0
i = i + 1
j < number blades?
No
Trasfer gyroscopic moments from propeller hub to the corresponding structural node where propeller is atatched
Yes
Determine rotational velocity of local frame wrt body frame
Determine derivative of rotational velocity of local frame wrt body frame
Determine rotational velocity of blade frame wrt body frame
Calculate first and second derivatives of blade radial unit vector
k = 0
k < number blades segments?
Yes
Calculate moment associated to each segment mass rotation around hub and add on total moment
No
k = k + 1
Update previous values of the radial unit vector and its first derivative
j = j + 1

Figure 3.11: Flowchart illustrating function that calculates gyroscopic moment

- Determine derivative of the rotational velocity of the local frame with relation to body frame: performs numerical derivative of ${}^{B}\omega^{W}$, using a second-order backward scheme;

- Determine rotational velocity of blade with relation to body frame: determine the rotational velocity of blade with relation to propeller ${}^{P}\omega^{b}$ and add to the rotational velocity of local frame with relation to body (remembering that, as the propeller frame was defined not to rotate with relation to the local frame in Chapter 2, ${}^{B}\omega^{W}$ is the same as ${}^{B}\omega^{P}$), obtaining the rotational velocity of blade with relation to body frame ${}^{B}\omega^{b}$.

- Calculate first and second derivatives of blade radial unit vector: obtained based on the relations given by Eq. 3.1, using the rotational velocity of the blade with relation to the body;

- Loop blades segments and calculate and integrate gyroscopic moment based on Eqs. 2.89, 2.86 and 2.87;

- Finally, after looping all blades for a given propeller, the associated gyroscopic moment of the considered propeller is transferred to UM/NAST structural node where the propeller is attached (reference node).

3.3.4 Propeller and Lifting Surfaces Aerodynamics Interface Library

The propeller and lifting surfaces aerodynamics interface is a C++ class conceived for the coupling of a general lifting surface aerodynamics solver and a general propeller aerodynamics solver. The class is templated in terms of an aero object (a C++ class containing the lifting surface aerodynamic solver) and a prop object (C++ class containing the propeller aerodynamic solver).

As illustrated in Figure 3.12, for each solver (propeller or lifting surfaces), control points are defined (points where induced velocity needs to be calculated) and induced velocities are provided for the control points of the other solver. For the LL/VVPM propeller solver, for example, the propeller solver control points are the blades control points and particles positions. For the Vortex Lattice, in this case, the solver control points are the panels control points. The interface calls the appropriate solvers and intermediates the communication by accessing and modifying standard public variables for each solvers' control points and induced velocities.

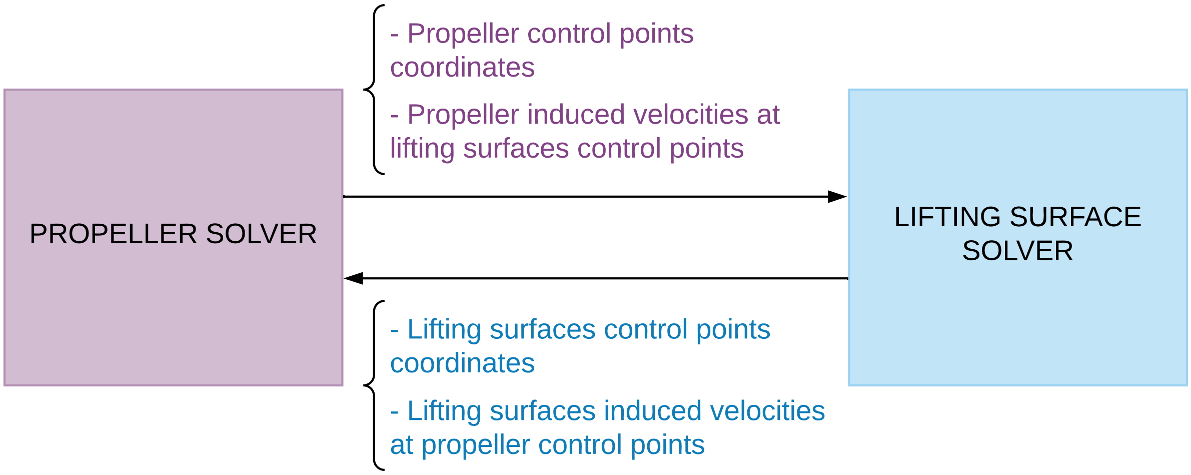

- Propeller control points coordinates
- Propeller induced velocities at lifting surfaces control points
PROPELLER SOLVER
LIFTING SURFACE SOLVER
- Lifting surfaces control points coordinates
- Lifting surfaces induced velocities at propeller control points

Figure 3.12: Basic idea of a general aerodynamic interface between propellers and lifting surfaces aerodynamics

If one of the definitions above does not apply for some of the methods, the respective variables are left empty, not influencing (or being influenced) by the other solver. In the following sections, more details are presented about the propeller and lifting surfaces interface class.

3.3.4.1 Architecture Overview

Figure 3.13 presents an overview of the propeller and lifting surfaces interface class. In this case, no input file is directly necessary, but, if applicable, input files need to be defined for the selected propeller and lifting surfaces aerodynamic solvers. Also, as just commented, the propeller and lifting surface solvers are given as arguments of the interface, provided by an external code, which can be a driver for the isolated propeller and lifting surfaces aerodynamics or an aeroelastic interface.

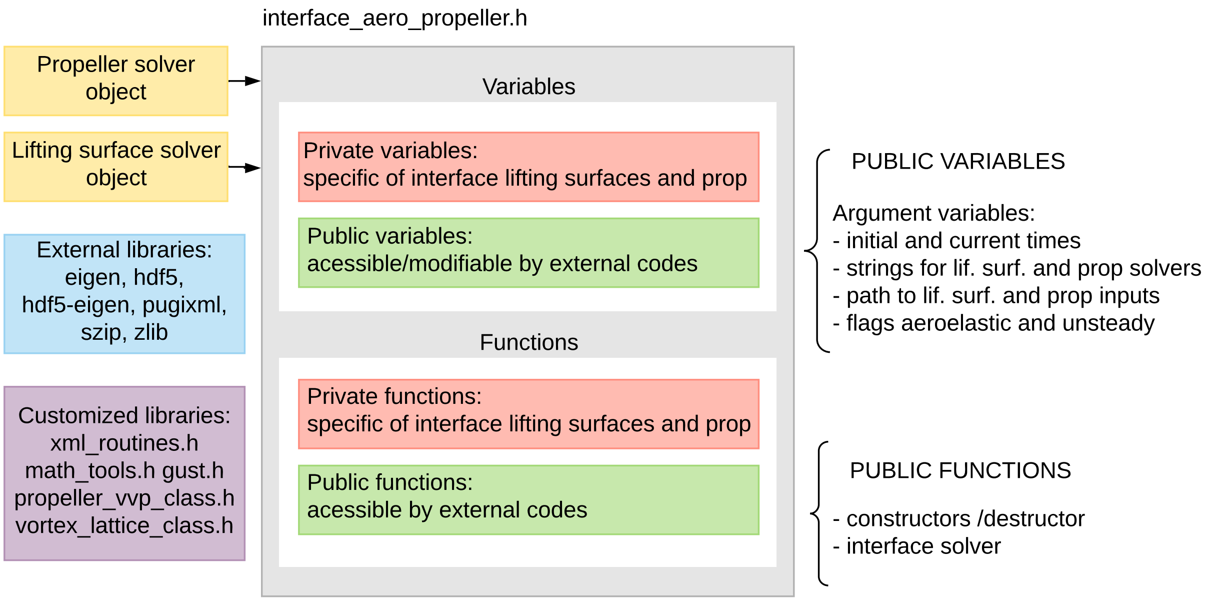

interface_aero_propeller.h
Propeller solver object
Lifting surface solver object
External libraries: eigen, hdf5, hdf5-eigen, pugixml, szip, zlib
Customized libraries: xml_routines.h math_tools.h gust.h propeller_vvp_class.h vortex_lattice_class.h
Variables
Private variables: specific of interface lifting surfaces and prop
Public variables: acessible/modifiable by external codes
Functions
Private functions: specific of interface lifting surfaces and prop
Public functions: acessible by external codes
PUBLIC VARIABLES
Argument variables:
- initial and current times
- strings for lif. surf. and prop solvers
- path to lif. surf. and prop inputs
- flags aeroelastic and unsteady
PUBLIC FUNCTIONS
- constructors /destructor
- interface solver

Figure 3.13: Propeller and lifting surfaces interface architecture overview

No public output variable is defined, but the outputs are indirectly given by the public output variables of the propeller and aerodynamic solvers. Argument variables are defined: path for the possible input files for selected solvers, strings defining solvers, flags to turn on/off aeroelastic coupling and unsteady simulation, initial time and current time. Such variables can be accessed and modified by an external code (in this case, the interface between UM/NAST and Vortex Lattice).

Each option of lifting surface or propeller solver needs to be added as a library in the interface class and their respective function calls added in the corresponding places inside the interface private functions. Also, the external and customized libraries used for each optional solver need to be integrated with this interface. The interface assumes a standard variable `dt_` for the time step in the lifting surfaces class and standard variable `t_start`, corresponding to start time of propellers, for the propeller solver. No other standard name for variables or functions is assumed, as the specific functions or variables are accessed under the condition of the type of solver chosen.

3.3.4.2 Propeller and Lifting Surface Interface Workflow

Figure 3.14 illustrates the propeller and lifting surface interface workflow. This loop is executed until the specified number of time steps is achieved. For the aeroelastic coupling, each call of this interface will perform one step, and the current time is provided as an argument.

Inside the loop, the following sequence is executed:

- Update blades control points (`UpdateBladesControlPoints`): attributes control points stored in public variable of propeller solver to the internal variable of the interface; this step is done to allow for flexible choice of propeller approach and flexible nomenclature;

- Calculate induced velocities of surfaces' wakes on propeller solver control points (`InducedVelocitiesWakeSufacesOnProp`): call to calculate wake surfaces influence is executed, according to the solver selected. For Vortex Lattice and LL/VVPM coupling, `WakeInducedVelocityOnSpecifiedPoints` function is called to calculate the influence of wake surfaces on the blades' control points and particles positions;

- Call selected solver for lifting surfaces aerodynamics (`AeroSolver`): according to the approach selected, attributes propeller induced velocities at the lifting surface solver control points (by modifying the corresponding public variables inside the lifting surface class), and call the respective solution;

- Calculate velocities induced by surfaces (`InducedVelocitiesSufacesOnProp`): according to solver selection, call the appropriate public function to calculate the influence of surfaces on propeller solver control points. For Vortex Lattice and LL/VVPM coupling, `InducedVelocitiesPanelsOnSpecifiedPoints` function

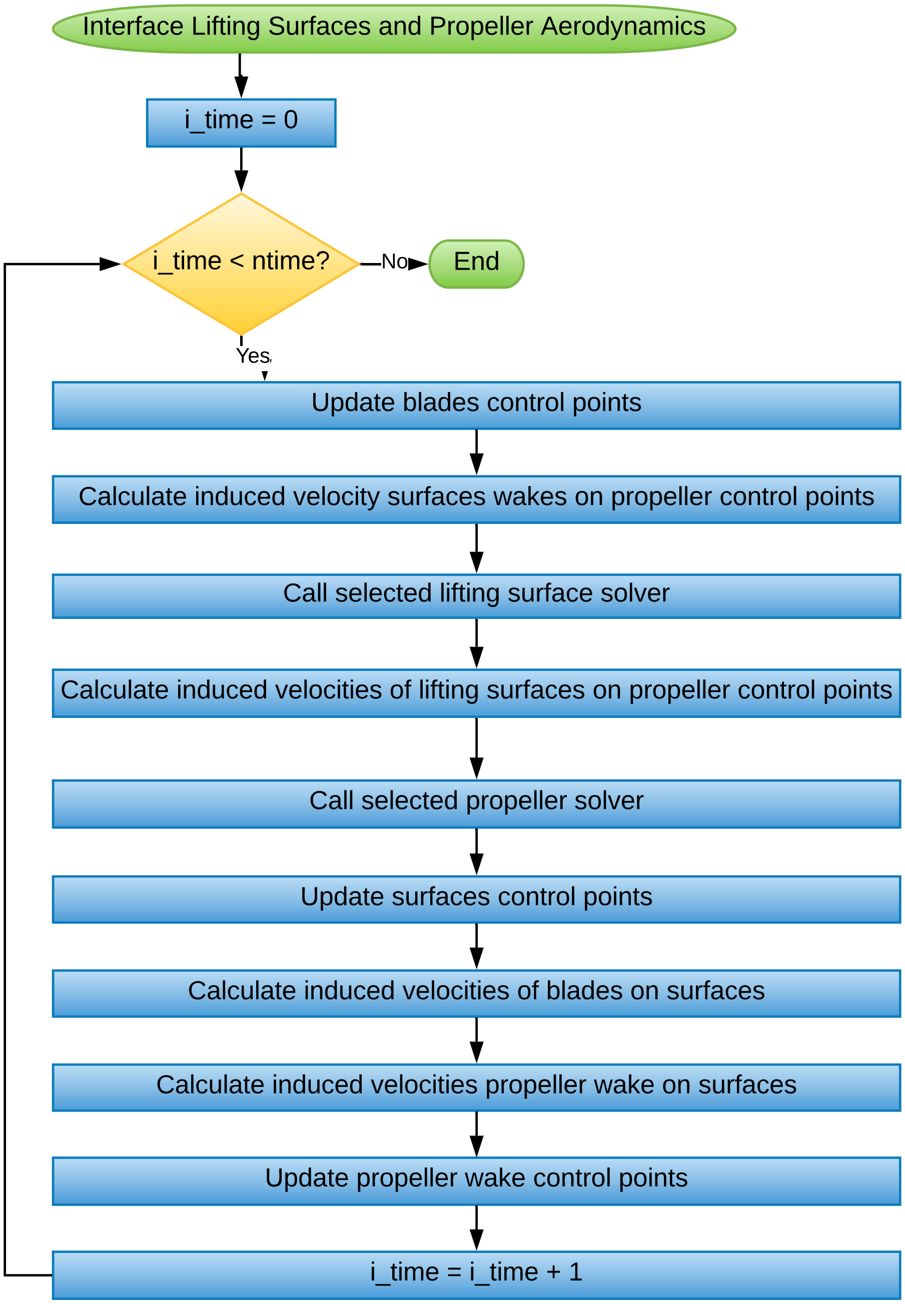

Interface Lifting Surfaces and Propeller Aerodynamics
i_time = 0
i_time < ntime?
No
End
Yes
Update blades control points
Calculate induced velocity surfaces wakes on propeller control points
Call selected lifting surface solver
Calculate induced velocities of lifting surfaces on propeller control points
Call selected propeller solver
Update surfaces control points
Calculate induced velocities of blades on surfaces
Calculate induced velocities propeller wake on surfaces
Update propeller wake control points
i_time = i_time + 1

Figure 3.14: Flowchart illustrating interface between lifting surfaces and propeller aerodynamics

is called to calculate the influence of surfaces on the blades' control points and particles positions;

- Call selected propeller solver (`PropSolver`): attributes induced velocities of lifting surfaces on propellers, by modifying the respective public variables and call propeller solver;

- Update surfaces control points (`UpdateSurfacesControlPoints`): attributes control points stored in the public variable of lifting surfaces solver to the internal variable of the interface; this step is done to allow for flexible choice of lifting surfaces aerodynamic approach and flexible nomenclature;

- Calculate blades induced velocities (`InducedVelocitiesBladesOnSurfaces`): Call respective propeller function to calculate induced velocities of blades on surfaces and its wake. For Vortex Lattice and LL/VVPM coupling, call for the function `InducedVelocitiesParticlesonEvaluationPoints` from propeller solver is executed to calculate the blades induced velocities at panels control points.

- Induced velocities propeller wake (`InducedVelocitiesPropWakeOnSurfaces`): Call propeller function that calculates wake induced velocities at specified points. For the LL/VVPM, function `InducedVelocitiesParticlesonEvaluationPoints` is called to calculate propeller slipstream influence at panels control surfaces;

- Update propeller wake control points (`UpdatePropWakeControlPoints`): attributes wake control points stored in the public variable of propeller solver to the internal variable of the interface;

3.3.5 Interface Class UM/NAST and Vortex Lattice

Besides interfacing the UM/NAST 1D structural model and the Vortex Lattice 2D non-planar grid, allowing the transferring of loads and displacements between the two solvers, the interface class for UM/NAST and Vortex Lattice allows the change of information between data structure specific of each solver. It works as a “glue,” linking and interfacing the new C++ developments with the UM/NAST framework. More details are provided in the next sections.

3.3.5.1 Architecture Overview

An overview of the interface class between UM/NAST and Vortex Lattice is provided in Figure 3.15. As with the interface between lifting surfaces and propellers, this class does not have a direct input file, but the respective input files for the solvers involved need to be defined.

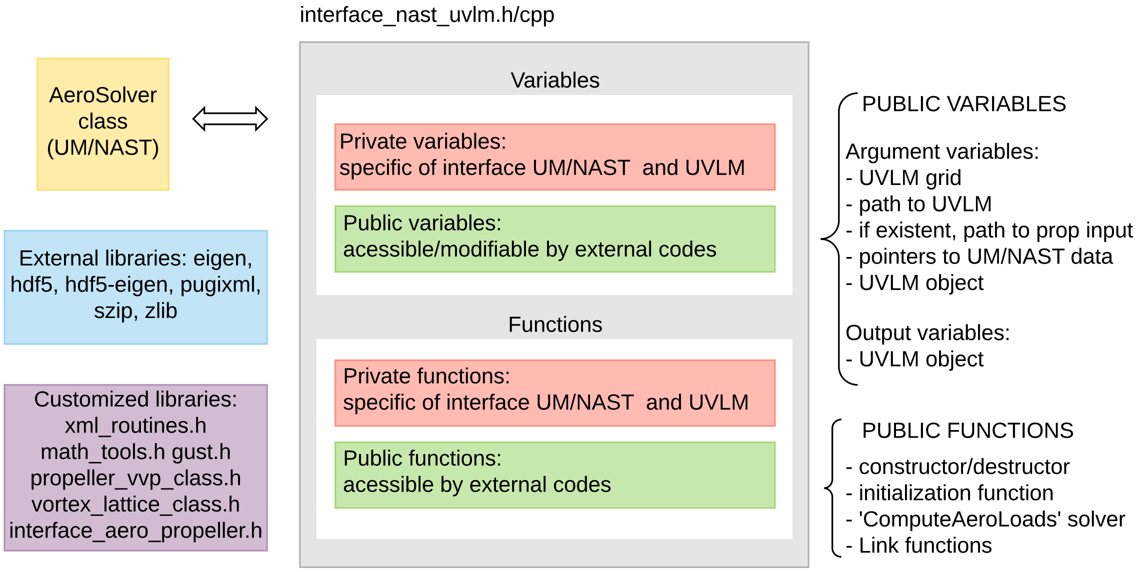

interface_nast_uvlm.h/cpp
AeroSolver class (UM/NAST)
External libraries: eigen, hdf5, hdf5-eigen, pugixml, szip, zlib
Customized libraries: xml_routines.h math_tools.h gust.h propeller_vvp_class.h vortex_lattice_class.h interface_aero_propeller.h
Variables
Private variables: specific of interface UM/NAST and UVLM
Public variables: acessible/modifiable by external codes
Functions
Private functions: specific of interface UM/NAST and UVLM
Public functions: acessible by external codes
PUBLIC VARIABLES
Argument variables:
- UVLM grid
- path to UVLM
- if existent, path to prop input
- pointers to UM/NAST data
- UVLM object
Output variables:
- UVLM object
PUBLIC FUNCTIONS
- constructor/destructor
- initialization function
- 'ComputeAeroLoads' solver
- Link functions

Figure 3.15: Interface class UM/NAST and Vortex Lattice architecture overview

As commented at the beginning of this chapter, in order to link this interface with UM/NAST 4.0 data and workflow, this class inherit from the UM/NAST class `AeroSolver`, defined inside UM/NAST 4.0, in order to allow the coupling between this new UM/NAST version with external aerodynamic solvers. Also, standard public functions defined as virtual functions in the AeroSolver class need to be defined and implemented in the external aerodynamic solver class. Currently, these virtual functions are i) `Initialize`, for some necessary initialization of the external solver; ii) `ComputeAeroLoads`, where the interface and aerodynamic solver call happens, and iii) the link functions `LinkModel` and `LinkStructuralSolver`, which links UM/NAST classes containing data about the model and quantities related to the structural/flight dynamic solver. Also, as a last observation about the coupling, both UM/NAST 4.0

and the external aerodynamic solver (in this case, named UVLM/Propeller module) are libraries that need to be installed and called by an external driver, where the aeroelastic solution finally takes place, as illustrated in Figure 3.1).

As with the other classes presented previously, public argument and output variables are defined and can be accessed by an external solver (driver). Finally, the external and customized libraries used for this class are summarized in Figure 3.1).

3.3.5.2 Initialization Functions

Figure 3.16 illustrate the initialization function of the interface between UM/-NAST and Vortex Lattice/Propeller.

- First, the UVLM input parameters, whose path is in a public variable provided by the driver, is read and processed.

- Then, if propellers are defined, the corresponding input file for the propeller solver (currently LL/VVPM is the only option) is read and an object of the class that interfaces lifting surfaces and propellers is created;

- Propeller grid, as well as root points matrices, are initialized with zeros. The "root points" the aerodynamic Vortex Lattice grid intersects with structural beam model representation (determined by Vortex Lattice spanwise discretization). They are important for the functions that transfer loads/displacements;

- Finally, airfoils coordinates are processed and the associated camber line determined.

3.3.5.3 UM/NAST and Vortex Lattice Interface Workflow

A brief overview of the `Compute aerodynamic loads` function is illustrated in Figure 3.17:

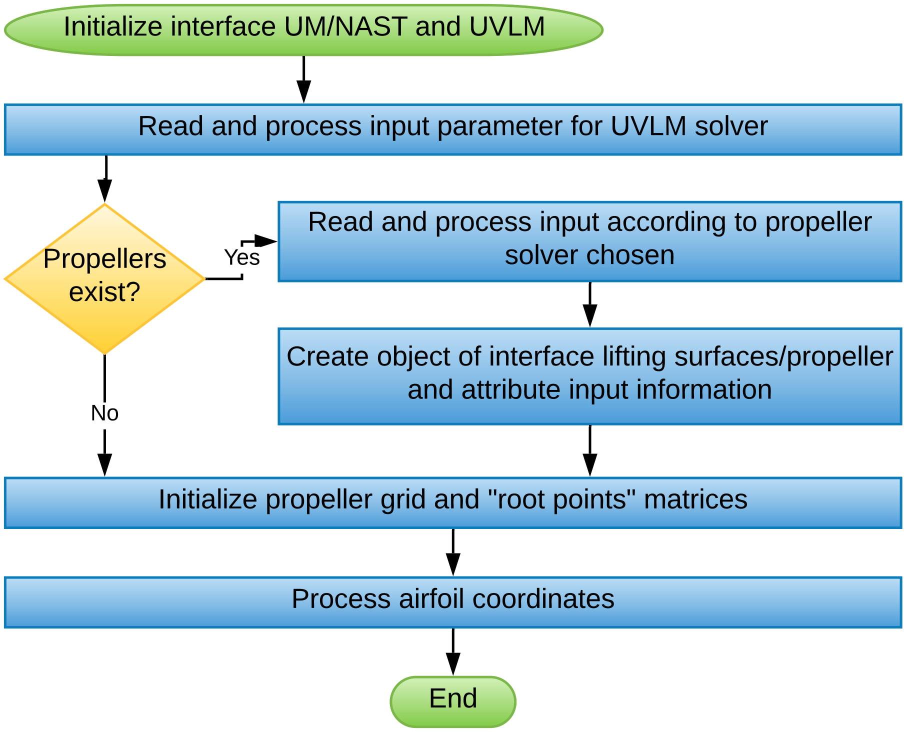

Initialize interface UM/NAST and UVLM
Read and process input parameter for UVLM solver
Propellers exist?
Yes
Read and process input according to propeller solver chosen
Create object of interface lifting surfaces/propeller and attribute input information
No
Initialize propeller grid and "root points" matrices
Process airfoil coordinates
End

Figure 3.16: Flowchart illustrating initialization interface between UM/NAST and Vortex Lattice solver

- First, based on the current model geometric configuration provided by UM/NAST, the Vortex Lattice grid is generated (function `Beam2Wing`): first calculates coordinates of root points by using UM/NAST formulation (summarized in Chapter 2); based on the root points positions (in body frame), corresponding local frame orientations and camber line information, and assuming rigid behavior in chordwise direction, the Vortex Lattice grid is detemined;

- Call appropriate Vortex Lattice solver (`CallVortexLattice`): steady without propeller, unsteady without propeller, steady with propeller or unsteady with propeller; for each case the necessary arguments are obtained and the corresponding solver is called;

- Once calculated, the loads over the panels are transferred to the structural nodes (`Wing2Beam`): first the loads are transferred from panels to root points,

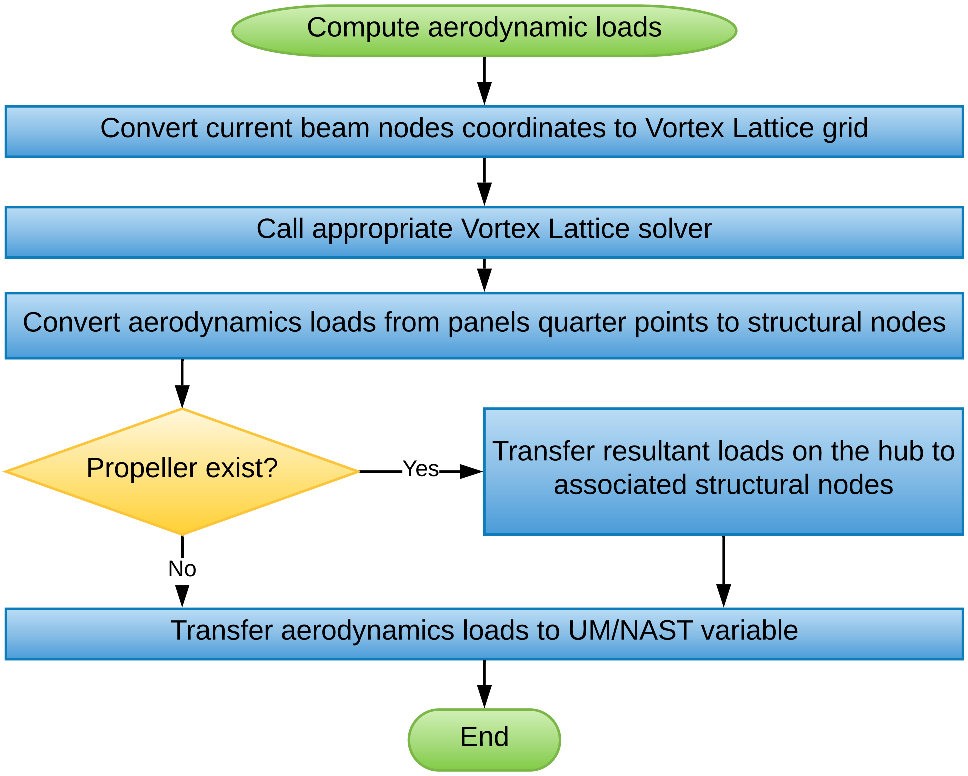

Compute aerodynamic loads
Convert current beam nodes coordinates to Vortex Lattice grid
Call appropriate Vortex Lattice solver
Convert aerodynamics loads from panels quarter points to structural nodes
Propeller exist?
Yes
Transfer resultant loads on the hub to associated structural nodes
No
Transfer aerodynamics loads to UM/NAST variable
End

Figure 3.17: Flowchart illustrating initialization interface between UM/NAST and Vortex Lattice solver

considering geometric relations between the quarter points (where loads are apllied) and the root points (assuming rigid chordwise direction) and making use of conservation of virtual work; then, the loads are transferred from to root node to neighboring structural nodes by using a linear interpolation (in accordance with the fact that strains are assumed constant for each UM/NAST element);

- If propellers are present, transfer resultant loads from propeller hub to associated structural nodes;

- Finally, transfer aerodynamic loads to UM/NAST variable.

CHAPTER 4

Stability Analysis of VFA Including Propellers

Traditionally, two main approaches are used to extract dynamic information (frequencies, damping, and modes) and perform a stability analysis of an aeroelastic system: frequency and time-domain methods. The frequency-domain methods are primarily developed for linear aeroelastic systems, as they assume the superposition of signals with different constant frequencies. Therefore, they are more suitable for problems with small deformations. Typical methods for stability analysis based on frequency domain are the k method [155], p-k method [156] and p method [157].

For very flexible aircraft undergoing large deformations, however, strong nonlinearities can arise, and time-domain methods are usually preferred [158]. Examples of time-domain methods for stability analysis are reviewed in McNamara and Friedmann [159]: moving-block approach (MBA) [160], least squares curve-fitting method (LSCFM) [161] and system identification using the autoregressive moving-average (ARMA) model [162].

Even for linear systems, the complexity of the problem can make the use of frequency-domain techniques a cumbersome task, as the equations are not easily manipulated. In this case, time-domain simulations may also be preferred. In Silva [163], for example, the dynamic information of a linear transonic aeroelastic problem based on modal solutions is obtained by the system identification of CFD simulations.

In UM/NAST, the flutter solution is obtained based on the linearization of the governing equations of the system. The model is first linearized about its trimmed condition, and then the dynamic information is extract based on the eigendecomposition of the so-called state matrix A. This was developed by the original UM/NAST aerodynamic formulation, which used Peters' inflow theory.

Incorporating the UVLM aerodynamics for the lifting surfaces and LL/VVPM formulation for the propellers in the UM/NAST linearized formulation would require an in-depth review of the linearization formulation. Also, while the analytical expressions of Peters' inflow theory allow a direct coupling with the structural equations, the same is not the case for the UVLM and LL/VVPM approaches, which are solved separately from the structural solver at each time step. This option was, then, out of the scope for the present work. Instead, an alternative approach for the stability analysis of VFA was proposed and applied [164]. The approach makes use of system identification combined with Proper Orthogonal Decomposition (POD) modes (Sys ID + POD) in order to reduce the number of degrees of freedom (outputs) of the problem. This is important, as the system identification is not well suited for a high number of outputs. Another advantage is that a modal structural approach is not required and the method can be used for deformed configurations far from the undeformed condition.

This chapter describes how the Sys ID + POD method was applied and presents verification results for an available purely structural reference.

4.1 Alternative Approach to Extract Dynamic Information

4.1.1 Proper Orthogonal Decomposition

Proper Orthogonal Decomposition is a powerful method, based on statistical approaches, used to extract a compact basis representation of a large set of data, typically a matrix containing several snapshots of the dynamic system states. The two primary goals for which it is usually applied are the order-reduction of a dynamic system with large number of states (or, alternatively, a compact representation for an output-based approach) and the extraction of insightful information about the physical characteristics associated with the data set [165], as the extraction of dominant features from experimental data [166]. It has been demonstrated to be a preferred basis for many applications, becoming widely applied in several different fields of engineering such as imaging processing, signal analysis, chemical engineering, oceanography, civil engineering, structural dynamics, and aerodynamics [166].

One fundamental characteristic of the POD method is that it offers an optimal choice of the compact basis, minimizing the average squared distance between the original signal and its reduced linear representation [165], often making use a surprisingly few numbers of modes [166]. It is important to note, however, that for the resulting basis to be a good representation of the real system, the signal provided needs to be generated in a manner to excite the main dynamic features of the system main dynamic features. Another important characteristic is that, although the POD method provides a linear representation of the signal, it makes no assumption about the linearity of the problem, becoming an attractive approach for non-linear systems.

The POD method was developed independently by different people. Lumley [167] attributes its origin to the independent investigations of Kosambi (1943), Loéve (1945), Karhunen (1946), Pougachev (1953) and Obukhov (1954). Although POD has been widely employed in different areas, its nomenclature has different interpretations, as discussed in [166]. In some contexts, it is referred to as the Karhunen Loéve decomposition (KLD) [168]. However, it can also be referred to as the principal component analysis (PCA) [169] or the singular value decomposition (SVD) [170]. Liang *et al.* [166] suggests that the POD nomenclature should be interpreted a set of approaches consisting of the three methods cited above. Although such approaches have different formulations, they share the common goal of extracting a compact and optimal representation of a data set, and were demonstrated by Liang *et al.* [166] to produce equivalent results when applied to find a general basis representation of discrete random vectors.

In this work, the POD based on singular value decomposition was used. The SVD can be seen as an extension of eigenvalue decomposition for non-square matrices. Considering, for example, a general rectangular complex matrix $M_{m\times n}$ with m rows and n columns, it can be factorized based on the singular value decomposition as:

$$M_{m\times n} = U_{m\times m}\ \Sigma_{m\times n}\ V^*_{n\times n} \tag{4.1}$$

where:

- $U_{m\times m}$ is a real or complex unitary square matrix whose columns are called left singular vectors of $M_{m\times n}$ and correspond to the orthonormal eigenvectors of $M_{m\times n}M^*_{m\times n}$;

- $\Sigma_{m\times n}$ is a rectangular diagonal matrix with non-negative real entries, σ_i, in its diagonal. Such values are called singular values and its a common practice to order them in descending order. They correspond to the square roots of both $M_{m\times n}M^*_{m\times n}$ and $M^*_{m\times n}M_{m\times n}$;

- $V_{n\times n}$ is a real or complex unitary square matrix whose columns are called right singular vectors of $M_{m\times n}$ and correspond to the orthonormal eigenvectors of

$M^*_{m\times n}M_{m\times n}$;

and matrix $M^*_{m\times n}$ is the conjugate transpose of matrix $M_{m\times n}$.

The left singular vectors provide a modal basis of the original matrix $M_{m\times n}$. On the other hand, the singular values (diagonal values of $\Sigma_{m\times n}$) are associated with the importance of the POD mode to represent the data given by matrix M. The higher the value of a singular value σ_i associated to the $i-th$ left-eigenvalue, the more important this mode is to represent the system behavior. Based on this fact, a metric called relative information content (RIC), is typically used to characterize the importance of a left-eigenvalue (system mode). For the i-th singular value, the associated RIC is given by:

$$RIC_i = \frac{\sigma_i}{\sum_{i=1}^{m}\sigma_i} \tag{4.2}$$

Then, considering that the singular values are numerated in descending order, the number p of POD modes can be determined by choosing the first p modes that provide a cumulative RIC higher than a pre-defined fraction, f, chosen depending on the problem.

$$\sum_{i=1}^{p} RIC_i = \frac{\sum_{i=1}^{p}\sigma_i}{\sum_{i=1}^{m}\sigma_i} \geq f \tag{4.3}$$

Fraction f gives a percentage of how good the selected POD modes will represent the system, and its choice depends on the accuracy and system reduction desired and the number of degrees of freedom (d.o.f.) of the problem. Usually, the first few POD modes already provide a cumulative RIC higher than $f = 99\%$.

4.1.2 System Identification

System identification is a discipline dedicated to building (identifying) a mathematical model of a system based on a provided set of inputs and outputs, which can be in either the time or frequency domain. It involves: i) the data collection (*e.g.*, from experiments or simulations); ii) selection of a parameterized mathematical model, based on the identification goals (*e.g.*, transfer functions, state-space equations, non-linear functions), iii) selection and application of an estimation method (to evaluate the adjustable parameters); and iv) evaluation of the estimation quality, based on the application needs [171].

The term "system identification" was coined by Zadeh [172] in 1956 as part of control theory [173]. Several approaches were developed since then, and a detailed review on historical aspects and a discussion about different methods can be found in Gevers [174] and in Ljung [173].

Given a set of measurements of a real system, represented by the operator $\tilde{G}$, the goal of the system identification is to obtain a mathematical operator G that approximates the real operator, such that the following condition holds for all input/output combinations [175]:

$$||d - \tilde{d}|| < tol, \;\; d = G(x, m) \tag{4.4}$$

where $||.||$ is some chosen norm definition, d are the outputs obtained from the identified model G, $\tilde{d}$ are the real system measured outputs, x are the input variables and m the model parameters.

In the present work, the subspace state-space system identification (N4SID), an input-output based approach which is available in the MATLAB System Identification Toolbox [171], was applied. Based on the inputs/outputs provided, this algorithm finds a state sequence which is associated to outputs of unsteady Kalman filter banks, and a mathematical representation of the system in state-space form can be determined:

$$\dot{x}(t) = Ax(t) + Bu(t) + Ke(t) \tag{4.5}$$

$$y(t) = Cx(t) + Du(t) + e(t) \tag{4.6}$$

where A, B, C, and D are the estimated matrices of the state-space model, K a matrix which accounts for disturbances, $u(t)$ is the input, $y(t)$ is the output and $x(t)$ is a vector with n_x states associated with this mathematical representation. More details about the N4SID method can be found in Van Overschee and De Moor [176].

4.1.3 Combining POD and Sys ID for Stability Analysis

In order to extract dynamic information (frequencies, damping, and modes) from the coupled aeroelastic-flight dynamics framework with propellers, it is first necessary to find a linearized mathematical representation of the system containing all the coupled structural, flight dynamics, lifting surfaces aerodynamics and propeller effects (inertial and aerodynamics).

One natural choice for this is to apply system identification and find a state-space representation of the global system. However, one difficulty in directly applying system identification in this context is that the VFA model often has a high number of degrees of freedom (the model considered later, for example, has more than 500 degrees of freedom), resulting in a high number of output states. However, system identification is not well suited for a high number of outputs, and the prediction performance is deteriorated.

One option to tackle this problem would be to write the output data as a function of a linear combination of the aircraft linear modes. This was done, for example, by Silva [163] and studies for aeroelastic problems involving small deformations. However, for a VFA, a representation of the deformed equilibrium state with linear modes is no longer adequate.

Another alternative, used in this work, is to reduce the output matrix dimension by using POD modes. Then, instead of directly providing the output matrix containing the snapshots of displacements, the snapshots containing the coefficients of a much smaller number of POD modes are provided, and the original degrees of freedom are later recovered. This has the additional advantage of providing a better basis representation for an aircraft whose equilibrium state exhibits large deformation compared to the undeformed shape. Figure 4.1 illustrate how this combination of POD + Sys ID works.

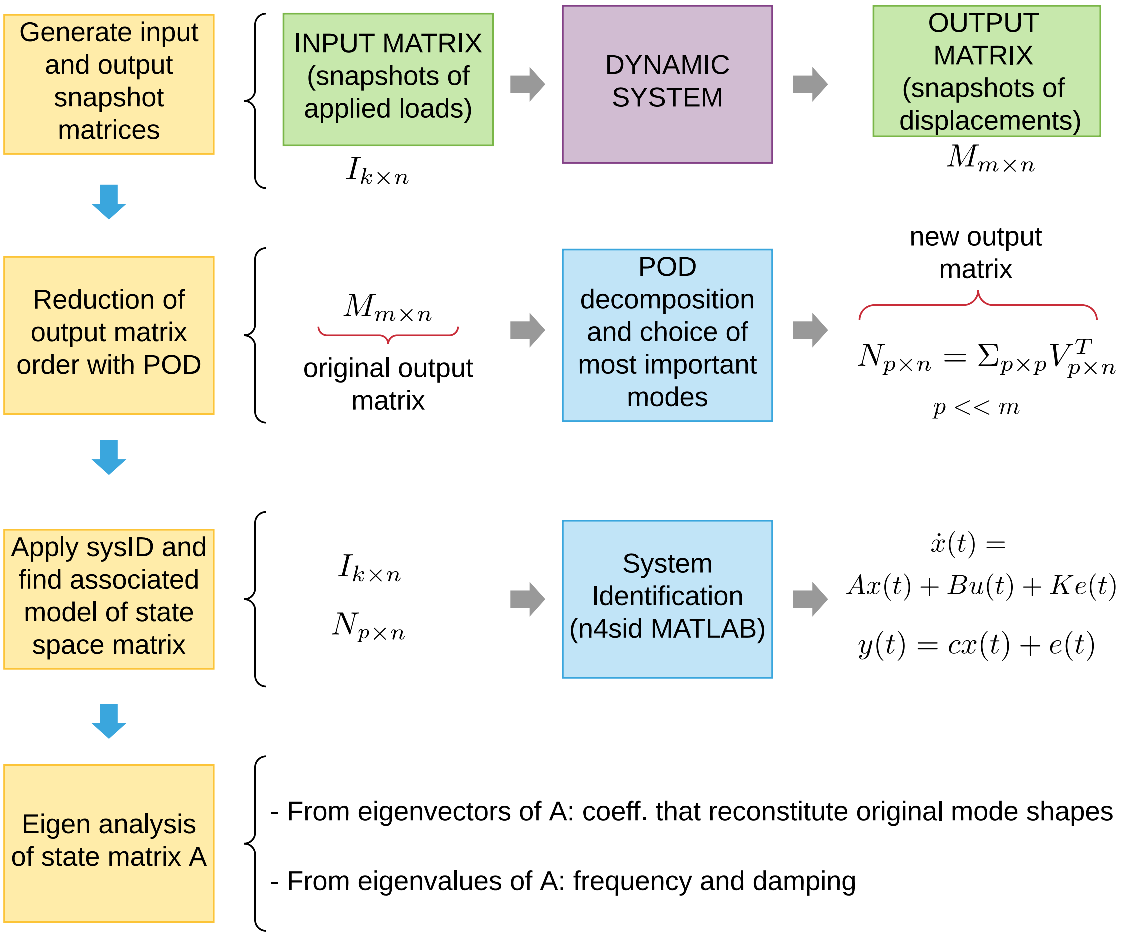

Generate input and output snapshot matrices
INPUT MATRIX (snapshots of applied loads)
$I_{k\times n}$
DYNAMIC SYSTEM
OUTPUT MATRIX (snapshots of displacements)
$M_{m\times n}$
Reduction of output matrix order with POD
$\underbrace{M_{m\times n}}$
original output matrix
POD decomposition and choice of most important modes
new output matrix
$N_{p\times n} = \Sigma_{p\times p} V^T_{p\times n}$
$p << m$
Apply sysID and find associated model of state space matrix
$I_{k\times n}$
$N_{p\times n}$
System Identification (n4sid MATLAB)
$\dot{x}(t) = Ax(t) + Bu(t) + Ke(t)$
$y(t) = cx(t) + e(t)$
Eigen analysis of state matrix A
- From eigenvectors of A: coeff. that reconstitute original mode shapes
- From eigenvalues of A: frequency and damping

Figure 4.1: POD + Sys ID workflow

First, the original data snapshots are collected by performing a time simulation disturbing the aircraft from its equilibrium condition. It is worthwhile to note that, for the extraction of dynamic information, it suffices to apply small disturbance around the deformed equilibrium condition, making possible the use of linear system identification approaches (as is the case of the selected method, n4sid). Then, the input matrix consists of the snapshots of the values of loads (disturbances) is applied. The original output matrix consists of the snapshots of the displacements (in x, y and z directions) of each structural nodes with relation to an initial equilibrium condition.

Next, the POD modes of the original output need to be determined and selected. For this, SVD is first applied to the original output matrix M:

$$M_{m \times n} = U_{m \times m} \; \Sigma_{m \times n} \; V^T_{n \times n} \tag{4.7}$$

where m is the number of d.o.f., n is the number of snapshots, U and V are orthogonal matrices containing left-singular vectors and right-singular vectors of M, with the columns of U corresponding to the POD modes, and Σ is a diagonal matrix of non-negative real number ordered in descending order. Also, the product $\Sigma_{m \times n} \; V^T_{n \times n}$ is associated with the coefficients of the POD modes which reconstitute the matrix M.

Usually, the first few POD modes are responsible for more than 99% of the trace of Matrix Σ. Reducing the number of POD modes to p, one obtains the new output matrix containing snapshots of the coefficients of the p POD modes which approximate the matrix M where $p << m$. For this, a reduced singular value decomposition representation of M can be found by using just the p POD modes selected as:

$$M_{m \times n} = U_{r,m \times p} \; \Sigma_{r,p \times p} \; V^T_{r,p \times n} \tag{4.8}$$

The new reduced output matrix N to be provided to the system identification is then given by:

$$N_{p \times n} = \Sigma_{p \times p} \; V^T_{p \times n} \tag{4.9}$$

Once the reduced output matrix is determined, Sys ID is applied and the matrices in the equations 4.5 and 4.6 are determined. In this work, the tool `n4sid` available on Matlab [171, 176] was used.

Finally, the dynamic information of the system can be found by the eigenvalue decomposition of matrix A, in Eq. 4.5. As the state-space representation is in a continuous time domain, the frequencies and damping can be directly obtained from the eigenvalues λ_i of A, as:

$$freq = \frac{Re(\lambda_i)}{2\pi} \tag{4.10}$$

$$damp = \frac{Imag(\lambda_i)}{2\pi} \tag{4.11}$$

where $Re(\lambda_i)$ and $Imag(\lambda_i)$ are the real and imaginary parts of λ_i, respectively.

From the eigenvectors of A, and remembering that they are related to coefficients of the chosen POD modes, the mode shapes can be obtained. Considering E as the matrix containing the eigenvectors of A and by equation 4.6, the modes in terms of displacements at structural nodes can be recovered by:

$$N_{p\times n} = U_{r,m\times p} C_{p\times k} E_{k\times k} \tag{4.12}$$

where k is the order chosen for the system identification method.

Due to nonlinearities, non-proportional damping, and possible noise, the modes obtained by the system identification can be complex. In Rainieri and Fabbrocino [177] a discussion of those complex modes from system identification is presented, and an approach, used in the present work, to convert them to real mode shapes was proposed. The mode shapes can be obtained by adding the real modes of displacements with the initial equilibrium condition.

Finally, it was observed that, for better accuracy, the number of frequencies determined should be no more than the number of POD modes selected.

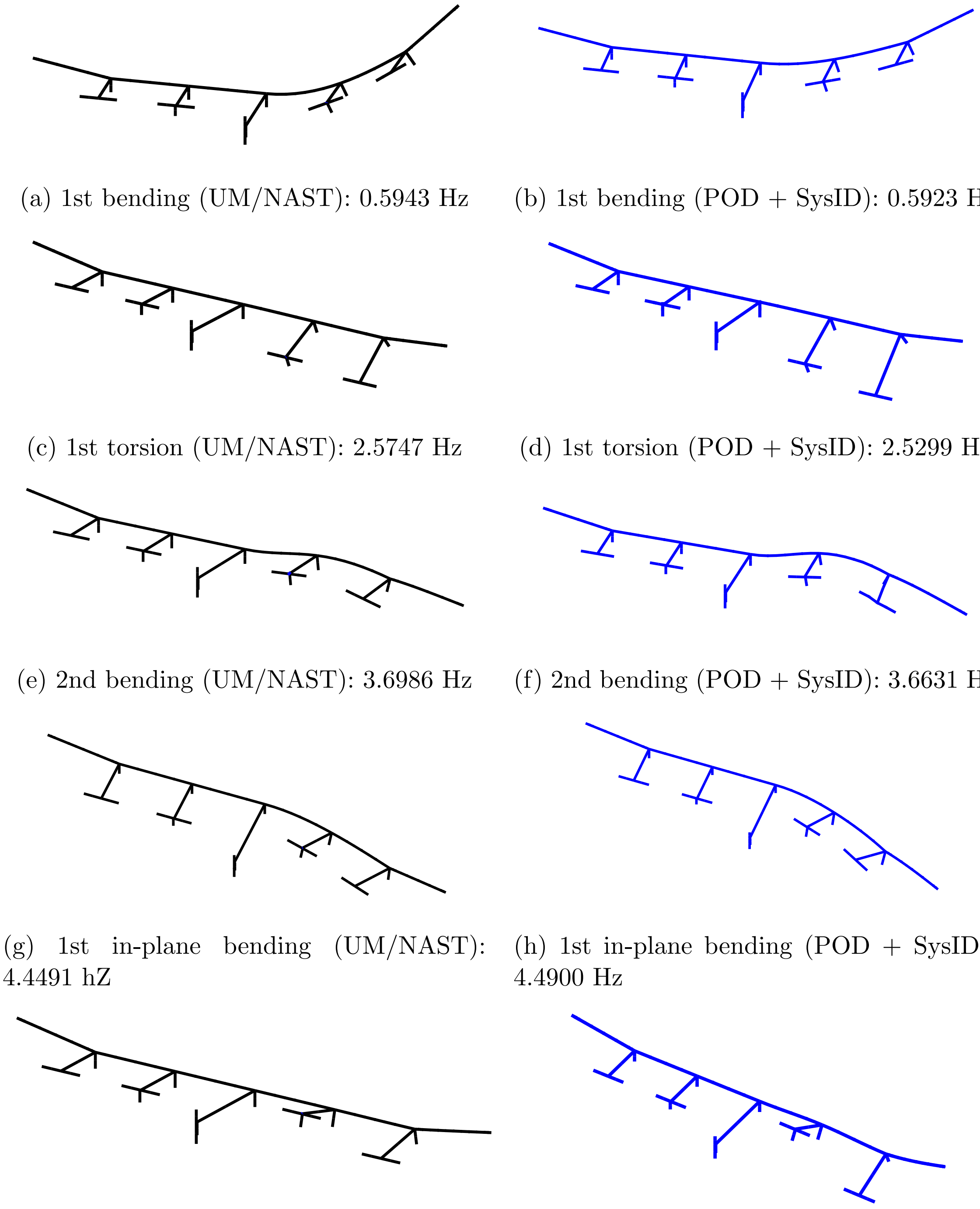
4.2 Verification of the Method

In order to verify the capability of the POD + Sys ID method to extract frequencies, damping, and modes based on the snapshots generated by the simulation of a VFA, a purely structural case was used, for which a reference from the UM/NAST modal solver was available about its undeformed shape. The model used was the University of Michigan's X-HALE UAS [178, 179] and more details about this model are provided in Chapter 5 and Appendix B.

For the POD + Sys ID, a time simulation was performed for the clamped X-HALE model. From the undeformed configuration, the model was disturbed by a 5 Nm torsion moment and a 1 N step force in the vertical and chordwise direction, all applied at the right wing tip of the clamped model and chosen to excite different modes. The snapshots were then provided to the POD + Sys ID method, 5 POD modes were used (contributing for more than 99% of the snapshot energy), and an order of 18 was chosen for the system identification method based on the fitting quality of the POD coefficients. As the reference was for the undeformed, undamped case, no gravity or damping effects were included. Also, as the identification order of 18 provides 9 frequencies, but just 5 POD modes are used (and up to 5 frequencies can be more accurately identified), the main frequencies were determined by choosing the five higher norms of the corresponding coefficients identified for the POD modes. Therefore the frequencies associated with the five columns of $C_{p\times k}E_{k\times k}$ with higher Euclidean norms were kept and shown in Figure 2.

Table 4.1: Comparison of natural frequencies for purely structural case

UM/NAST	POD + SysID	Error (%)	MAC
0.5943 Hz	0.5923 Hz	-0.3422	0.9879
2.5747 Hz	2.5299 Hz	-1.7407	0.9896
3.6986 Hz	3.6631 Hz	-0.9596	0.9922
4.4491 Hz	4.4900 Hz	0.9195	0.9937
6.5696 Hz	6.5286 Hz	-0.6241	0.9980



(a) 1st bending (UM/NAST): 0.5943 Hz

(b) 1st bending (POD + SysID): 0.5923 Hz

(c) 1st torsion (UM/NAST): 2.5747 Hz

(d) 1st torsion (POD + SysID): 2.5299 Hz

(e) 2nd bending (UM/NAST): 3.6986 Hz

(f) 2nd bending (POD + SysID): 3.6631 Hz

(g) 1st in-plane bending (UM/NAST): 4.4491 hZ

(h) 1st in-plane bending (POD + SysID): 4.4900 Hz

(i) 2nd torsion (UM/NAST): 6.5696 Hz

(j) 2nd torsion (POD + SysID): 6.5286 Hz

Figure 4.2: Comparison of mode shapes and natural frequencies for the X-HALE UAS vehicle about its undeformed configuration

Table 4.1 presents the comparison between frequencies as well as a comparison of the corresponding modes by applying Model Assurance Criteria (MAC). For this purely structural case, one can see the method had an excellent agreement with the data calculated by the modal solver in UM/NAST with a maximum error in frequency prediction smaller than 2% and a MAC value of about 0.99 for all five modes.

CHAPTER 5

Coupled Nonlinear Aeroelastic-Flight Dynamics Framework Verification

In this chapter, verification cases are presented for each new piece added to the enhanced coupled nonlinear aeroelastic-flight dynamics framework including propellers. First, in Section 5.1, comparisons are provided for static, trim and dynamic aeroelastic simulations using the integrated UM/NAST and Vortex Lattice against other codes. In Section 5.2, the propeller aerodynamics is checked by comparing simulations using the implemented LL/VVPM code with numerical and experimental results. The integration between the propeller and lifting surfaces is also evaluated by comparing against numerical data in Section 5.3. Finally, in Section 5.4, the gyroscopic loads calculation is verified against an analytical example.

5.1 Integration of UM/NAST and Vortex Lattice

5.1.1 16-m Wing Comparisons

5.1.1.1 16-m Wing Model

The 16-m wing model (Figure 5.1) consists of a rectangular flat wing with 16-m semi-span and 1-m chord and is a representative model of a slender generic very flexible wing [72]. Its corresponding structural model consists of a clamped 16-m

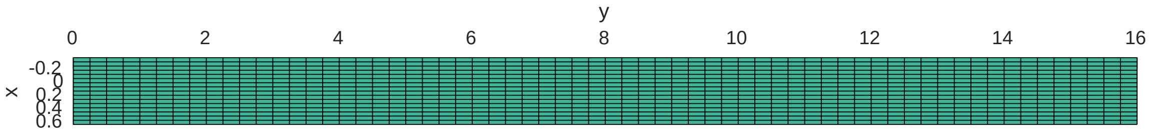


Figure 5.1: Vortex Lattice representation of the 16-m wing model

beam at 35% chord from the LE, with area moments of inertia varying quadratically along its length. Also, discrete masses with prescribed mass moments of inertia are attached along the structural model. Such selection of properties was made to provide bending deformations with constant curvature for typical aerodynamic loadings [72]. More information about its properties is provided in Appendix A.

5.1.1.2 16-m Wing Static Comparisons

As a first verification case, static aeroelastic results for the clamped 16-m wing model at $v = 40$ m/s and different angles of attack, with gravity included, were compared against different codes. These cases were also presented in Ritter *et al.* [72] and in Riso *et al.* [62].

For the results presented here, a Vortex Lattice discretization of 16 panels chordwise and 64 panels spanwise was used in UM/NAST + VLM. The structural beam was discretized in 32 elements (each one with three nodes), resulting in 65 nodes. This means that, in this example, the aerodynamic spanwise discretization coincides with the structural discretization. Therefore, just the first step of the transfer of loads described in Chapter 2, from quarter-chord points to root points (positions along the beam determined by aerodynamic spanwise discretization), is performed. The examples presented in the next sections will also excite the second step, which is the linear transfer along the beam from the root points to the neighboring structural nodes.

Figure 5.2 presents the comparison of the results with two external codes: DLR toolbox with VLM [4,72] and a code combining MSC.Nastran SOL 400 and VLM [62]. The DLR toolbox is a FEM-based aeroelastic framework that enables the simulation

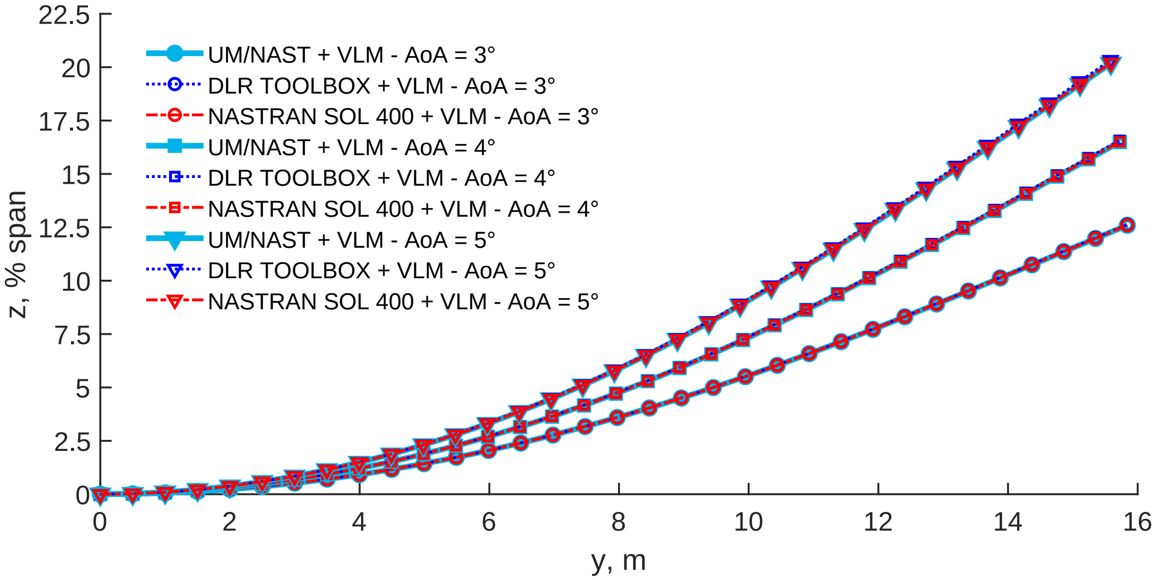


Figure 5.2: Comparison of aeroelastic static results for 16m wing

of complex configurations using three different options for the structural part: linear, reduced-order or fully non-linear formulations [4,72]. For the 16-m wing comparison, the DLR toolbox solution also used a combination of MSC.Nastran SOL 400 and VLM. All the three frameworks used the same vortex lattice code, developed by Ritter [81], but employed different approaches for the aeroelastic coupling: a 6 DOF splines for the MSC.Nastran SOL 400 plus VLM in [62], radial basis functions for the DLR toolbox and the approach described in Chapter 2 for UM/NAST (section-based transfer of loads from quarter points to beam and linear interpolation along the beam, if aerodynamic and structural spanwise discretization are different). Despite the differences, and as one can see in Figure 5.2, the results for the three codes show excellent agreement for all angles of attack considered.

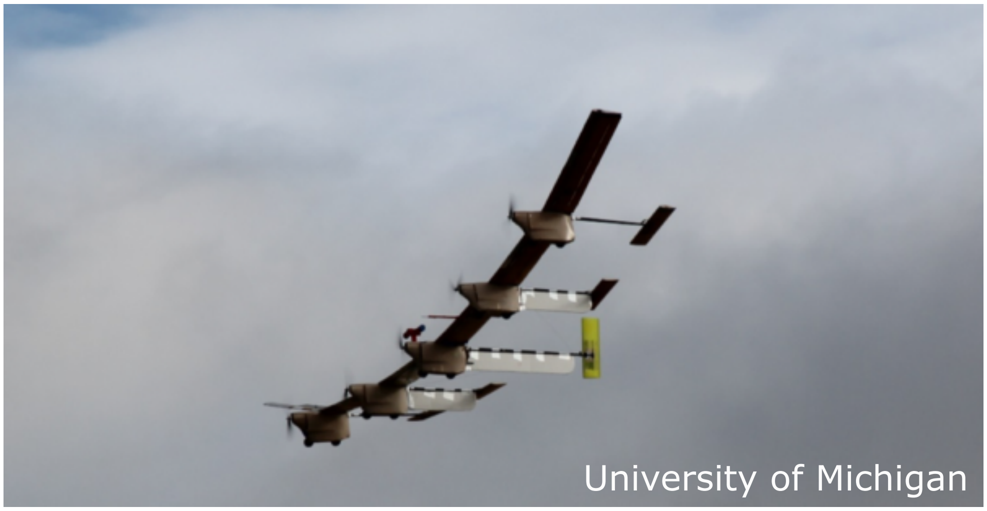


Figure 5.3: 6-m span X-HALE vehicle in flight

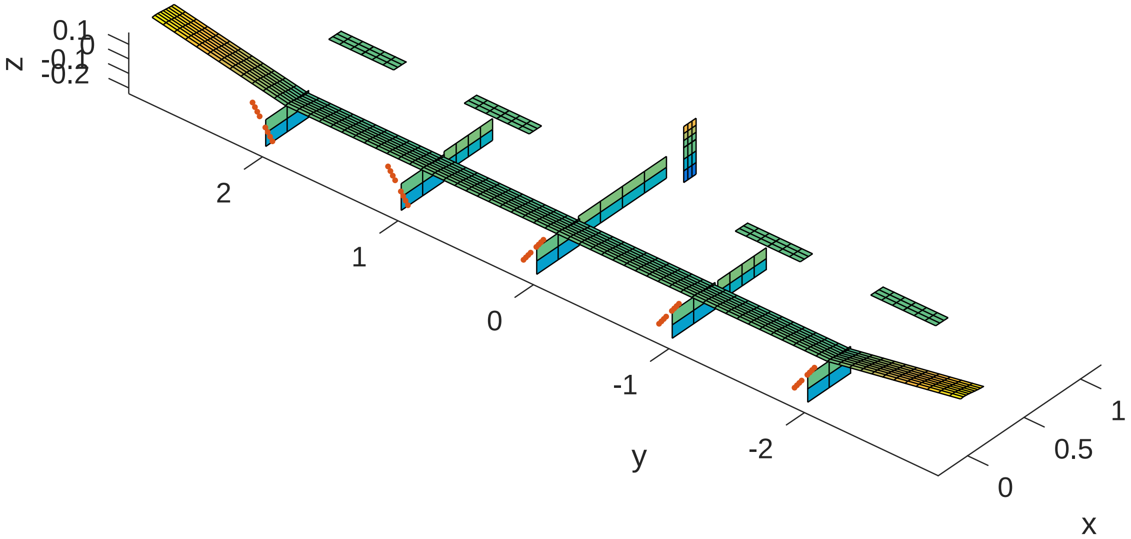


Figure 5.4: Undeformed panel model with propellers for the X-HALE UAS vehicle (units: meters)

5.1.2 X-HALE Comparisons

5.1.2.1 X-HALE Model

The University of Michigan's X-HALE UAS [178, 179] (Figure 5.3) is a representative very flexible aircraft conceived to collect experimental data for validation of nonlinear aeroelastic-coupled-flight-mechanics solvers. Three configurations with wing spans of 8, 6 and 4 meters have been designed. As shown in Figure 5.4 (a), the X-HALE 6-m configuration has 0.2-m chord, five pods along the wing, five tails (4 horizontal and 1 vertical stabilizers), three fins (vertical surfaces between the wing and tails) and five electric motor-propeller combinations located in front of each pod at spanwise locations $y = -2/3, -1/3, 0, 1/3$, and $2/3$ of semispan, where the origin corresponds to the aircraft centerline. The wing is mounted with an incidence angle of 5 degrees and has a 10-degree dihedral at its 1-m outer portion. More details about the properties of the X-HALE model used in this work are provided in Appendix B.

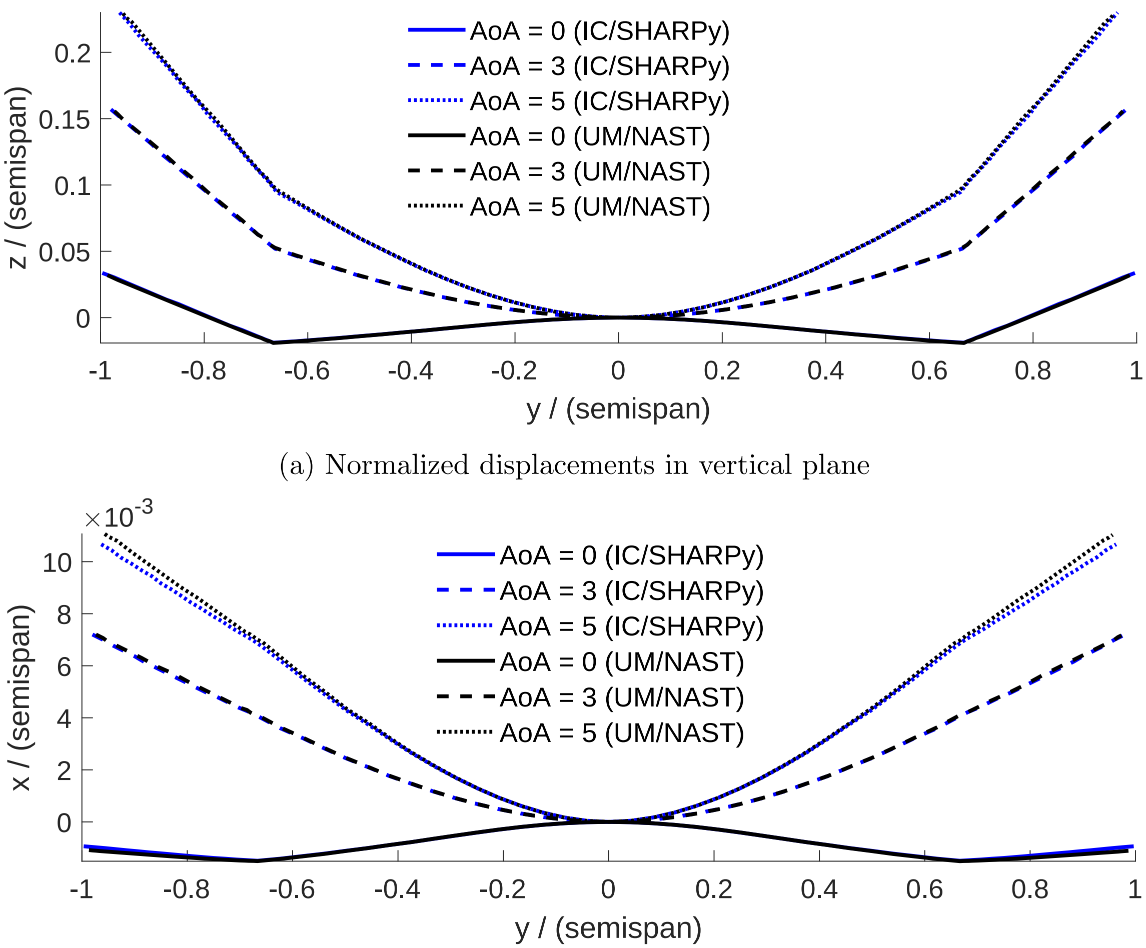
For the comparisons presented in this chapter, propeller modeling was not included.

5.1.2.2 X-HALE Static Results

Static simulations

Results for static aeroelastic X-HALE simulations are compared against an external aeroelastic framework for $v = 14$ m/s and angles of attack of 0, 3, and 5 degrees, with gravity included and the model clamped in the middle of the main wing. These same cases are also presented in Alfonso *et al.* [1]. Table 5.1 summarizes the structural and aerodynamic spanwise discretization and the type of members (elastic or rigid) used for the UM/NAST simulations in this section. In this case, the structural and spanwise aerodynamic discretizations in the dihedral region of the wing are different, exciting also the second step of the loads transfer described in Chapter 2 (linear transfer to the neighboring nodes along the beam, if structural and aerodynamic spanwise discretization are different).

The external solver in this case is SHARPy [63–66], developed at Imperial College London. SHARPy is also based on geometrically-nonlinear composite beam structures, but uses a displacement-based structural solver, while UM/NAST uses a strain-based approach. The aerodynamics is also Vortex Lattice, but the VLM code



(a) Normalized displacements in vertical plane

(b) Normalized displacements in horizontal plane

Figure 5.5: Comparison for aeroelastic cases with angle of attack 0, 3 and 5 at $v = 14$ m/s

was developed independently and coupled to the structural part in different ways. In SHARPy, the aerodynamic and structural models are not completely detached, and in order to simplify the mapping between them, the spanwise discretization of the VLM grid needs to match the spanwise discretization of the beam model [1]. Another difference is about the fluid-structure interaction (FSI) approach: UM/NAST + Vortex Lattice employs a single FSI interaction (weak FSI coupling) approach while SHARPy involves multiple FSI interactions (strong FSI coupling) for each time step (which makes the simulation considerably more expensive). As will be shown for the dynamic cases in the next section, however, this last difference does not play a role

Table 5.1: Aerodynamic and structural discretization used in UM/NAST

Part	Aerod. discretization		Structural Information	
	chord	span	number elem.	type
inner/mid wing segment	8	12	4	elastic
outer wing segment	8	12	8	elastic
tails (inner+outer parts)	8	6	1	rigid
fins	8	2	1	rigid
PODs	8	2	1	rigid
booms	-	-	1	rigid

Table 5.2: Trim results for cruise flight

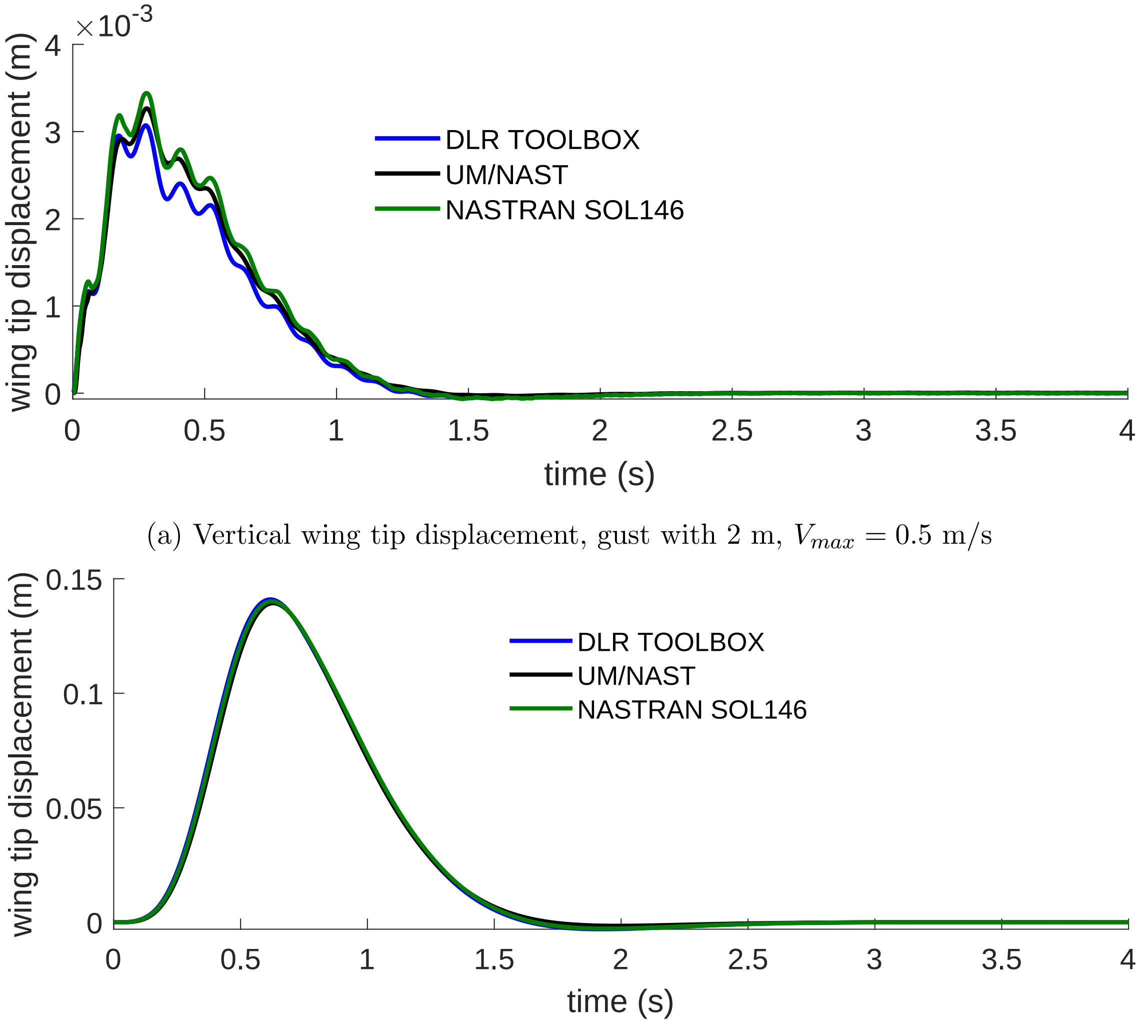
Code	Tail orientation	α [°]	δ [°]	T [N]
IC/SHARPy	Vertical	2.64	1.19	0.223
	Horizontal	2.21	0.52	0.213
UM/NAST	Vertical	2.59	1.15	0.179
	Horizontal	2.38	0.66	0.179

past a certain level of grid refinement.

Figure 5.5 presents the comparisons of X-HALE deformed equilibrium shape for vertical and horizontal planes (where x is along the downstream direction, y spanwise pointing to the right-wing and z vertical up). An excellent agreement can be observed for all the three angles of attack, with just slight differences in the horizontal plane for chordwise displacements (which are much smaller than the vertical ones).

Trim simulations

Verification for the trim parameters calculated for each solver was also performed for two X-HALE configurations: with the center tail vertical and with the center tail horizontal. The same discretization as in Table 5.1 was applied. Table 5.2 present the results for longitudinal trim. In general, the agreement is good, with some higher differences for the thrust, which is related to small differences in the induced drag prediction between the codes. In this case, no viscous drag was included.



(a) Vertical wing tip displacement, gust with 2 m, $V_{max} = 0.5$ m/s

(b) Vertical wing tip displacement, gust with 10 m, $V_{max} = 0.5$ m/s

Figure 5.6: Comparison for clamped X-HALE model under *1-cos* gust with different lengths

5.1.2.3 X-HALE Dynamic Results

Clamped simulations

In order to check the integration between UM/NAST and unsteady Vortex Lattice method (uVLM) for the transient response, simulations for a clamped X-HALE model were first performed. In this comparison, the X-HALE model was simplified: the 5-degree incidence of the wing was removed, and symmetric airfoil sections are employed for all surfaces. Table 5.3 summarizes the structural and aerodynamic

Table 5.3: Aerodynamic and structural discretization used in UM/NAST

Part	Aerod. discretization		Structural Information	
	chord	span	number elem.	type
inner/mid wing segment	12	12	4	elastic
outer wing segment	12	12	8	elastic
tails (inner+outer parts)	6	12	1	rigid
fins	6	12	1	rigid
pods	6	6	1	rigid
booms	-	-	1	rigid

spanwise discretization for this case.

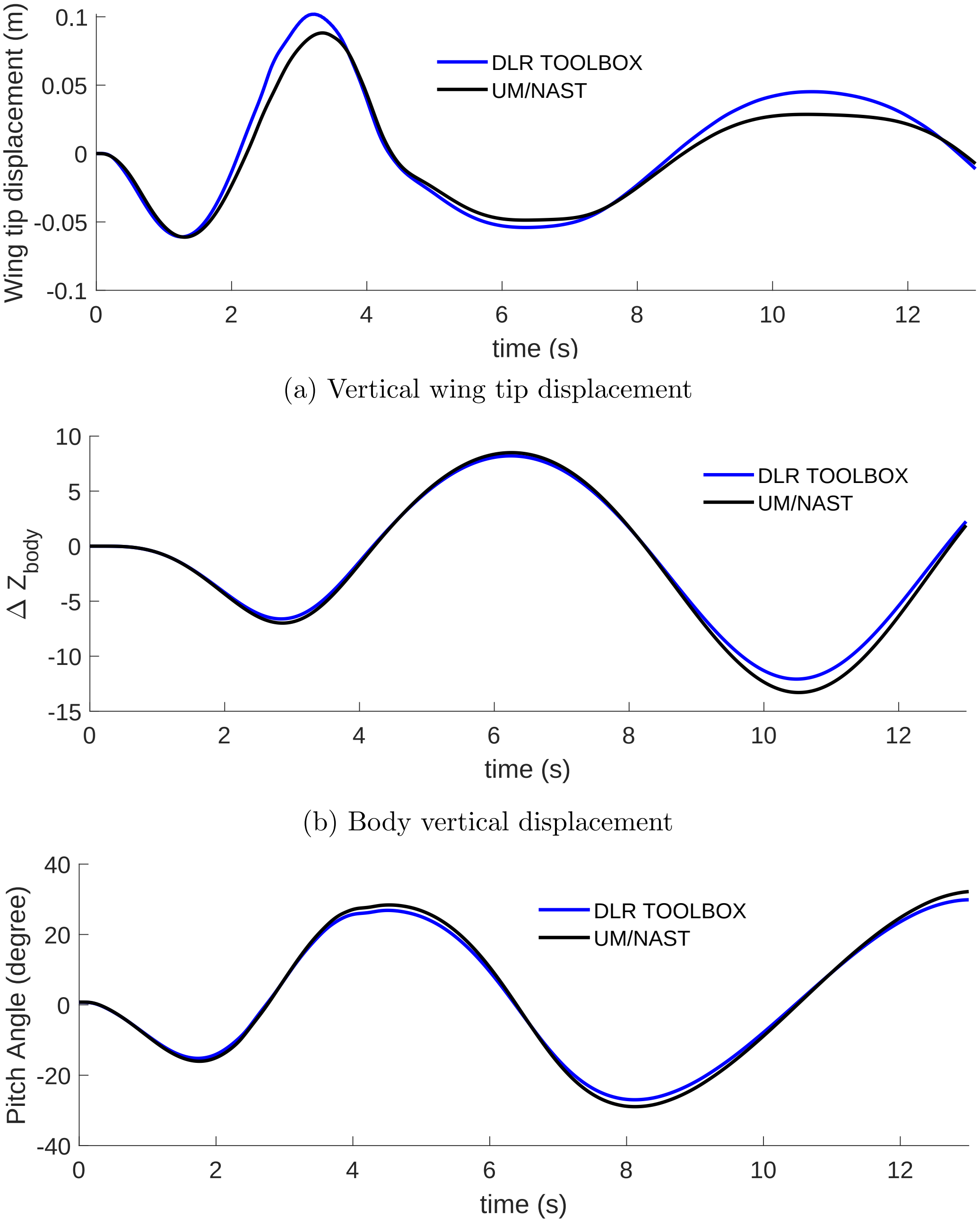
Figure 5.6 shows a comparison for a *1 - cos* vertical gust encounter (gust is moving in the direction of the clamped aircraft), also presented in [82]. The gust maximum disturbance velocity is 0.5 m/s, with a length of 2 m and 10 m. The aircraft is at zero angle of attack and facing an undisturbed freestream velocity of $v = 16$ m/s. Gravity is included. This case was designed for linear deformation regime, allowing the use of MSC.Nastran SOL 146 (which uses DLM for the aerodynamics) and the DLR toolbox with a linear modal approach for the structure.

Slight differences are observed in the decay of the gust disturbances for the short gust, but the agreement is in general very good, especially for the longer gust case, which excites higher displacements.

Free-flight simulations

Table 5.4: Aerodynamic and structural discretization used in UM/NAST

Part	Aerod. discretization		Structural Information	
	chord	span	number elem.	type
inner/mid wing segment	12	12	4	elastic
outer wing segment	12	12	8	elastic
tails (inner+outer parts)	3	6	1	rigid
fins	4	2	1	rigid
pods	2	2	1	rigid
booms	-	-	1	rigid



(a) Vertical wing tip displacement

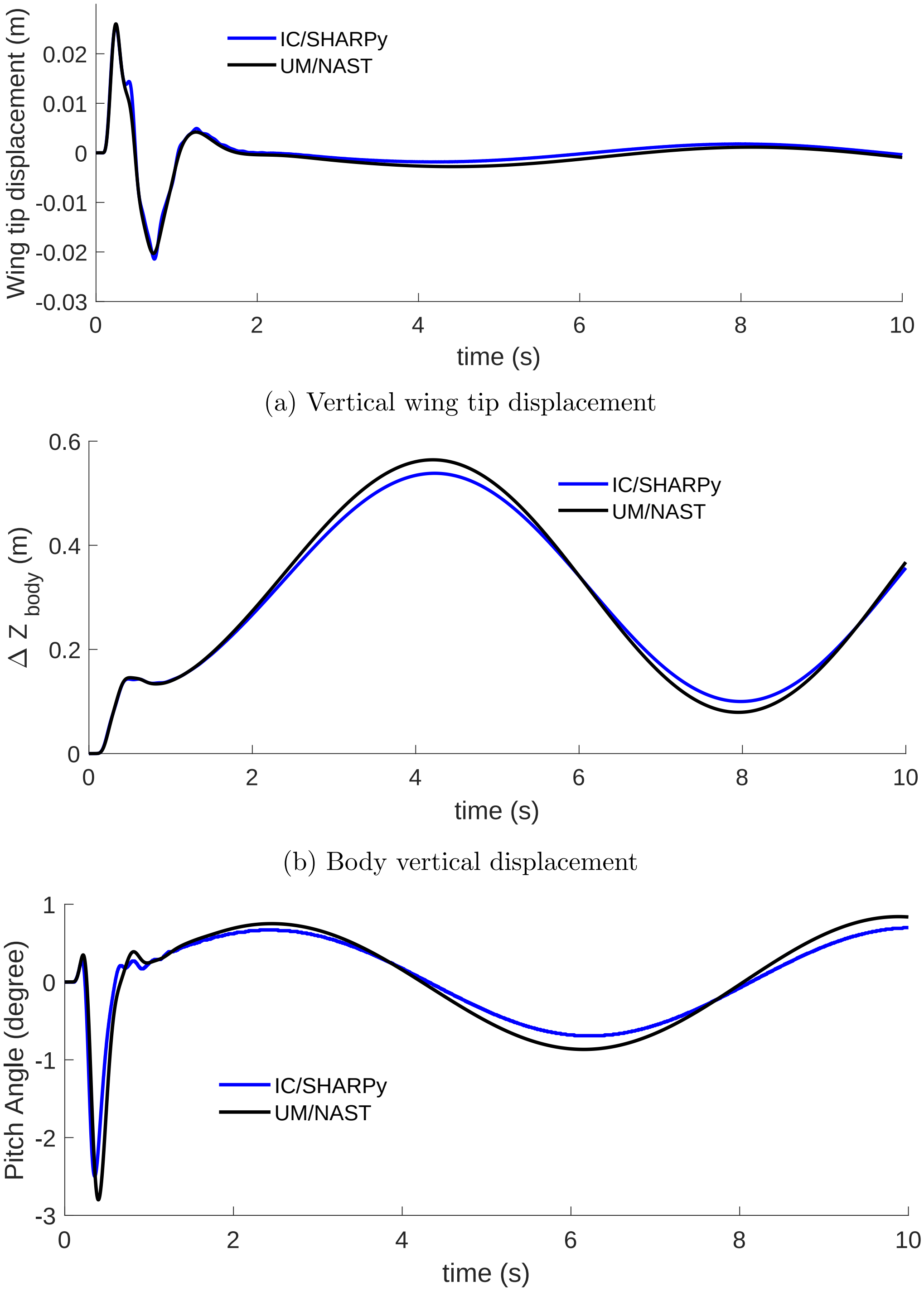
(b) Body vertical displacement

(c) Variation of pitch angle

Figure 5.7: Comparison between UM/NAST and DLR toolbox for tails maneuver with frequency of 0.25 Hz and amplitude of 2 degrees (applied during one cycle only)

For the verification of UM/NAST + uVLM in free-flight simulations, two comparisons are presented: a tails maneuver and a *1 - cos* vertical gust.

For the tails maneuver, a sinusoidal input with frequency 0.25 Hz and amplitude of



(a) Vertical wing tip displacement

(b) Body vertical displacement

(c) Variation of pitch angle

Figure 5.8: Comparison between UM/NAST and IC/SHARPy for free-flight simulation of 1 - gust with maximum velocity of 2.1 m/s and length of 15 m

2 degrees was provided simultaneously to the four horizontal tails. Gravity is included. Table 5.4 summarizes the structural and aerodynamic spanwise discretization for this case. The results for wing tip displacement, vertical body displacement, and pitch angle are presented in Figure 5.7 against DLR toolbox results, which in this case used a nonlinear extended modal approach [72]. An overall good agreement can be obtained, especially for the free-flight quantities. The small differences can be attributed to the different formulations for the coupling and structural approaches.

Figure 5.8 shows a comparison for a free-flight simulation at $v = 14$ m/s passing through a *1 - cos* gust, whose front was initially located at 1 m ahead of the wing root point. Gravity is included. This case is also presented in [1]. As previously commented, SHARPy is based on different Vortex Lattice code and has differences in the structural and coupling approaches. Results are shown for the wing tip displacement, vertical body displacement, and pitch angle. Despite the differences between the codes, an overall good agreement is observed. The aerodynamic and structural discretizations used for UM/NAST, in this case, are the same as in Table 5.1.

5.2 Propeller Aerodynamics

As a first check for the implementation of the propeller aerodynamics, a comparison with results from a vortex particle code developed by Singh and Friedmann [17] was performed. The information about the propeller and conditions used are summarized in Table 5.5. There, the Reynolds number is calculated based on propeller diameter and free-stream velocity. The thrust coefficient from the reference code is $C_T = 0.036$ and, for the same conditions, the thrust coefficient with the newly developed code is $C_T = 0.039$. The axial velocity profile along a line at a distance of one radius behind the propeller is compared in Figure 5.10. The profiles show good agreement, with maximum values of axial velocity slightly higher in the case of the present code. The minor differences are attributed to the different aerodynamic models employed for the blades: Lifting Line in the present code and Vortex Panels for the code of Singh and Friedmann [17].

Table 5.5: Three-bladed propeller parameters

Blade properties	APC 11X5.5 E plus 5-degree pitch angle
Airfoil type	flat plate
Number of blades	3
Propeller RPM	6000
Reynolds number	3.07×10^5
Blade discretization	17 segments
Time step	$T_P/30$
Sigma particles	0.045

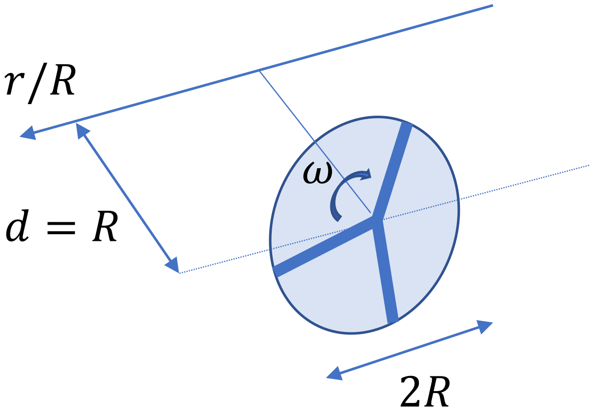


Figure 5.9: Axial profile velocity at a distance of one radius behind the isolated propeller

Further verifications were performed against LDV measurements made by Sundar [2] and numerical results from Cho and Williams [180], who used a frequency domain panel method and a rigid wake model. The propeller used is a two-bladed Purdue propeller model whose blades are straight, with a constant chord of 2 in, a diameter of 12 in, and constant NACA0010 airfoil sections. For this case, the polar tables were generated using $N_{crit} = 5$ for the e^N transition method in XFOIL. More details about this propeller model and the experiment can be found in Sundar [2] and Usab

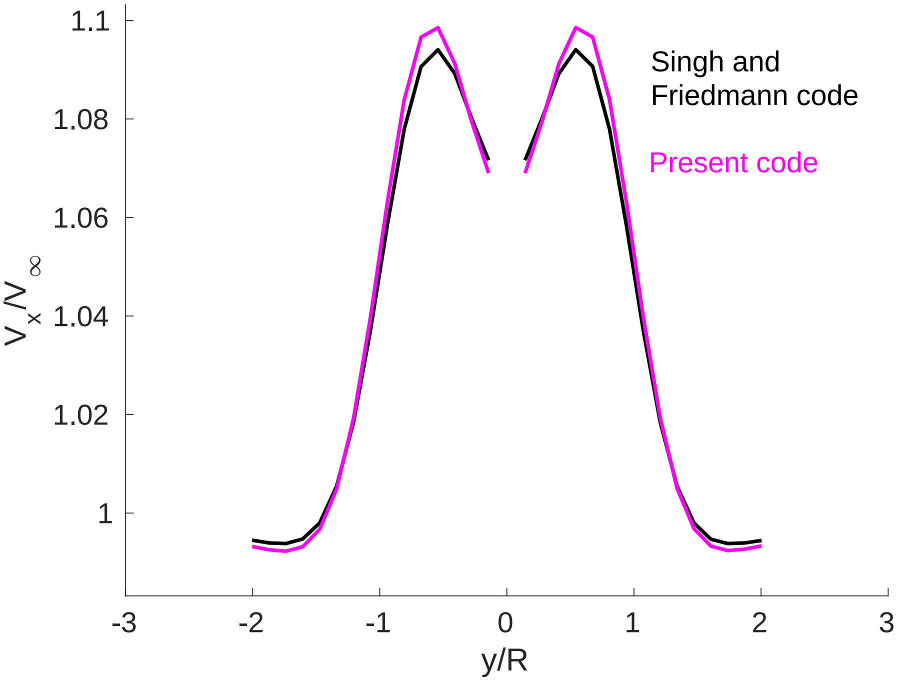


Figure 5.10: Axial profile velocity at a distance of one radius behind the isolated propeller.

et al. [181]. A summary of the propeller parameters used for this comparison is presented in Table 5.6. Figure 5.11 presents a comparison of thrust versus power coefficient. The agreement with experimental results is overall quite good, with some overprediction near windmill conditions, also observed in the numerical results of Cho and Williams [180]. Such overprediction can be possibly attributed to inaccuracy of polar tables for those conditions, which could be related, for example, with the parameter choice for the transition method in XFOIL.

The effect of the propeller on the velocity field behind it was also compared. Figure 5.12 shows the circumferential velocity distribution at a distance of one radius behind the propeller plane and at a radial position of $r/R = 0.93$, where the centerbody effects are less important, at the operating condition of $C_T = 0.14$ and $C_P = 0.30$. Due to lack of more precise information about the angular reference used, just amplitudes are checked and the peaks of axial velocity for experimental and numerical results were aligned for the comparison (thus, the phase is not com-

Table 5.6: Propeller parameters used for comparison with experiment of Sundar [2]

Airfoil type	NACA0010
Number of blades	2
Diameter	12 in
Propeller RPM	6000
Reynolds number	$1.02 \times 10^6 - 1.4 \times 10^6$
Blade discretization	16 segments
Time step	$T_P/120$
Sigma particles	0.0064

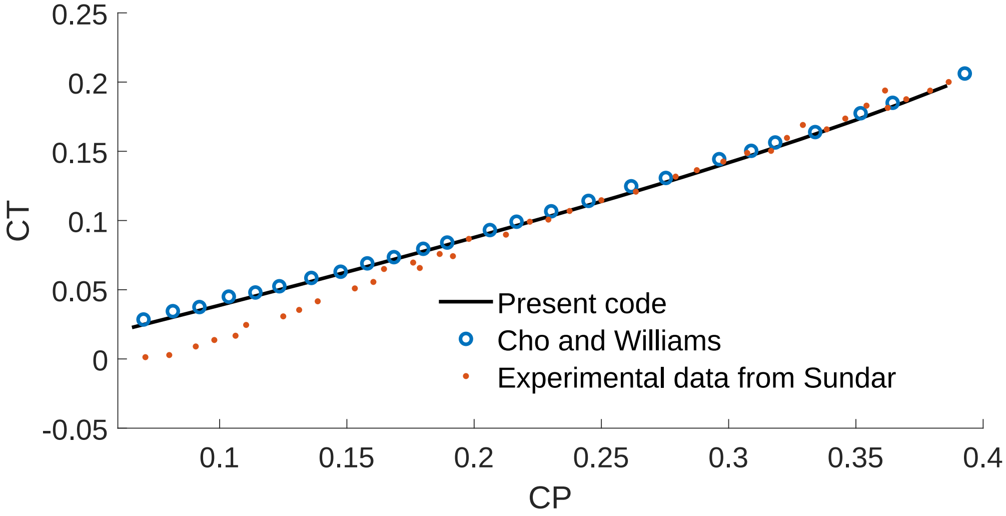


Figure 5.11: Comparison of propeller thrust and power coefficients

pared). One can observe a generally good agreement between the experimental and numerical results, with the variation of the radial component with angular position better captured by the present code.

5.3 Propeller-Wing Interaction

In order to verify the wing-loading under propeller effects, a propeller-wing configuration based on the experiment of Witkowski *et al.* [9] shown in Figure 5.13 (b) was considered. The propeller parameters are the same used for the comparison with experiment of Sundar, listed in Table 5.6 and an advance ratio $J = V_\infty/nD = 1.66$

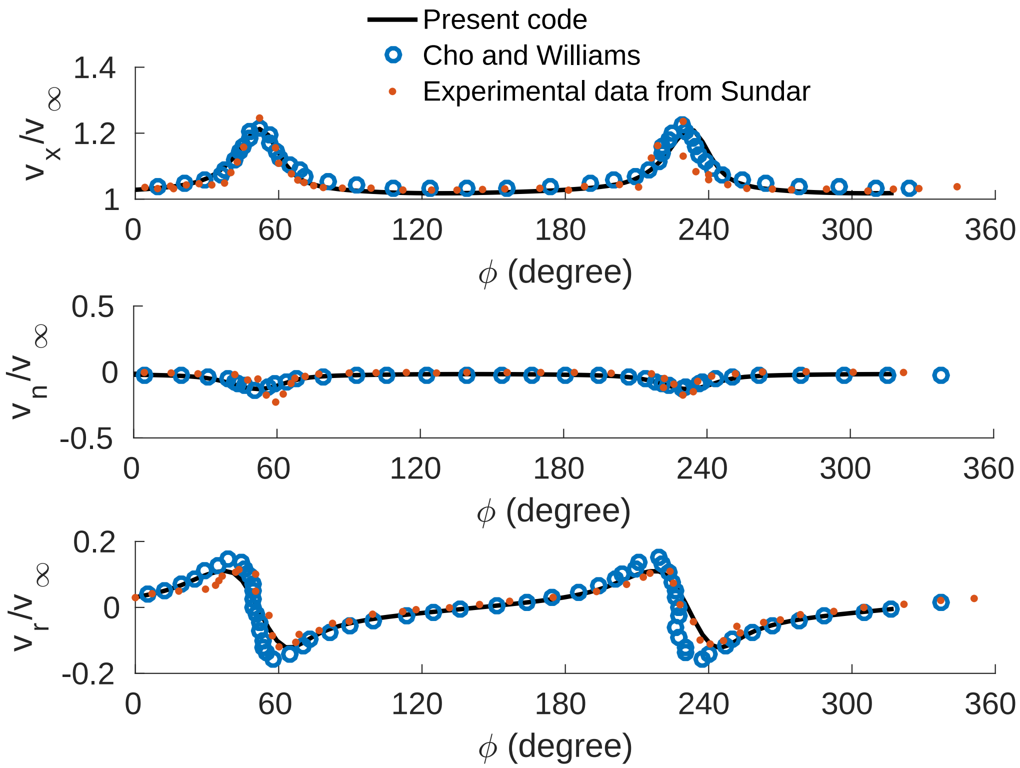
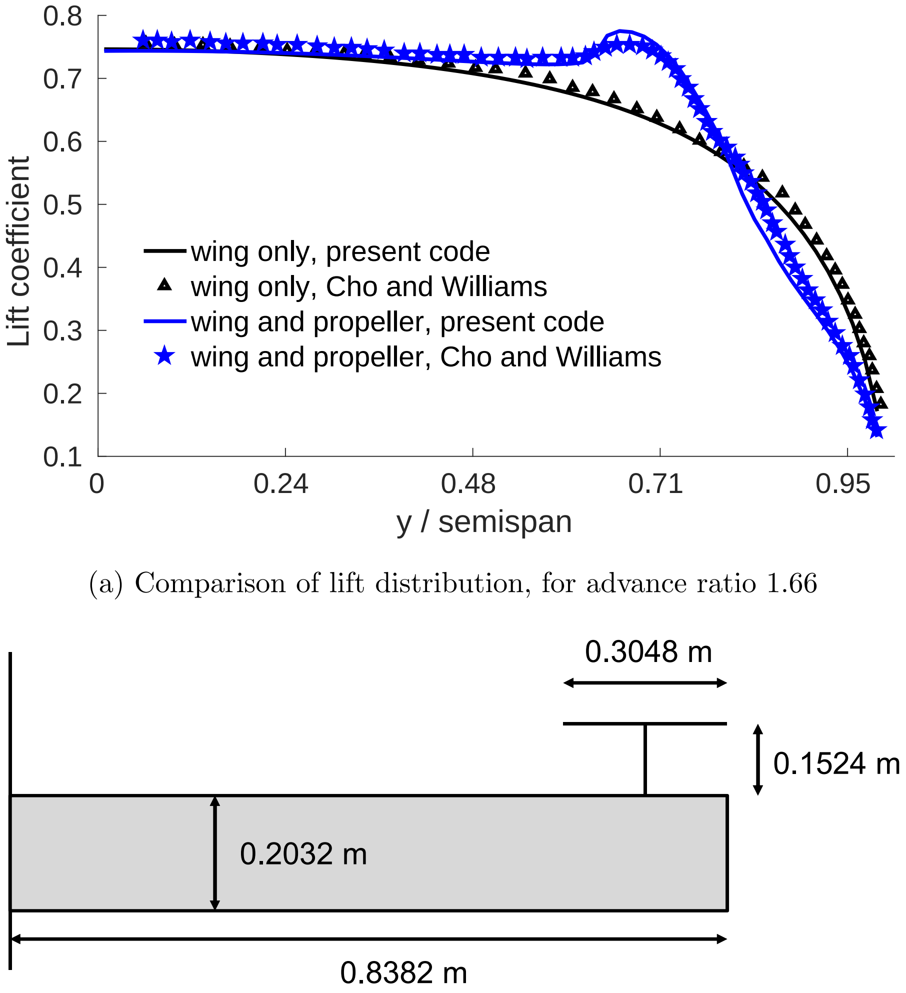


Figure 5.12: Circumferential distribution of velocity components at a distance of one radius from propeller plane and at a radial position of $r/R = 0.93$ ($C_T = 0.14$ and $C_P = 0.30$).

was considered, where n is the revolution per second and D is the propeller diameter. The wing has a rectangular form, with aspect ratio 8.25, constant NACA0010 airfoil section. As experimental data in terms of lift distribution was not available, the comparison was performed against numerical results of Cho and Williams [180], who applied a Frequency Domain Panel method. Figure 5.13 (a) shows a comparison of lift distribution for cases of wing only and wing and propeller. In both cases, the results showed a good agreement, with slight differences that can be attributed to the use of totally different numerical approaches.

A comparison was also made against available integral force data in the form of global lifting surface C_L (for the whole wing) versus angle of attack. However, as one can see in Figure 5.14, the curve for the case without propeller (VLM only) is almost the same as for the case with the propeller. This is because the variations of local



(a) Comparison of lift distribution, for advance ratio 1.66

(b) Top view of the wing-propeller setup

Figure 5.13: Wing-propeller verification.

lift on the two sides of the propeller location approximately cancel each other, not affecting the integral lift significantly.

5.4 Gyroscopic Loads Modeling

In order to verify the implementation of the additional term given by Eq. 2.85, which accounts for propeller gyroscopic moment, a comparison of the additional moment was performed against the analytical solution for a simple case of a rigid wing undergoing a pitching sinusoidal motion with a propeller located in its middle (Figure

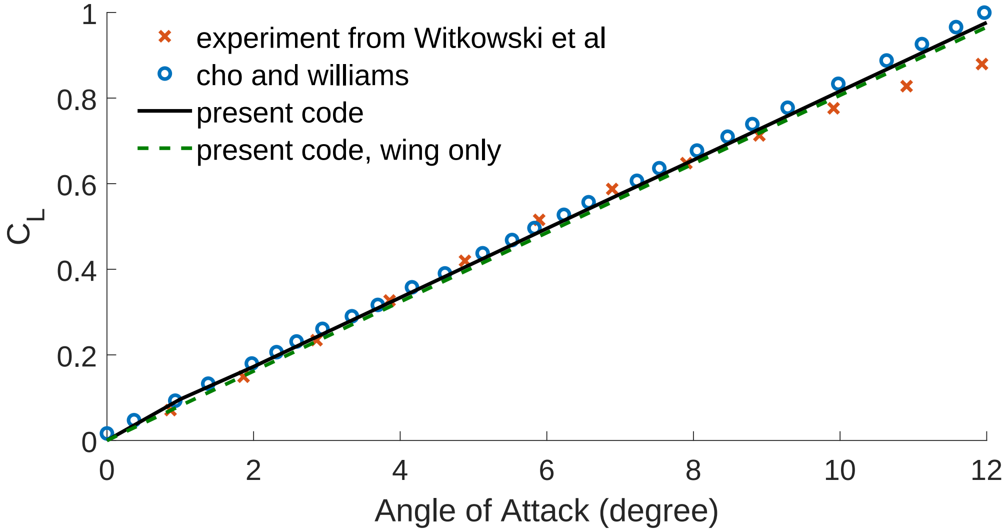


Figure 5.14: Global lift coefficient versus angle of attack

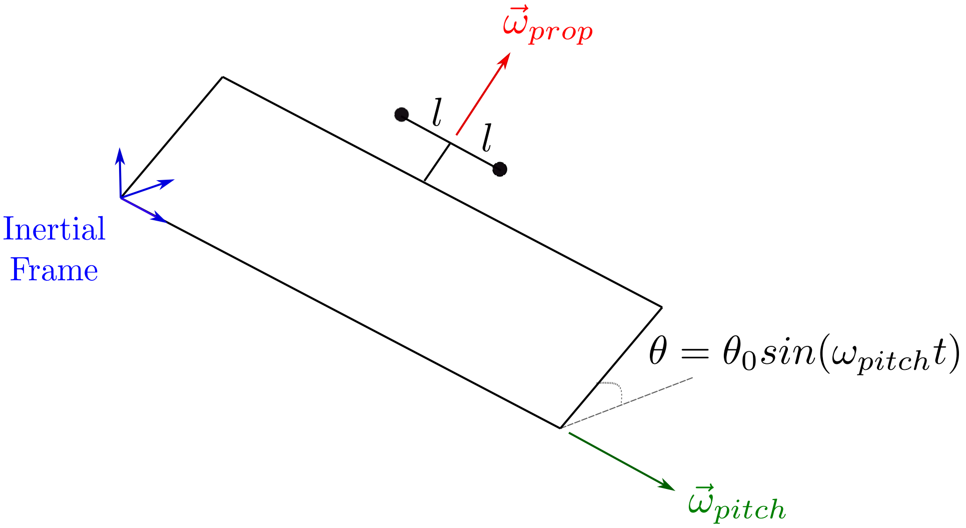


Figure 5.15: Pitching wing with rotating propeller represented by a rigid, massless rod, with concentrated masses on its ends

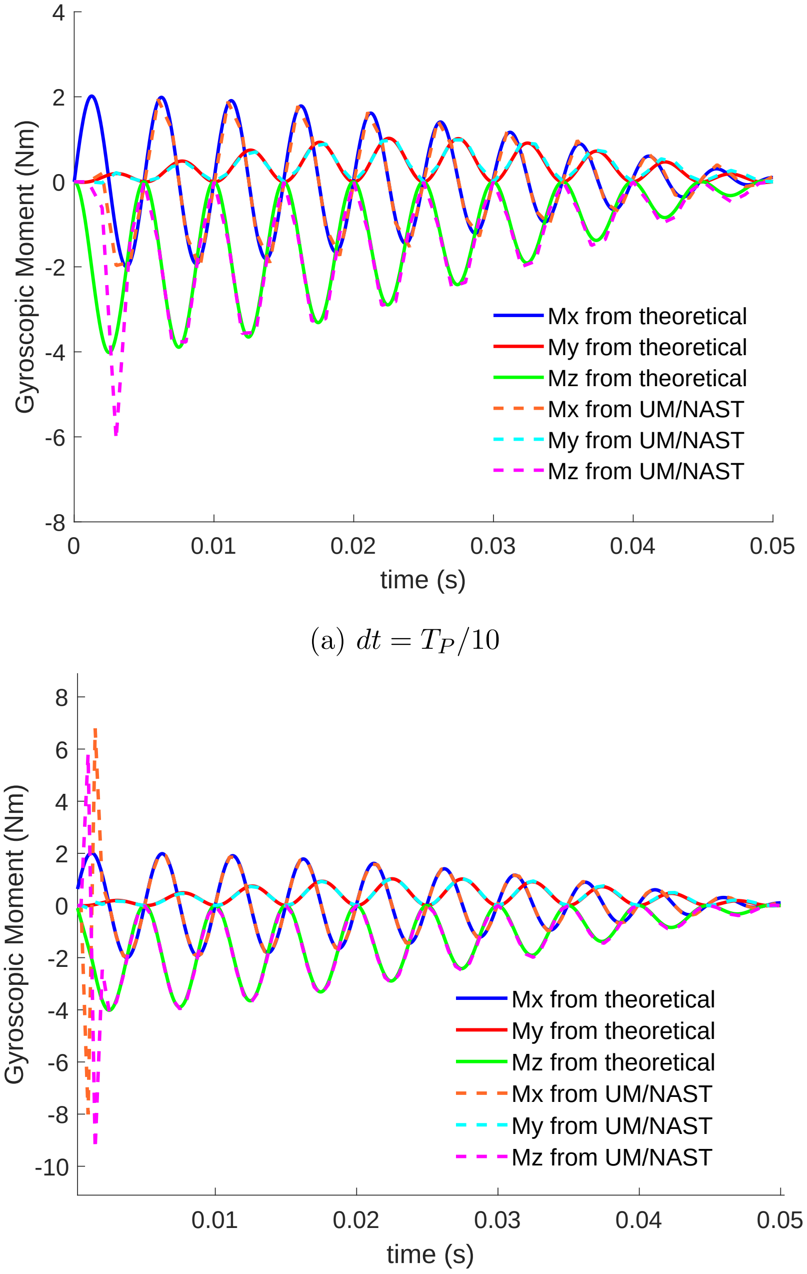
5.15). More details and a MATLAB script of this analytical example are provided in Appendix C, and Table 5.7 summarizes the data for this comparison.

Figure 5.16 shows the agreement between analytical and numerical solutions is good after few steps and tends to improve as the time step reduces. The differences at the first time steps are due to some numerical derivatives not being precisely calculated at the initial time since they are not included as initial conditions. This

Table 5.7: Parameters used for the verification of gyroscopic moment

Propeller RPM	6000
Propeller mass	0.027 kg
Pitch motion amplitude	30 degrees
Pitch frequency	5 Hz

problem can be avoided by waiting for an accommodation time (about two propeller revolutions) before transmitting loads to the structure.



(a) $dt = T_P/10$

(b) $dt = T_P/20$

Figure 5.16: Comparison of analytical and numerical calculation of gyroscopic moment

CHAPTER 6

Propeller Effects on HALE Aircraft

In this chapter, investigations of propeller effects on the static and dynamic aeroelastic response, as well as the aeroelastic stability of a representative HALE aircraft are presented. For this, three models based on the University of Michigan's X-HALE UAS are simulated using the newly enhanced coupled nonlinear aeroelastic-flight dynamics framework. First, the details of the models, the propeller parameters, and the simulation settings are described in Section 6.1. Then, in Section 6.2, the effects of propellers on the static response and aerodynamic coefficients are analyzed and discussed. Next, the dynamic response of the complete aircraft for clamped and free-flight cases for different control surface/loads inputs is investigated in Section 6.3. Finally, investigation of the propeller effects on the aeroelastic stability of a HALE aircraft is presented in Section 6.4.

6.1 Simulation Details

6.1.1 Aeroelastic Models

For the numerical studies, three models based on the University of Michigan's X-HALE UAS are considered:

i) X-HALE complete vehicle: the same model as presented in Chapter 5, with properties detailed in Appendix B. This model was used for the static and dynamic responses investigations;

ii) X-HALE isolated wing: X-HALE wing only, in order to isolate the effects of the tails and other surfaces on the wing. This model was used for static investigations;

iii) X-HALE UAS with added tip mass: X-HALE model with an added tip mass (in both wings) of 0.5 kg located 0.36 m behind the wing trailing edge. This tip mass was added such to lower the occurrence of unstable aeroelastic behavior to a range of velocities within the aircraft original flight envelope. This model was used for the aeroelastic stability investigations.

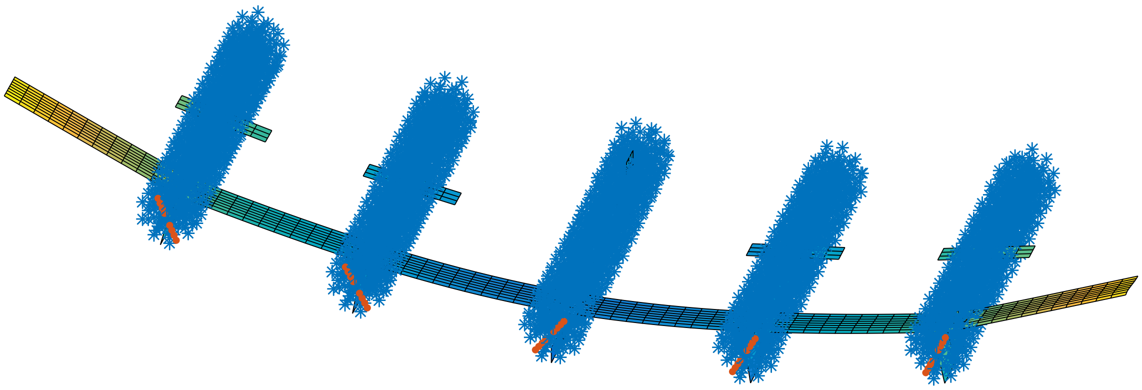


Figure 6.1: View of deformed X-HALE with particles shed up to the cut-off distance. The propellers on the right wing rotate at opposite orientation as those on left wing.

Table 6.1: Aerodynamic and structural discretization used in UM/NAST

Part	Aerod. Discrcetization		Structural Information	
	chord	span	number elem.	type
inner/mid wing segment	8	12	4	elastic
outer wing segment	8	12	8	elastic
tails (inner + outer parts)	3	6	1	rigid
fins	2	4	1	rigid
pods	2	2	1	rigid
booms	-	-	1	rigid

For all cases, the propellers are located 20 cm ahead and 2.8 cm below of the wing leading edge, with a pitch angle of 5 degrees between the propeller axis and the

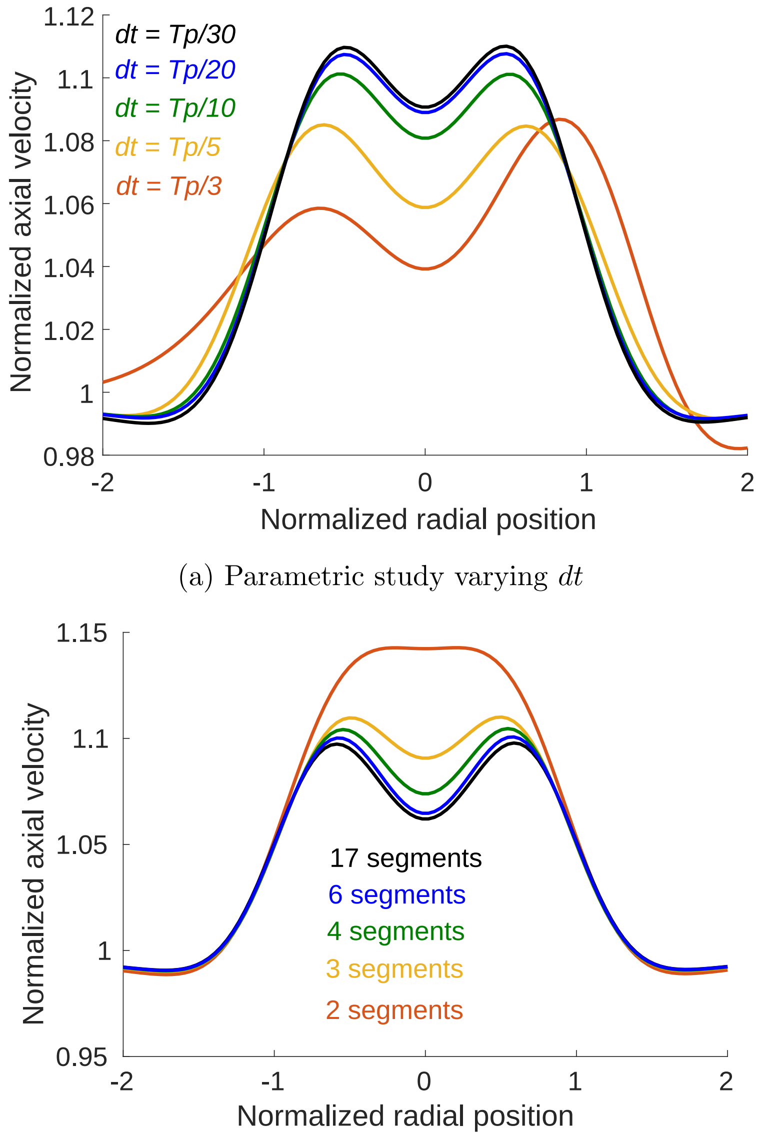
X-HALE wing, as the wing has an incidence angle with respect to the pods. In order to reduce asymmetric loads, the propellers on the right wing have a different direction of rotation with relation to the propellers on the left wing, as can be seen in Figure 6.1. The two propellers along the right wing rotate in the clockwise direction, and the remaining propellers rotate in the counterclockwise direction (for one looking from behind the wing). The aerodynamic and structural discretizations are summarized in Table 6.1.

6.1.2 Propeller Parameters

To determine the choice of parameters for the propeller solver, seeking a balance between accuracy and computational cost (with a smaller number of particles until cut-off distance), a parametric analysis for the blade discretization and the choice of time step was performed with results obtained for the three-bladed propeller used for comparison in Figure 5.10. Based on Figure 6.2 and considering that the objective of this work is to capture the dynamic influence of propellers, the parameters selected for the simulations are time step $dt = T_p/10$, and blade discretization of $N_{seg} = 4$.

Next, consider a propeller model also based on APC 11x5.5 thin-electric propeller with geometrical data described in Brandt [182]. Due to the lack of information about the airfoil type along the blades, a constant typical APC airfoil, NACA 4412, is assumed. Table 6.2 summarizes the data used for the propeller as well as conditions and choice of parameters. In this study, the value of the parameter σ, the smoothing parameter or core radius, is calculated by multiplying a reference length (related to the minimum distance between particles), h, by a factor c, chosen based on the convergence of C_T. The reference length, in this case, is chosen as the length of the arc described by the blade root in one time step.

For the following simulations $c = 1.5$ is used, resulting in $\sigma = 0.0195$. For this value of σ, C_T converges to a value of approximately 0.022, resulting in a thrust of



(a) Parametric study varying dt

(b) Parametric study varying blade discretization

Figure 6.2: Parametric study varying (a) time step and (b) blade discretization

1.68 N. Also for the cases considered here, shed particles (related to the variation of bound circulation with time) have a strength about three orders of magnitude smaller than that of trailing particles. Therefore, their effect is not significant, and they were not considered for simulations, reducing the number of particles and saving computational time.

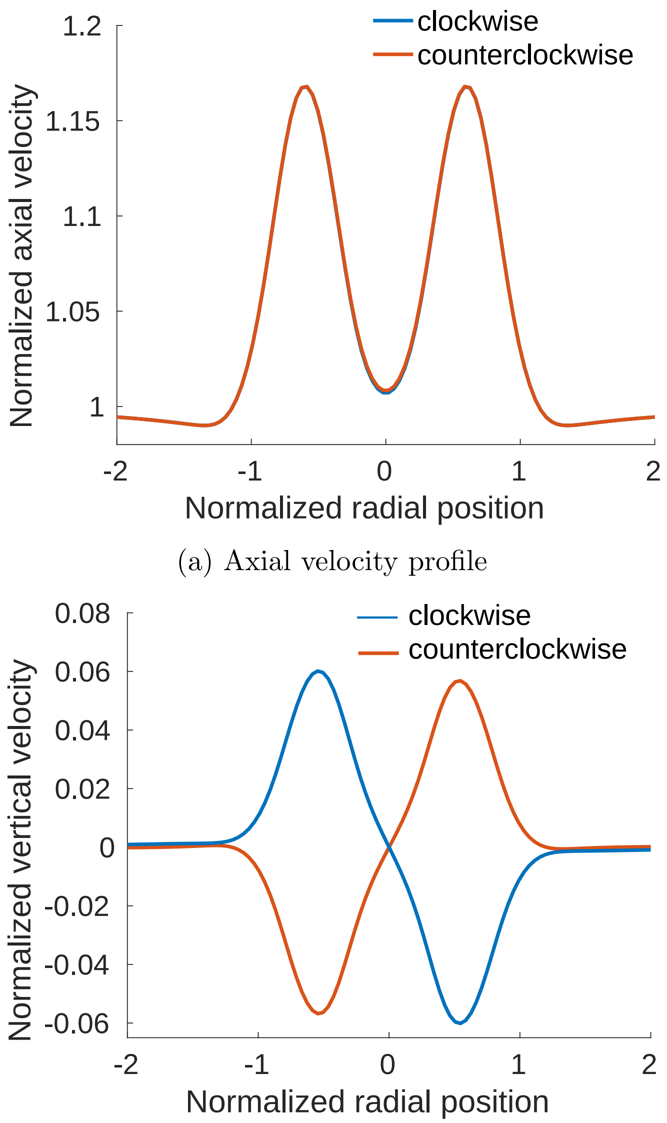
Figure 6.3 presents the profile of axial and vertical velocities for the two-bladed propeller at one radius behind it. The behavior is in accordance with that described

Table 6.2: Two-bladed APC 11X5.5E propeller parameters

Blade properties	APC 11X5.5 E
Airfoil type	NACA 4412
Propeller mass	0.023 kg
Number of blades	2
Propeller RPM	6000
Free-stream velocity	14 m/s
Blade discretization	4 segments
Time step	$T_P/10$
Reference length (h)	0.013
Factor sigma (c)	1.5
Sigma particles	0.0195

in the literature, *e.g.*, Veldhuis [13], Khan and Nahon [183], and Agostinelli *at al.* [34], with a symmetric pattern for axial velocity and asymmetric distribution of vertical velocity. Also, the location of the maximum vertical velocity depends on the blade direction of rotation, with positive vertical velocities at the side where the blade goes up.

As discussed in Veldhuis [13], the increase in axial velocity (and dynamic pressure) does not affect the local angle of attack but increases total lift produced. On the other hand, vertical velocity affects the local angle of attack, which increases in the region where upwash is generated and decreases in the region of downwash. Then, in the upwash side, an increase in axial velocity and local angle of attack increases lift with relation to a configuration without a propeller. On the downwash region, the effects have contrary trend (increase in dynamic pressure tends to increase lift, but a decrease in angle of attack tends to reduce lift) and the results are a smaller increase of lift on that side when compared to the no-powered configuration.



(a) Axial velocity profile

(b) Vertical velocity profile

Figure 6.3: Profiles of axial and vertical velocities at one radius behind the two-bladed propeller (clockwise or counterclockwise directions are defined with relation to one looking from behind the propeller).

6.1.3 Pre-setting Parameters for the Aeroelastic Simulations

As stated previously, in the simulations to follow just trailing particles are considered (magnitude of shed particles is negligible for this case), with five trailing particles per blade. A cut-off distance of two radii after the end of the X-HALE vertical tails and three radii after the trailing edge of the isolated wing were applied to the complete

X-HALE and the isolated wing, respectively. In the aeroelastic solver, a trapezoidal method was employed to integrate the equations of motion, a stiffness-proportional damping coefficient of 0.005 s and a time step dt of 0.002 s (for 6000 RPM) and 0.0017 (for 7000 RPM case in Section 6.4) were considered (with a sub-time step of $dt/2$ for the propellers). Induced and profile drags were taken into account, and the loads acting on the propeller hub are transmitted to the wing structure. The dynamic viscosity is $\mu = 1.7855\ 10^{-}5$ N.s/m^2 and the air density $\rho_\infty = 1.225$ kg/m^3. Gravity effects are also considered.

Finally, in order to investigate the importance of various propeller effects, three kinds of results were obtained for dynamic simulations, including:

i) propeller equivalent thrust only: just thrust, as a concentrated force at spanwise locations $y = -2/3, -1/3, 0, 1/3,$ and $2/3$ of semispan, is considered;

ii) propeller aerodynamics: considering the effect of slipstream and all resultant aerodynamic loads at the propeller hub, which are transferred to the wing structure;

iii) propeller aerodynamics and gyroscopic moments: the same effects as in (ii) plus the modeling of gyroscopic moments, including all propeller effects.

In the next subsections, additional information for the parameters used for each category of simulation is provided.

6.1.3.1 Static and Clamped Cases

For the static and clamped dynamic cases, the simulations were performed considering a centerbody angle of attack of 2 degrees, zero aileron deflection and sideslip angle and, when applicable to the model, zero tail elevon deflections, as summarized in Table 6.3. Also, for the cases with thrust only (without including other propeller effects), it is modeled as a concentrated force given by the thrust produced by an isolated propeller at spanwise locations $y = -2/3, -1/3, 0, 1/3$, and $2/3$ of semispan.

A free-stream speed of $v = 14$ m/s, for the investigations of static and dynamic response, sections 6.2 and transient results. For the aeroelastic stability investigations, different values of velocity were consired, as indicated in Section 6.4.

Table 6.3: Parameters used for clamped cases

Centerbody angle of attack (AoA)	2 degrees
Sideslip angle (β)	0 degree
Aileron deflection (δ_a)	0 degree
Tail elevon deflection (δ_t)	0 degree
Symmetric thrust (T_s)	1.68 N
Differential thrust (T_a)	0 N

6.1.3.2 Free-Flight Simulations

For the free-flight simulations, the initial conditions for the cases with thrust only are those for the complete X-HALE vehicle trimmed for straight-level flight at a free-stream speed of $v = 14$ m/s and altitude $h = 30$ m. The trim parameters are: centerbody angle of attack (AoA), sideslip angle (β), asymmetric aileron deflections (δ_a), symmetric tail elevon deflections (δ_t), symmetric thrust (T_s) (same for all motors), and differential thrust (T_a) (asymmetric thrust at motors near wing tip). Those values were calculated using the trim solver in UM/NAST and are listed in Table 6.4. The thrust is modeled by a concentrated force at spanwise locations $y = -2/3, -1/3, 0, 1/3$, and $2/3$ of semispan. Notice the small aileron and differential thrust (yaw control) that result due to a slight asymmetry on the model (which corresponds to an actual UAS [178]) in terms of mass distribution.

The initial conditions for the free-flight simulation, including propellers, were determined by:

i) running the static simulation with propellers for the complete vehicle clamped

Table 6.4: Straight-level flight trim parameters for X-HALE at $v = 14$ m/s (without propeller effects other than thrust)

Centerbody angle of attack (AoA)	2.54 degrees
Sideslip angle (β)	0.044 degrees
Aileron deflection (δ_a)	-0.144 degrees
Tail elevon deflection (δ_t)	1.03 degrees
Symmetric thrust (T_s)	1.28 N
Differential thrust (T_a)	-0.009 N

at the origin of the body frame (centerbody), using parameters from the trim without other propellers effects than thrust and identifying residual forces at the clamped point, and;

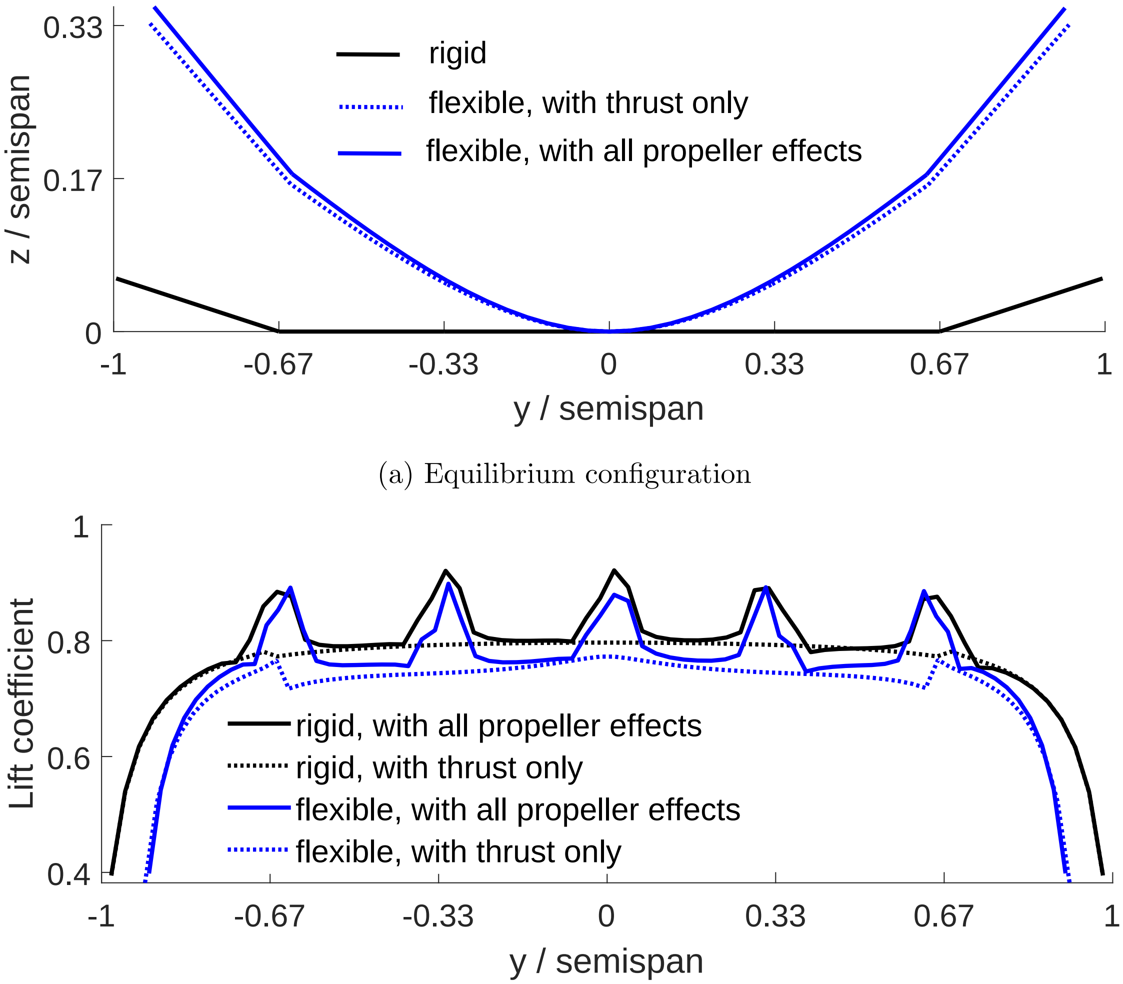
ii) applying additional forces at the origin to balance residual forces to achieve equilibrium condition for free flight. The forces to balance the residual are presented in Table 6.5.

Table 6.5: Additional forces at the origin of the body frame for initial equilibrium in free-flight case with propellers

F_x	F_y	F_z	M_x	M_y	M_z
-0.13 N	-2.02 N	-4.72 N	-0.16 Nm	-0.38 Nm	0.008 Nm

6.2 Aeroelastic Static Response

In order to better understand the effects of propellers on the lift distribution of the flexible aircraft, the static equilibrium condition for the clamped isolated X-HALE wing and the complete X-HALE vehicle are analyzed for the conditions specified in Table 6.3.



(a) Equilibrium configuration

(b) Lift distribution

Figure 6.4: Clamped isolated wing at $v = 14$ m/s and $AoA = 2$ degrees

6.2.1 Isolated X-HALE Wing

In Figure 6.4 (a) a comparison is given for the clamped wing equilibrium state with all propellers effects and with thrust only, for flexible and rigid configurations at $v = 14$ m/s and $AoA = 2$ degrees. As one can see, there is a slight increase in wing deflection for the case with all propeller effects when compared to the one with concentrated thrust only, with an increase in wing-tip displacement (from its undeformed configuration) of 8%, corresponding to a displacement of 2% of the semispan. The wing tip vertical position after deformation in the presence of propeller effects corresponds to 35% of the wing semispan.

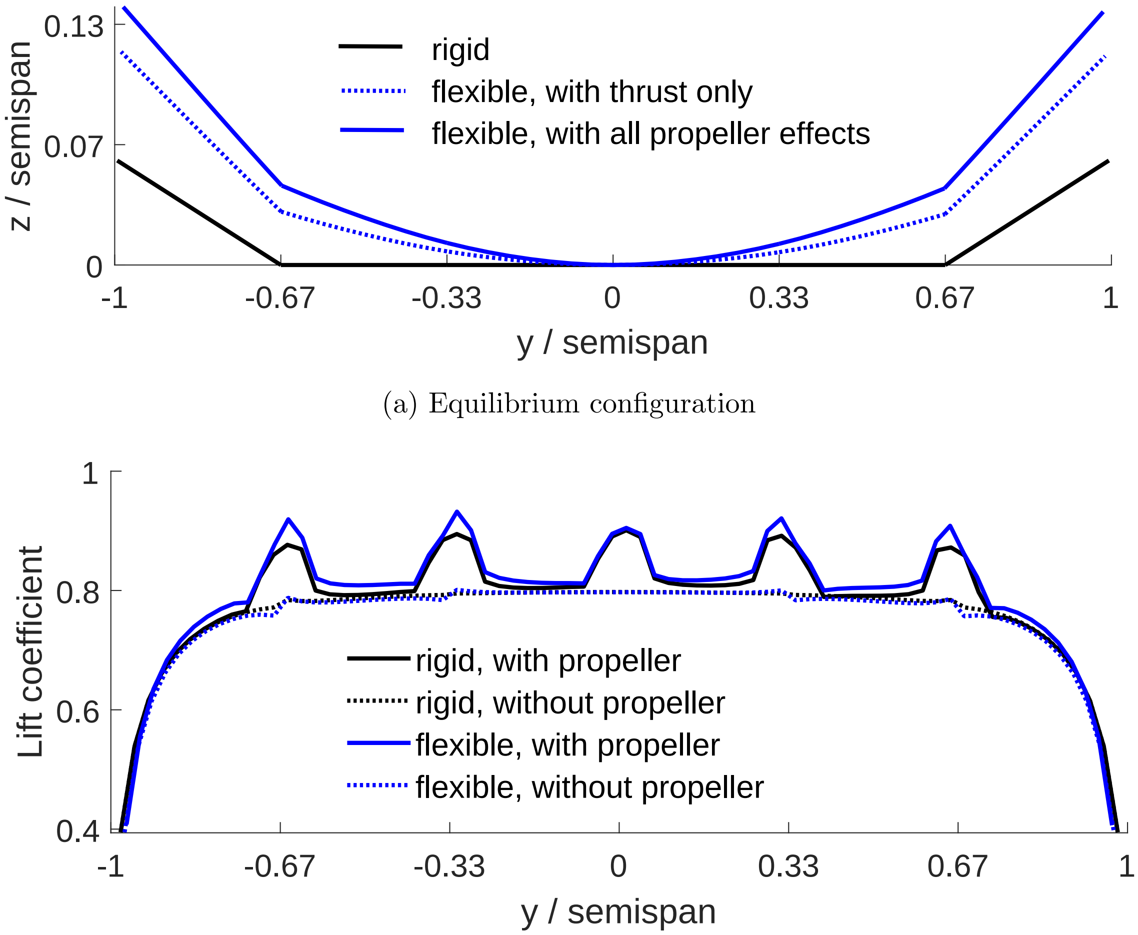
A comparison of lift distribution for the rigid and flexible cases with all propeller effects and with thrust only for the same conditions as above is presented in Figure 6.4 (b). In both rigid and flexible cases, the propeller slipstream causes a local increase in lift close to the propeller locations, with a peak located on the side where the blade goes up. In general, both flexible cases present a smaller value of lift, indicating a smaller effective angle of attack in those cases. As can be seen in Table 6.6, when all propeller effects are included rigid and flexible cases show a similar increase in total lift (between 3% and 4%) , but a higher increase in total drag (between 5% and 6%), resulting in a slight decrease of about 2% in the c_L/c_D ratio.

Table 6.6: Comparison of c_L and c_D for the clamped isolated wing at $v = 14$ m/s and $AoA = 2$ degrees

Case	c_L	c_D	c_L/c_D
Rigid, with all propeller effects	0.7669	0.0185	41.45
Rigid, with thrust only	0.7425	0.0176	42.19
Flexible, with all propeller effects	0.6337	0.0182	34.82
Flexible, with thrust only	0.6108	0.0172	35.51

6.2.2 Complete X-HALE

Similarly as for the isolated wing, Figure 6.5 (a) presents a comparison for the wing of the clamped X-HALE model in equilibrium state with all propeller effects and with thrust only, for flexible and rigid configurations at $v = 14$ m/s and $AoA = 2$ degrees. In comparison with the absolute deflections of the isolated wing, there is a more moderate deflection with wing tip vertical position reaching 14% of wing semispan, instead of 35% as in the isolated wing case. This smaller deflection is related to the higher weight of the complete aircraft (about four times the weight of the isolated wing) and the role of tails and fins in moderating the deflection by counterbalancing wing loads. However, in terms of relative increase in displacements, the differences



(a) Equilibrium configuration

(b) Lift distribution

Figure 6.5: Wing of the clamped X-HALE vehicle at $v = 14$ m/s and $AoA = 2$ degrees

with all propeller effects and with thrust only for the complete vehicle are higher than for the isolated wing, with an increase in wing tip displacement (from its undeformed configuration) of about 42%, which corresponds to 2.5% of wing semispan.

The corresponding distributions of lift for the X-HALE wing at the same conditions as in Figure 6.5 (a), considering flexible and rigid configurations with all propeller effects and with thrust only, are compared in Figure 6.5 (b). For this case, the distributions of lift between flexible and rigid configurations are similar, which may have to do with the moderate displacements of the wing for the complete aircraft model. The flexible case has a slightly higher increase in lift due to the inclusion of all propeller effects than the rigid one, as it can also be observed in Table 6.7, with a 3.5% and 4.8% increase of lift for rigid and flexible cases, respectively. As in the case of the isolated wing, also an increase in drag is observed, although by a smaller factor (about 1.4% for both, rigid and flexible cases), such that the ratio c_L/c_D has a slight increase of about 2% for rigid cases and about 3.5% for flexible cases when all propeller effects are included.

In Figure 6.5 (b), one can notice that the flexible configuration with all propeller effects has a slight increase in lift distribution between two propeller locations compared to the case with thrust only, which can be associated with the increase in wing local angle of attack due to its twist caused by the new tails loading. In fact, as one can observe in Table 6.8, for the flexible case, the inboard tail shows a reduction of lift when all propeller effects are included, which causes a positive twist of the wing, increasing lift. By its turn, the outboard tail shows an increase in the lift when propellers are present, contributing to reduce the local angle of attack in that region. The overall effect, however, is the increase in wing local angle of attack, increasing wing lift, as we can see in Figure 6.6, with an increase in the twist in the inboard tail region and a decrease in the outboard tail region. This should explain also the higher peaks for the flexible case considering all propeller effects, as the propellers there are operating in a higher local angle of attack than the propellers in the rigid case.

Table 6.7: Comparison of c_L and c_D for the complete clamped X-HALE vehicle at $v = 14$ m/s and $AoA = 2$ degrees

Case	c_L	c_D	c_L/c_D
Rigid, with all propeller effects	0.7694	0.0375	20.52
Rigid, with thrust only	0.7433	0.0370	20.09
Flexible, with all propeller effects	0.7601	0.0374	20.32
Flexible, with thrust only	0.7252	0.0369	19.65

Figure 6.7 shows a comparison of the effect of the load distribution on the tails. As already discussed, we can see a reduction of lift for the inner tail, while the outboard

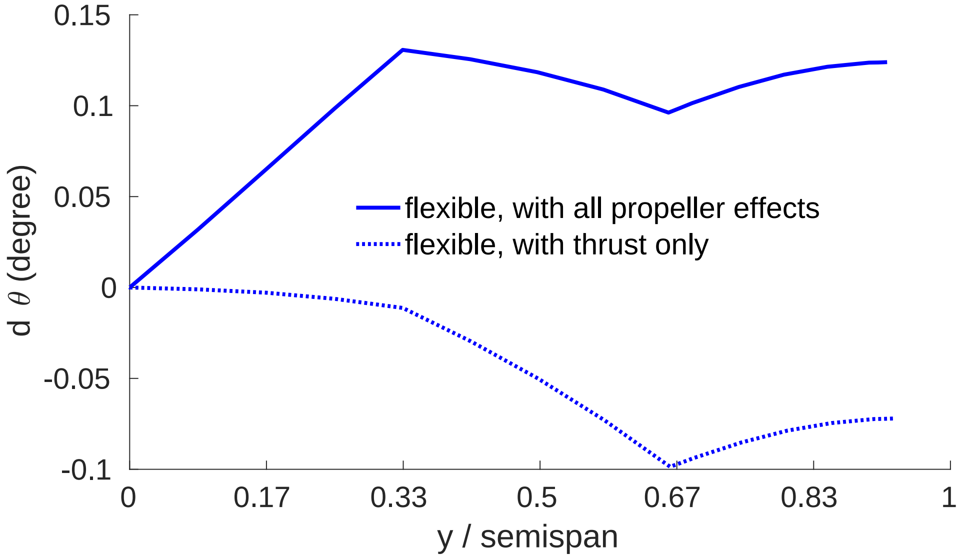


Figure 6.6: Twist distribution along the right wing of clamped X-HALE vehicle at $v = 14$ m/s and $AoA = 2$ degrees

one has an increase in lift. The inner tail is situated behind a ventral fin surface and this may justify the different effects of propeller influence on the inboard tails.

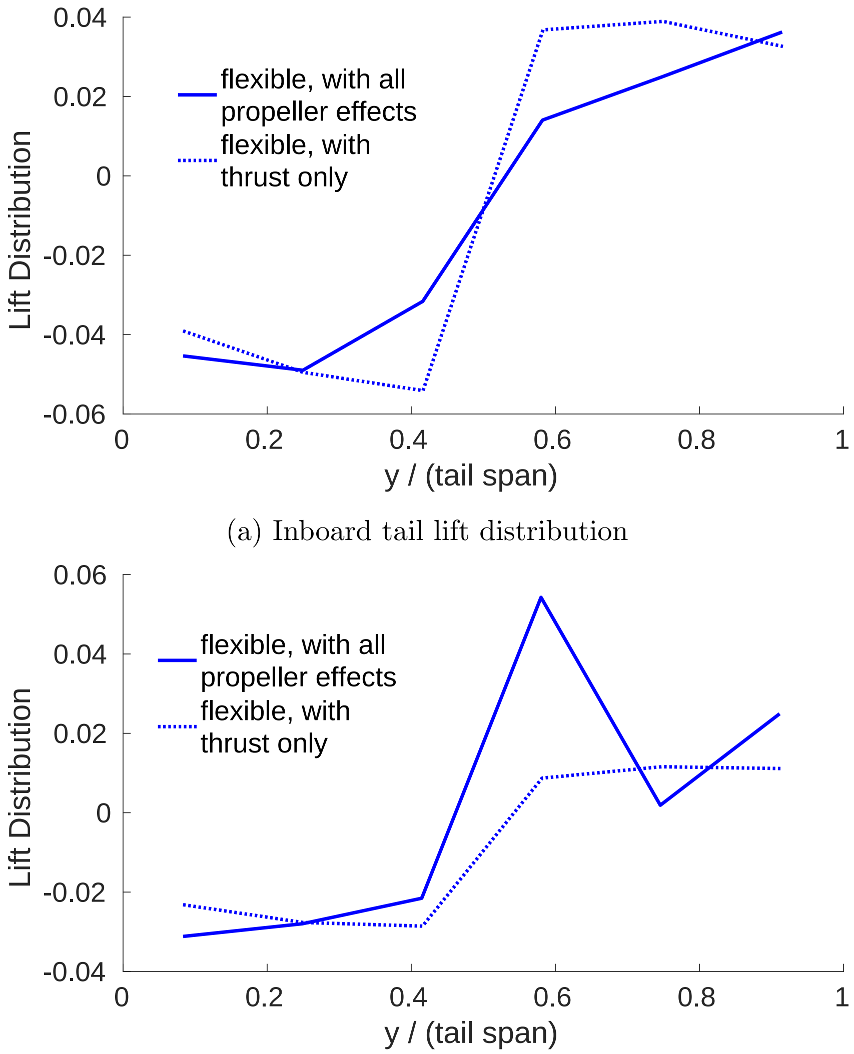
Table 6.8: Comparison of c_L for the inboard and outboard tails of the complete clamped X-HALE vehicle at $v = 14$ m/s and $AoA = 2$ degrees

Case	c_L inboard tail	c_L outboard tail
Rigid, with all propeller effects	-0.1032	-0.1289
Rigid, with thrust only	-0.0155	-0.0735
Flexible, with all propeller effects	-0.1116	0.0008
Flexible, with thrust only	-0.0756	-0.1056

6.2.3 Conclusions

From the static cases investigated, it is concluded that:

- Propellers were found to influence wing equilibrium shape and lift distribution, with higher impact on a wing-tail aircraft configuration where the tails are also influenced by the propeller slipstream.



(a) Inboard tail lift distribution

(b) Outboard tail lift distribution

Figure 6.7: Lift distribution for the inboard and outboard tails of the clamped X-HALE vehicle at $v = 14$ m/s and $AoA = 2$ degrees

- For the wing-tail aircraft configuration, there was a decrease in the tails total lift for the rigid model with all propeller effects when compared to the same rigid model with just thrust. The behavior was different for the flexible vehicle case, with an increase in total lift on the outer tails and a decrease on the inner tails (possibly due to the presence of fins in the X-HALE case).

- Lift and drag coefficient increases due to the inclusion of all propeller effects were found in both rigid and flexible X-HALE models, with a slight increase in the c_L/c_D ratio of about 2% and 3.5% for rigid and flexible cases, respectively. For an isolated wing, this caused a slight reduction in c_L/c_D ratio of about 2%.

6.3 Aeroelastic Transient Response

6.3.1 Clamped Case

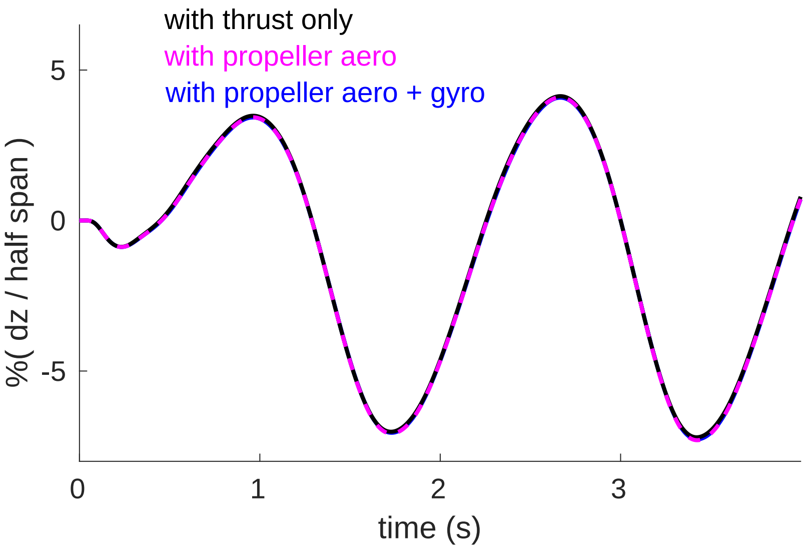
In order to evaluate the response of the aircraft with propellers for a bending excitation a continuous sinusoidal bending moment of amplitude 35 Nm with a frequency of 0.59 Hz, close to the wing first natural bending frequency, was considered. This loading amplitude was selected to achieve a wing tip vertical displacement of about 5% of semispan. A comparison of the displacements is presented in Figure 6.8. No difference is noticed for vertical displacement and just slight differences are observed for twist displacement, although for very small amplitudes.

To investigate the aeroelastic response of the clamped aircraft to a torsional excitation, a case of a step input of 15 degrees applied simultaneously at all horizontal tails at $t = 0$ s was considered. As presented in Figure 6.9, the difference is more noticeable in both wing tip displacement and twist, with a 13% increase in vertical displacement and an initial tip twist of about 1.5 times higher for the cases with propellers. Negligible differences are found between cases with and without gyroscopic moment.

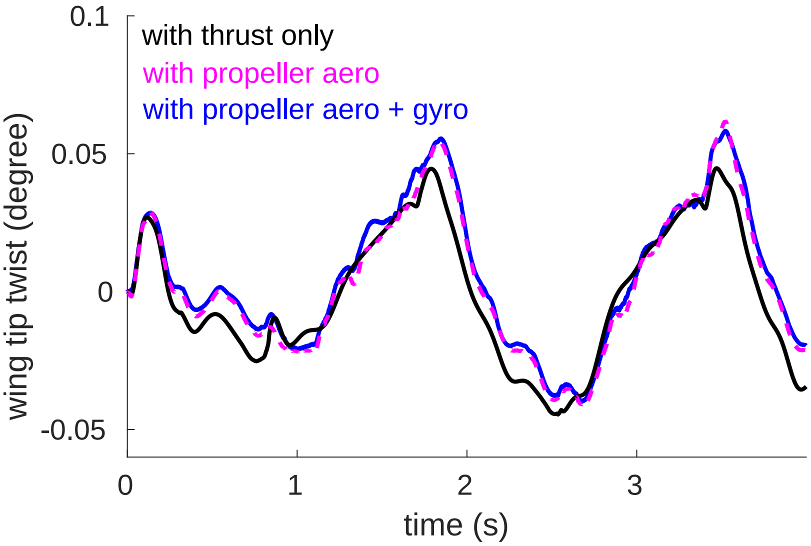
From the cases considered, propeller seems to affect more the response to a torsional than a bending excitation and the differences in angular displacements are usually higher than for vertical ones.

6.3.2 Free-flight Case

To check the influence of propellers to a longitudinal (symmetric) disturbance in free flight, a tip bending excitation similar to the one of the clamped cases is

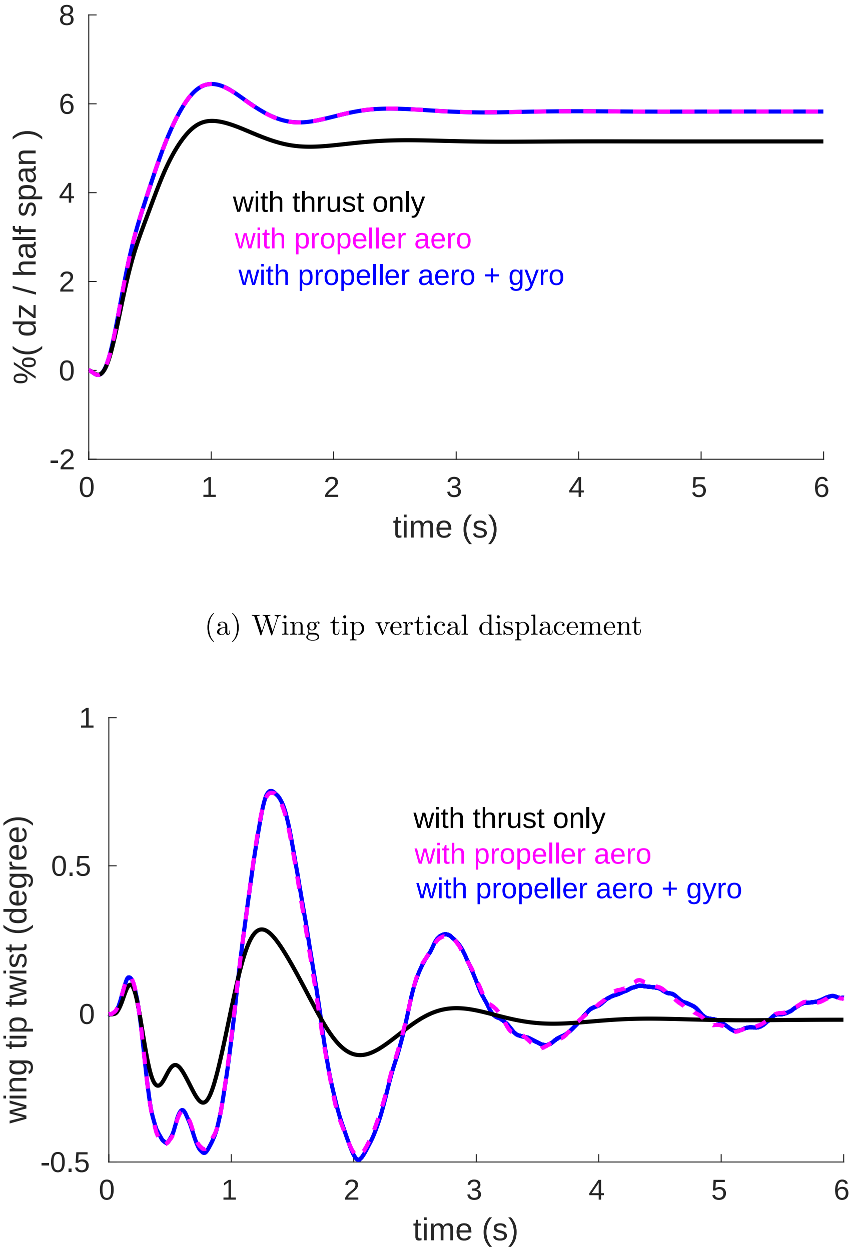


(a) Wing tip vertical displacement



(b) Wing tip twist

Figure 6.8: Response to a continuous sinusoidal tip bending moment signal of amplitude 35 Nm and frequency 0.59 Hz at $v = 14$ m/s and $AoA = 2$ degrees (clamped model)



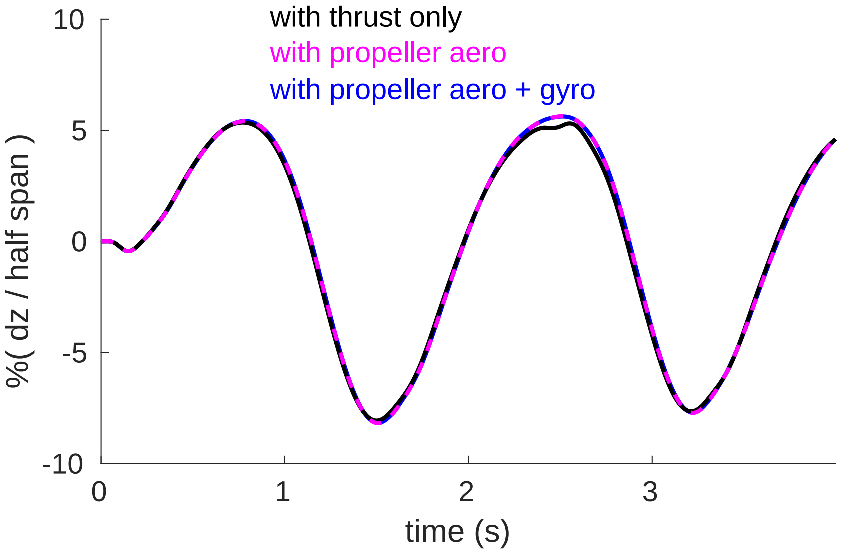
(a) Wing tip vertical displacement

(b) Wing tip twist

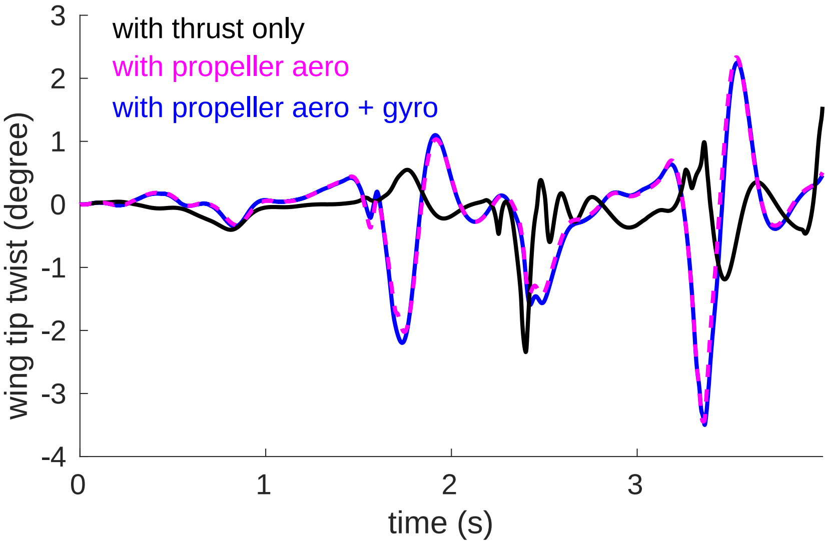
Figure 6.9: Response to a tail step excitation of amplitude 15 degrees at $v = 14$ m/s and $AoA = 2$ degrees (clamped model)

considered, but now including rigid body degrees of freedom: a continuous sinusoidal bending moment of amplitude 35 Nm and frequency of 0.59 Hz.

Figures 6.10 (a) and (b) show the wing tip displacement and twist response. As



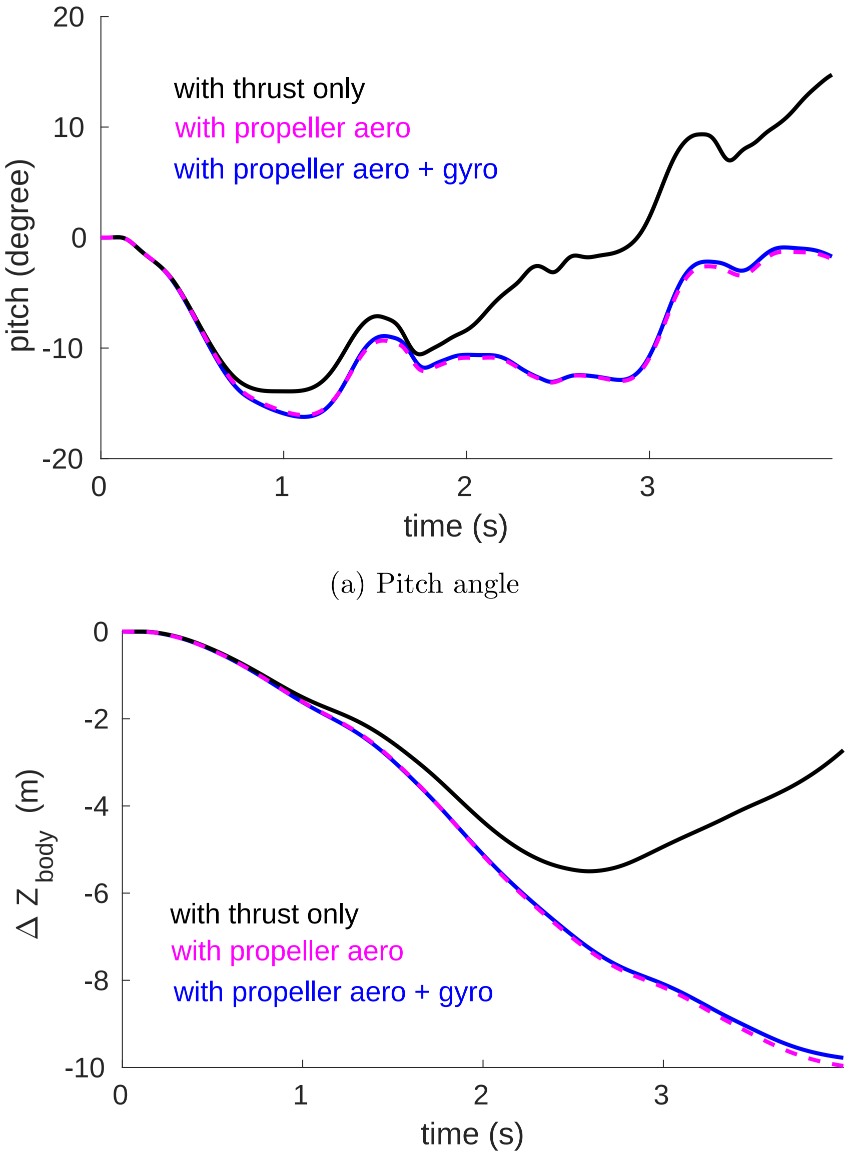
(a) Wing tip vertical displacement



(b) Wing tip twist

Figure 6.10: Wing tip response response to sinusoidal vertical tip bending moment excitation of amplitude 35 Nm and frequency 0.59 Hz, for free-flight X-HALE vehicle trimmed at $v = 14$ m/s.

in the clamped case, the vertical displacement presents negligible differences, but now more important differences are noticed for the wing twist, which presents also



(a) Pitch angle

(b) Vertical body displacement

Figure 6.11: Free-flight response to sinusoidal vertical tip bending moment excitation of amplitude 35 Nm and frequency 0.59 Hz, for X-HALE vehicle trimmed at $v = 14$ m/s.

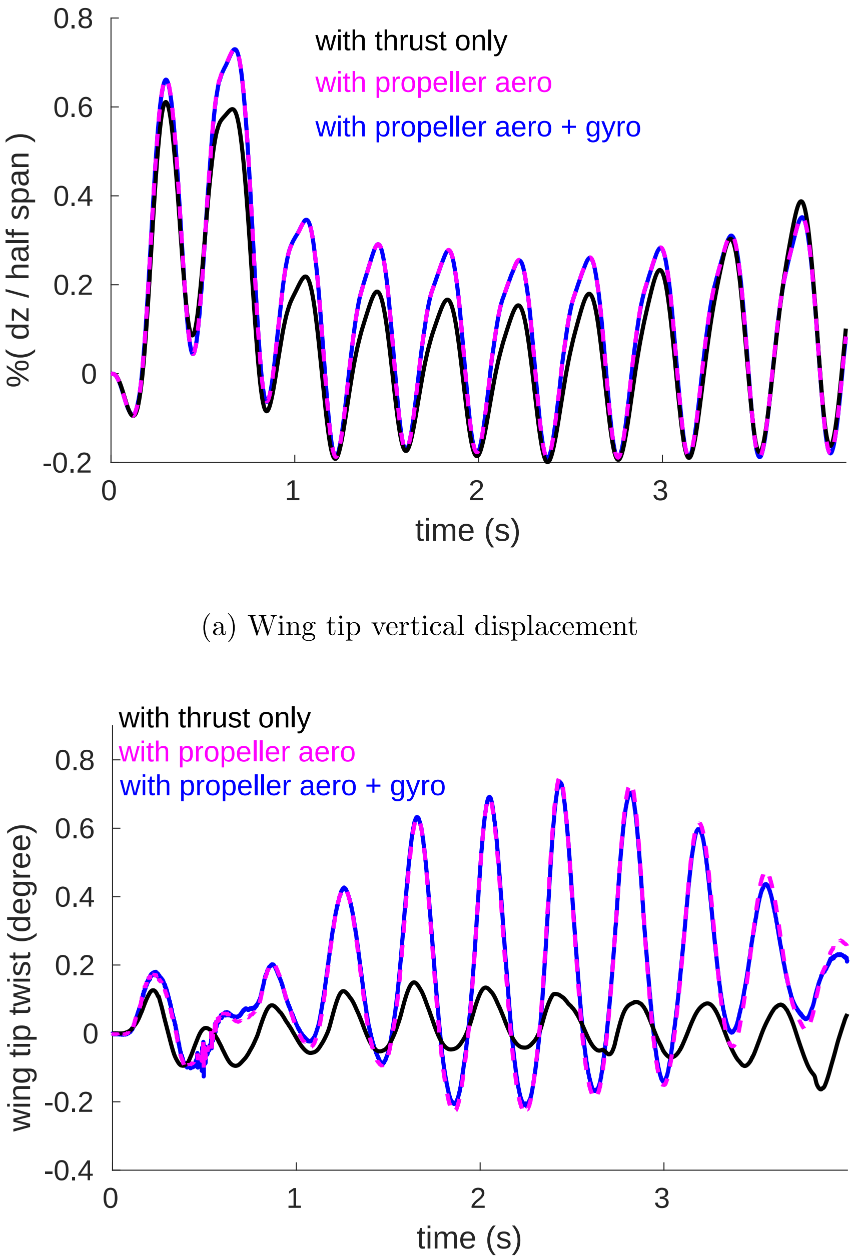
a richer frequency content. A remarkable difference can be seen for pitch angle and body vertical displacement, as shown in Figures 6.11 (a) and (b). The pitch angle shows a similar trend for cases with other propeller effects and with thrust only, but with visible differences in amplitude. Also, the aircraft is continuously reducing altitude for the case with other propeller effects, while it recovers altitude for the case with just thrust modeling. A possible explanation for those differences is the influence of the propeller slipstream on the aircraft surfaces as well as the inclusion of additional loads at the propeller hub (not only thrust but also moments and forces in the plane of the propeller). In general, the modeling of gyroscopic effects has a secondary influence. Due to the symmetric nature of the excitation to a vehicle with a small c.g. offset along the spanwise direction from its plane of symmetry, roll and pitch angles undergo very little change (under 2-degree variation) during the perturbation.

To evaluate the response of the free-flight model to torsional disturbance (still longitudinal), a continuous tail excitation with an amplitude of 15 degrees and a frequency of 2.6 Hz is used. From Figures 6.12 (a) and (b) one can observe much more pronounced relative differences for wing tip vertical and angular displacements in this case when compared to the bending excitation (Figures 6.11 (a) and (b)). The differences are also clear in pitch angle and rigid body vertical displacement, as shown in Figures 6.13 (a) and (b). The variation in pitch has a higher average value for cases with propellers, which show also smoother variations of body vertical displacement. Again, negligible differences for propeller models with or without gyroscopic moment can be observed.

6.3.3 Conclusions

From the dynamic cases investigated, it is concluded that:

- For the cases investigated, the most pronounced propeller effects were for the free-flight cases and for the clamped cases involving torsional excitation.
- For the X-HALE related cases where the light-weight (0.023 kg) plastic propellers are used, the gyroscopic moment presented a negligible influence in the cases investigated. However, the gyroscopic effects are anticipated to be more

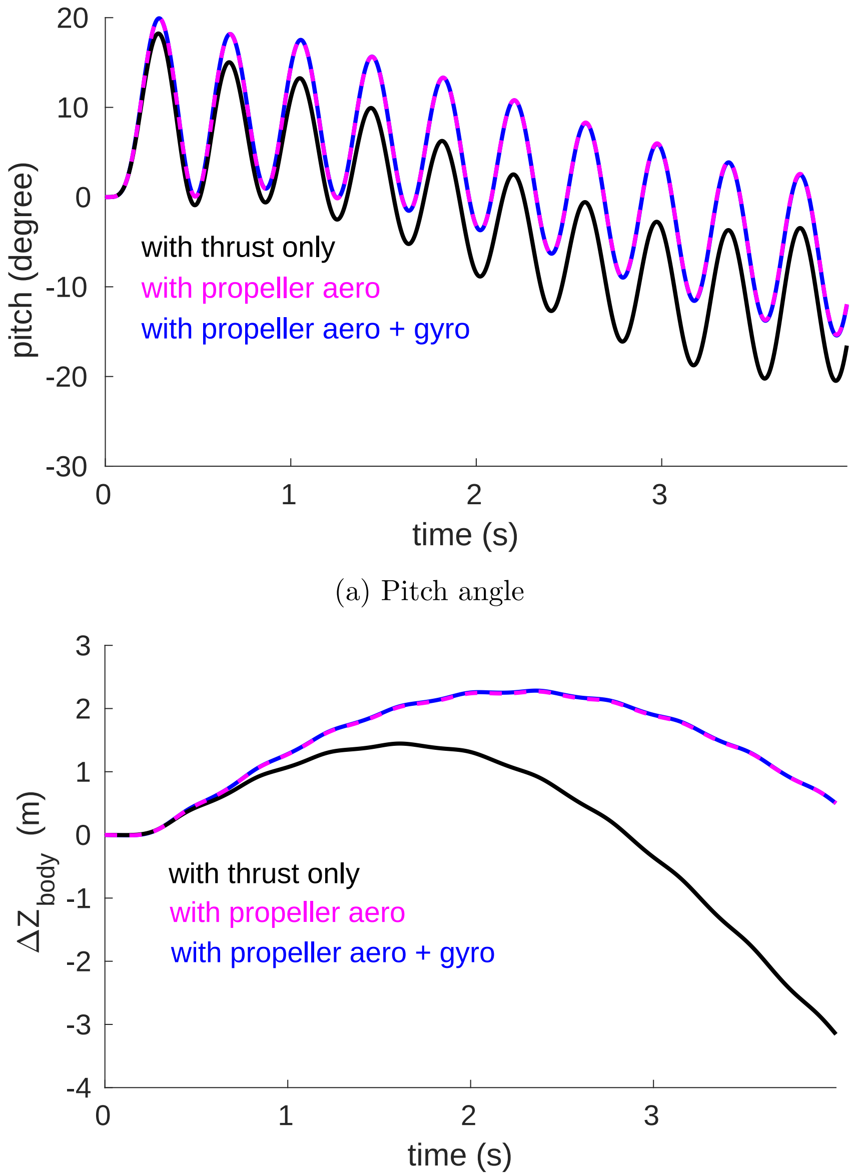


(a) Wing tip vertical displacement

(b) Wing tip twist

Figure 6.12: Wing tip response to a continuous sinusoidal tail excitation of 15 degrees and frequency 2.6 Hz, for free-flight X-HALE vehicle trimmed at $v = 14$ m/s.

important in larger vehicles and/or higher rotational speed.



(a) Pitch angle

(b) Vertical body displacement

Figure 6.13: Free-flight response to a continuous sinusoidal tail excitation of 15 degrees and frequency 2.6 Hz, for X-HALE vehicle trimmed at $v = 14$ m/s.

6.4 Propeller Effects on HALE Aircraft Stability

6.4.1 Propeller Complete Modeling vs. Thrust Only

In order to check the additional influence of the propeller slipstream and gyroscopic effects, besides the influence of thrust, transient solutions for the clamped X-HALE model with tip masses (described in Section 6.1.1) were conducted. Beginning from an equilibrium state and after 1.5 seconds of simulation, a step perturbation of 5 Nm in torsional moment and a 1 N step force in vertical and chordwise directions were applied. In this case, gravity and a stiffness-proportional damping coefficient of 0.005 s were considered to have a more realistic response. Three cases with different speeds: 12.5 m/s, 13 m/s and 13.5 m/s were simulated for a model with just thrust at the propeller location (equivalent to the thrust produced by the isolated propeller at each speed for the considered RPM) and a complete propeller model, including thrust (and other loads at hub), slipstream, and gyroscopic effects, with a rotation of 6000 RPM.

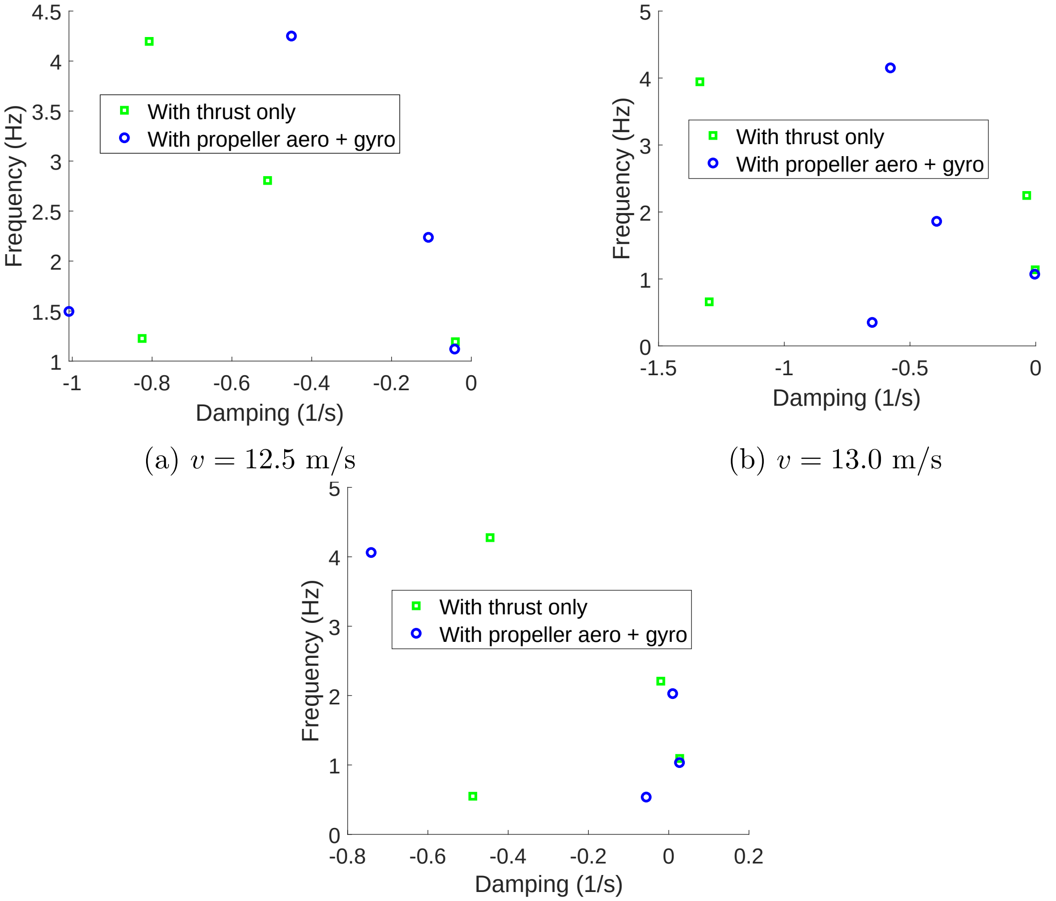
First, the same set of parameters determined in the verification case was tried (5 POD modes and Sys ID order of 18), but for the cases with aerodynamic and, for some of them, gyroscopic moment, the adjusting of the 5th POD coefficient using the same set of parameters for all cases was hard to get (especially for cases after the flutter boundary). Then, just 4 POD modes were considered (representing more than 99% of the snapshot energy), and an order of 18 was again included, improving the fitting of the first 4 POD coefficients. For the purely structural case the effect of reducing the POD modes to 4, keeping the same order for the Sys ID, was a reduction in the accuracy of higher frequencies. But the first four frequencies were kept with good accuracy (less than 2% error). Then, for the cases in this section, just four frequencies and modes are compared, corresponding to the most important modes for the system response. Also, the sensitivity of the results with the number of snapshots provided before the perturbation starts was found to be higher for cases with higher numerical perturbations, as in the cases after flutter. All cases use the same set of parameters and it is expected that the results can capture frequency variations due to the different effects being modeled.

Figures 6.14 and 6.15 present a comparison of the frequencies, dampings, and dynamic responses in terms of wing tip vertical and angular displacements for each simulated speed. From the dynamic responses, it is possible to see that for this model and propeller RPM the flutter boundary is about 13 m/s for both cases, *i. e.*, thrust only, and including complete propeller modeling. However, as one can also observe from the dynamic responses, increasing differences in phase arise between cases with thrust only and with all propeller effects for all speeds. For the same set of POD + Sys ID parameter choice, one can also note differences on frequencies and especially damping for modes farther away from the stability boundary. Considering that the variations of frequency and damping are captured, this indicates propellers can affect dynamic response.

6.4.2 Influence of Increasing Propeller RPM

In order to investigate the effect of propeller RPM in the aeroelastic stability, a transient solution similar to the one presented in Figure 6.15 was simulated for the clamped X-HALE model with tip masses at $v = 13$ m/s and with a higher propeller rotation of 7000 RPM. Increasing the propeller RPM with all other parameters constant means a higher influence on the velocity flowfield behind the propeller plane of rotation, as well as a higher thrust and gyroscopic loads. It can be interpreted as the degree of propeller effects intensity, which could be affected also by other parameters like propeller mass, geometry, velocity flowfield, etc.

Figure 6.16 presents a comparison of the wing tip dynamic responses. Now, more noticeable differences can be observed in the stability behavior; while the case with 6000 RPM is yet stable (although close to flutter), the case with 7000 RPM shows an unstable behavior with increasing amplitudes, then a lower flutter speed. This destabilizing effect with the increase in RPM is possibly related to the increase of dynamic pressure proportioned by the propeller slipstream, causing higher lift forces for the same local angle of attack. This example indicates that depending on the in-



(a) $v = 12.5$ m/s

(b) $v = 13.0$ m/s

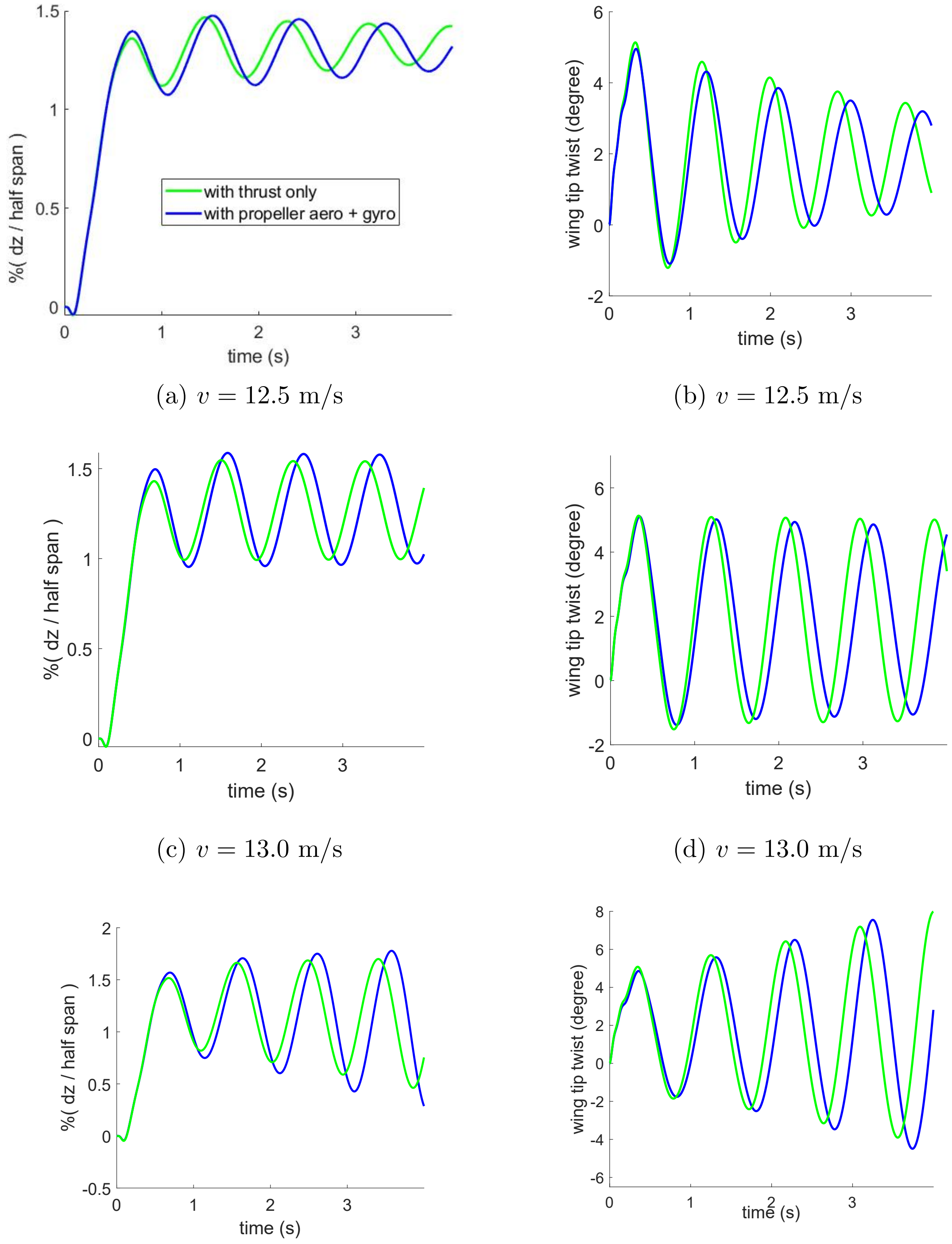
(c) $v = 13.5$ m/s

Figure 6.14: Frequencies and dampings at speeds of $v = 12.5$, $v = 13.0$ and $v = 13.5$ m/s extracted from response after perturbation with step loads of 5 N in vertical and chordwise directions and 1 Nm in torsion (6000 RPM)

tensity of propeller effects the influence on stability boundary can be more significant. Damping and frequency were not compared, as the same set of parameters used in $v = 13$ m/s for 6000 RPM does not provide a good fitting for the case of 7000 RPM.

6.4.3 Contribution of Different Propeller Effect Components

Figure 6.17 compares the dynamic response as well as frequency and damping for the same disturbance as in Figure 6.16 for the clamped X-HALE model with tip masses at $v = 13$ m/s. This case is already in the unstable regime. Due to more difficulties in find a common set of parameters with good fitting and in order to reduce the



(a) $v = 12.5$ m/s

(b) $v = 12.5$ m/s

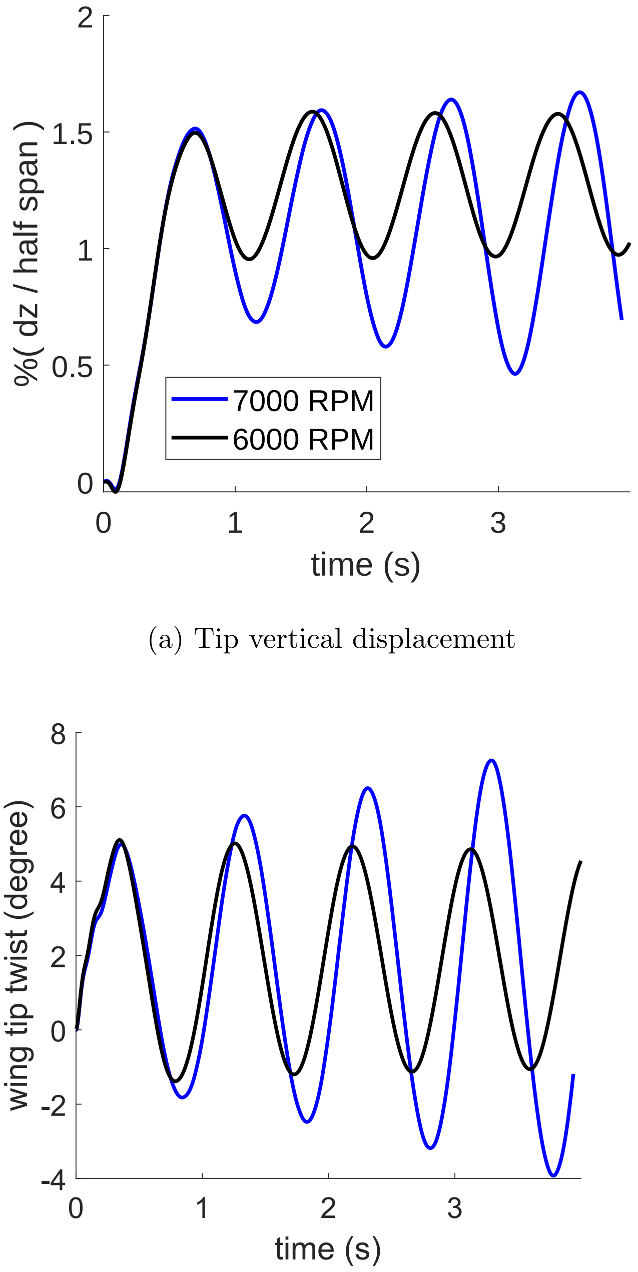
(c) $v = 13.0$ m/s

(d) $v = 13.0$ m/s

(e) $v = 13.5$ m/s

(f) $v = 13.5$ m/s

Figure 6.15: Response after perturbation with step loads of 5 N in vertical and chordwise direction and 1 Nm in torsion (6000 RPM), at speeds $v = 12.5$, $v = 13.0$ and $v = 13.5$ m/s

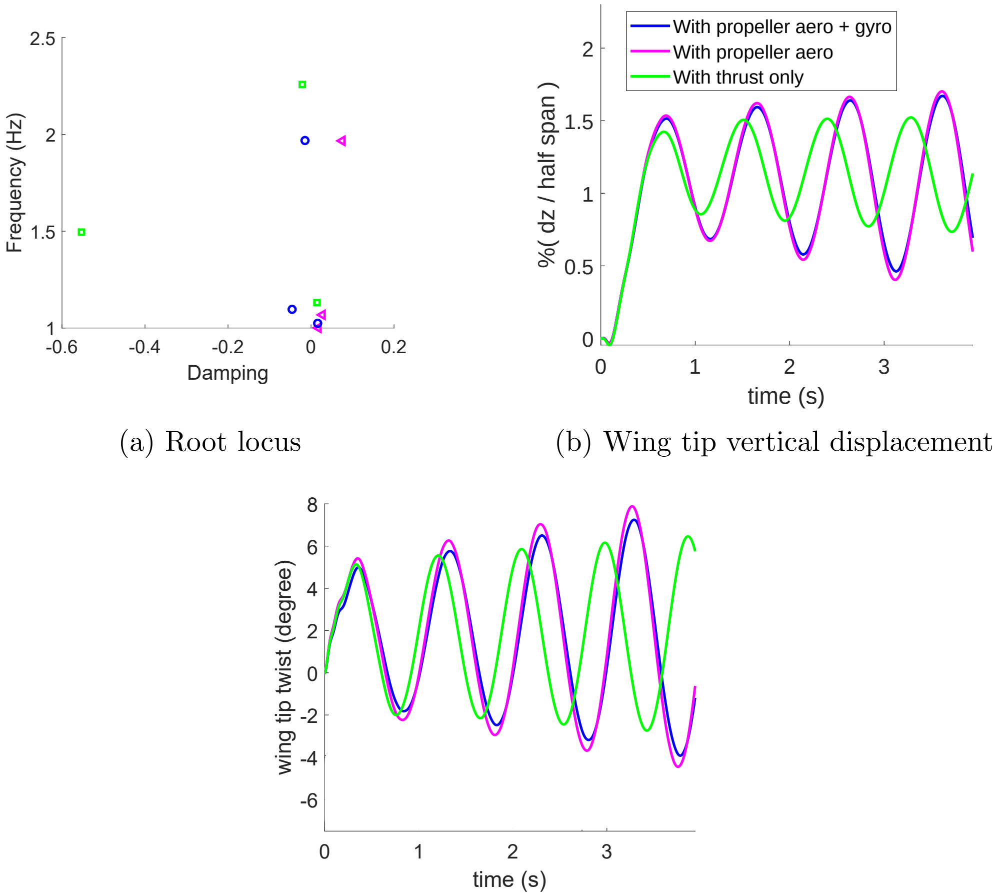


(a) Tip vertical displacement

(b) Tip twist

Figure 6.16: Wing tip response after perturbation with step loads of 5 N in vertical and chordwise direction and 1 Nm in torsion, including propeller aerodynamics and gyroscopic effects

perturbation due to numerical noise, just three POD modes were incorporated (representing more than 97% of the snapshot energy), keeping the identification order as



(a) Root locus

(b) Wing tip vertical displacement

(c) Wing tip twist

Figure 6.17: Frequencies, dampings, and response after perturbation with step loads of 5 N in vertical and chordwise direction and 1 Nm in torsion (7000 RPM)

18, and the snapshots were provided with 0.5 second after beginning of perturbations.

In order to observe the isolated propeller effect components (mainly thrust, slipstream and gyroscopic moment) three different modeling of propeller effects are considered:

i) with thrust only (equivalent to thrust produced by isolated propeller at $v = 13$ m/s and 7000 RPM);

ii) with thrust (and other loads at hub) and slipstream (propeller aero), and;

iii) with thrust (and other loads at hub), slipstream and gyroscopic effects (propeller aero + gyro).

From the dynamic response, it is clear that the inclusion of propeller aerodynamic effects can cause a non-negligible difference in the aeroelastic stability, with the model including just thrust having a smoother amplitude increase, suggesting a higher flutter boundary. As previously commented, this destabilizing effect may be related to the increased dynamic pressure due to the propeller slipstream, causing higher lift (and consequently higher structural deformations) for same freestream conditions. An increasing difference of phase with time is also noticeable between the case with just thrust and other two cases. The inclusion of gyroscopic effects causes negligible differences for vertical displacements but more visible differences in the amplitude of angular displacement, which is smaller for the case including gyroscopic effects. This is in accordance with the root locus, where it is clear that the positive damping for the unstable modes in the case with slipstream and no gyroscopic effects is higher, suggesting the gyroscopic effect has a stabilizing effect.

6.4.4 Conclusions

From the analysis of aeroelastic cases and the clamped model considered, it was observed that:

- The presence of propeller aerodynamic and gyroscopic effects influences the values of damping and frequencies of some modes and can influence stability boundary. Moreover, an increase in phase delay and differences in amplitude in the response to a perturbation close to flutter were shown as compared to the case with just thrust;

- A reduction of flutter boundary was found by an increase in propeller RPM. This destabilizing effect may be associated with the impact of slipstream, increasing the dynamic pressure and local lift;

- Also, a slight stabilizing effect due to the gyroscopic moment was noticed, suggesting this effect may be more important for cases with higher gyroscopic moment loads, *i.e.*, higher RPM and propeller mass.

CHAPTER 7

Conclusions and Recommendations

This chapter summarizes the main contributions and conclusions of the work presented in this dissertation. Ideas for future related research topics and improvements are also recommended.

7.1 Summary

High Altitude Long Endurance (HALE) aircraft is a new promising UAV concept, capable of long-duration flights and satellite-like applications. Its high-efficiency requirements result in a very flexible structure, with a strong coupling of aerodynamics, structure, and flight dynamics disciplines. For such low-speed vehicles, electric propellers are often the choice of propulsion.

Although many studies with rigid aircraft configurations have demonstrated that propellers can impose a significant effect on aircraft performance and stability, few investigations have been conducted on the effects of propellers on very flexible aircraft. Moreover, state-of-the-art coupled aeroelastic-flight dynamics frameworks for such very flexible configurations lack complete modeling of propeller effects, and just thrust force is usually modeled.

This work addressed those issues by enhancing the UM/NAST framework, a coupled nonlinear aeroelastic-flight dynamics framework developed at the University of Michigan [6, 31, 32], to take into account various propeller effects (all the resultant aerodynamic loads on the hub, the aerodynamic effects of propeller slipstream, and its inertial effects). The unsteady aerodynamics is based on Lifting Line plus Viscous Vortex Particle method for the propellers combined with an Unsteady Vortex Lattice for the lifting surfaces, allowing it to capture propwash effects as well as the interaction of the flow among multiple lifting surfaces. Discussion about the selection of each formulation for this context and its peculiarities are presented in Chapter 2. Also presented there are the derivation of the inertial effects associated with the rotating blades and the procedure for the integration of the different involved approaches.

The implementation of the new pieces is presented in details in Chapter 3. The aerodynamic developments were initially in Python/FORTRAN and then re-written in C++ (for performance improvement and to facilitate integration into UM/NAST). Libraries for the Unsteady Vortex Lattice, propeller solver based on Lifting Line and Viscous Vortex Particle, a general interface between a lifting surface and a propeller aerodynamics, and an interface class between UM/NAST and Vortex Lattice, as well as a function for the calculation of gyroscopic moment, were developed and are presented in Chapter 3. The Vortex Lattice and Propeller libraries can also be used for isolated aerodynamic analyses.

As discussed in more details in Chapter 4, incorporating the UVLM aerodynamics for the lifting surfaces and LL/VVPM formulation for the propellers in UM/NAST linearized formulation would require an in-depth review of the linearization formulation, which was originally done considering Peter's finite-state inflow theory. Instead, an alternative approach for extracting the dynamic information (frequency, damping, and modes) based on the time simulation signal was proposed. Such alternative is based on system identification and makes use of Proper Orthogonal Decomposition in order to reduce the degrees of freedom. In this way, it does not assume small displacements (as in the works employing linear modal analysis) and the methodology can be applied for a structure under large deformations. Moreover, it can be used for any time signal, like those from experimental results. Details about the method, its motivation, and a verification case are presented in Chapter 4.

Next, verification cases are presented for each new piece added to the UM/NAST aeroelastic framework in Chapter 5. Comparisons for static and dynamic aeroelastic analysis with the coupled UM/NAST and Vortex Lattice framework, isolated propeller (in terms of resultant loads on the hub and profile velocity of the slipstream), propeller-wing interaction and the gyroscopic loads calculation are presented against results from external codes or published numerical and experimental data available in literature.

After checking the newly added components, the enhanced coupled aeroelastic-flight dynamics framework is used for numerical investigations of the effects of propellers on a HALE aircraft, as presented and discussed in Chapter 6. Static and dynamic aeroelastic results for the University of Michigan's X-HALE vehicle [178, 179] are first investigated, and simulations considering complete and partial propeller modeling are compared, as well as rigid versus flexible cases. Next, the influence of propellers on aeroelastic stability of a VFA is studied, and the POD plus system identification method was used to extract the associated dynamic information.

From the numerical investigations, the main conclusions are:

i) Propellers can influence the static and dynamic behavior of a VFA. The differences were found to be more significant for free-flight and clamped cases with torsional motion and less significant for bending motion;

ii) They also were found to modify the aeroelastic modes and affect flutter onset, with important destabilizing role of propeller slipstream, due to the increase of the dynamic pressure;

iii) For X-HALE, where light-weight (0.023 kg) propellers are employed, gyroscopic moments presented a negligible influence on the transient response;

iv) However, slight stabilizing effect due to gyroscopic moment was observed, indicating this effect can be more important for configurations with heavier blades and/or higher RPM;

v) In summary, results demonstrated that complete propeller effects (beyond just thrust) should be taken into account in the VFA simulation.

Finally, it is worth to mention that, although the framework with propellers was developed focusing on very flexible configurations, the enhanced framework is also capable of being used for the analysis of new propeller-driven aircraft, as Urban Air Mobility (UAM) [184, 185] and distributed propulsion (DP) [22–25, 27] concepts.

7.2 Key Contributions

The key contributions of this dissertation can be summarized as follow:

- Development and verification of a coupled nonlinear aeroelastic-flight dynamics framework capable of complete propeller effects modeling. Such effects are not included in other state-of-the-art coupled nonlinear aeroelastic-flight mechanics codes. This goal was achieved by enhancing the UM/NAST aeroelastic framework, which lead to the following additional contributions:

  i) Selection and implementation of a C++ Vortex Lattice library based on an existent Python/FORTRAN code developed by Ritter [81] for the lifting surfaces aerodynamics. This Vortex Lattice aerodynamic option allows taking into account mutual influences among the VFA lifting surfaces as well as among surfaces and propellers. Additionally, the code can be used coupled to UM/NAST framework of for isolated aerodynamic analysis;

  ii) Development and implementation of a C++ interface between UM/NAST and the Vortex Lattice code allowing the treatment of each code specific variables, as well as the transfer of loads and displacements between the UM/NAST beam structural model and the non-planar Vortex Lattice grid. The integrated UM/NAST plus Vortex Lattice framework was verified against three different external nonlinear aeroelastic codes also employing Vortex Lattice for the aerodynamics, for a range of static, trim and dynamic simulations. Excellent agreement was obtained, also providing benchmark cases for future comparisons;

iii) Implementation (first in Python/FORTRAN, then in C++) of a propeller aerodynamic solver based on selected approaches for the context of VFA: Lifting Line method for the propeller blades and Viscous Vortex Particle in order to model the propeller wakes. Such approaches result in a mid-fidelity tool, which demonstrated good agreement for comparisons against experimental and numerical results in terms of resultant loads on the blades and profile velocity behind the propeller. As in the case of Vortex Lattice code, it can be used coupled to UM/NAST framework or for isolated aerodynamic analysis (with one or multiple propellers);

iv) Development of a C++ general interface between propellers and airframe aerodynamics: this interface was designed to allow for different combinations of propellers and airframe aerodynamic solvers. For the integration with UM/NAST framework, a combination of Vortex Lattice for lifting surfaces and Lifting Line + Viscous Vortex Particle for the propellers was used. Comparison for a rigid propeller-wing combination was performed against published numerical results, and excellent agreement was observed;

v) Derivation of the propeller inertial loads and implementation of a C++ function to calculate the missing gyroscopic moments. The derivation took into account UM/NAST frames and defined new frames for the propeller and blades. Results for an analytical case of a rigid pitching wing with a propeller on its middle were compared against an analytical example, and excellent agreement was observed;

- Development of an approach, based on a combination of System Identification and Proper Orthogonal decomposition, for the stability analysis of VFA with propellers based on a time-series signal. The POD was employed in order to reduce the number of degrees of freedom, and then the size of the output matrix provided to the system identification. Results for a purely structural case (a reference that was available) were compared against UM/NAST results. A maximum error in frequency prediction smaller than 2% and a MAC value of about 0.99 were observed for all the modes compared.

- Study of the propeller effects on the aeroelastic response and stability of a representative HALE aircraft, the University of Michigan's X-HALE vehicle, showing the importance of incorporating propeller effects other than thrust for the adequate assessment of aeroelastic response and stability of Very Flexible Aircraft. For the cases investigated, aerodynamic effects (in terms of resultant loads on the blades and propeller wake) have demonstrated to play a major role when compared to the gyroscopic moments. However, a slight stabilizing effect was identified due to the gyroscopic moment and can potentially be of more relevance for models with higher gyroscopic loads (*e.g.*, a combination of higher diameter, RPM and/or heavier propellers);

7.3 Recommendations for Future Work

During the course of this work, additional topics have been identified for future studies:

- Acceleration of VVPM and VLM/uVLM methods: as discussed in Chapter 2, both, viscous vortex particle and Vortex Lattice methods are classified as N-body problems, with N particles (or panels) influencing each other, such that the calculation of their mutual influence has a computational cost of $O(N^2)$. Then, N-body acceleration algorithms as TreeCode method [147, 148] and the Fast Multipole Method (FMM) [149, 150] could be applied. A recent work of Kebbie-Anthony *et al.* [186] has used for the first time the FMM method to a nonlinear unsteady aerodynamic simulator based on Vortex Lattice method. Another option can be the development of reduced-order models (ROM), as physics-based ROMs based on POD modes;

- Another interesting investigation can be the comparison of simplified propeller aerodynamic methods for different aeroelastic analysis (static, dynamic response, stability), evaluating how they compare in terms of cost and accuracy;

- Using the developed enhanced framework (possibly accelerated as described above), investigate propeller effects for new propeller-driven configurations as Urban Mobility's air taxi concepts;

- Comparison of the numerical aeroelastic results with propellers against experimental data for a complete HALE aircraft or a clamped wing with propeller;

- Extend formulation of gyroscopic loads for variable propeller rotation. Equation 2.85, in Chapter 2 was developed with no assumption about the propeller angular velocity. However, this assumption was made in the equations 2.86 and 2.87, considering then that the aircraft is flying with a constant angular velocity and simplifying the calculation and implementation of such derivatives. No assumption of angular velocity is made for the propeller aerodynamics though, and if gyroscopic effects are not included, simulations with varying RPM can be conducted with the current framework;

- Extend framework applications by modeling blade flexibility.

BIBLIOGRAPHY

[1] del Carre, A., Teixeira, P. C., Palacios, R., and Cesnik, C. E. S., "Nonlinear Response of a Very Flexible Aircraft Under Lateral Gust," *International Forum on Aeroelasticity and Structural Dynamics, IFASD*, Savannah, Georgia, USA, Jun 2019.

[2] Sundar, R. M., *An Experimental Investigation of Propeller Wakes Using a Laser Doppler Velocimeter*, Ph.D. thesis, Purdue University, West Lafayette, IN, 1985.

[3] Jones, J. R. and Cesnik, C. E. S., "Preliminary Flight Test Correlations of the X-HALE Aeroelastic Experiment," *The Aeronautical Journal*, Vol. 119, No. 1217, 2015, pp. 855–870.

[4] Ritter, M., Jones, J., and Cesnik, C. E. S., "Enhanced Modal Approach for Free-flight Nonlinear Aeroelastic Simulation of Very Flexible Aircraft," *AIAA Science and Technology Forum and Exposition (SciTech2016), 15th Dynamics Specialists Conference*, AIAA Paper 2016-1794, January 2016.

[5] Hodges, D. H., Patil, M. J., and Chae, S., "Effect of Thrust on Bending-Torsion Flutter of Wings," *Journal of Aircraft*, Vol. 39, No. 2, 2002, pp. 371–376.

[6] Shearer, C. M. and Cesnik, C. E. S., "Nonlinear Flight Dynamics of Very Flexible Aircraft," *Journal of Aircraft*, Vol. 44, No. 5, 2007, pp. 1528–1545.

[7] Changchuan, X., Lan, Y., Yi, L., and Chao, Y., "Stability of Very Flexible Aircraft with Coupled Nonlinear Aeroelasticity and Flight Dynamics," *Journal of Aircraft*, Vol. 55, No. 2, oct 2017, pp. 862–874.

[8] Prandtl, L., "Mutual Influence of Wings and Propeller," Tech. rep., NACA TN-74, December 1921.

[9] Witkowski, D. P., Lee, A. K. H., and P., S. J., "Aerodynamic Interaction Between Propellers and Wings," *Journal of Aircraft*, Vol. 26, No. 9, 1989, pp. 829–836.

[10] Miranda, L. R. and Brennan, J. E., "Aerodynamic Effects of Wing Tip-mounted Propellers and Turbines," *4th Applied Aerodynamics Conference, Fluid Dynamics and Co-located Conferences*, 1986, San Diego, California.

[11] Kroo, I., “Propeller-Wing Integration for Minimum Induced Loss,” *Journal of Aircraft*, Vol. 23, No. 7, 1986, pp. 561–565.

[12] Marreta, R. M. A., “Different Wings Flowfields Interaction on the Wing-Propeller Coupling,” *Journal of Aircraft*, Vol. 34, No. 6, 1997, pp. 740–747.

[13] Veldhuis, L., *Propeller Wing Aerodynamic Interference*, Ph.D. thesis, Delft University of Technology, Delft, 2005.

[14] Ananda, G. K., Deters, R. W., and Selig, M. S., “Propeller-Induced Flow Effects on Wings of Varying Aspect Ratio at Low Reynolds Numbers,” *32nd AIAA Applied Aerodynamics Conference*, 2014, Atlanta, Georgia, 16 - 20 June.

[15] Mishra, A., Davoudi, B., and Duraisamy, K., “Multiple Fidelity Modeling of Interactional Aerodynamics,” *35th AIAA Applied Aerodynamics Conference*, 2017, 5-9 June 2017, Denver, Colorado.

[16] Calabretta, J. S., *A Three Dimensional Vortex Particle-panel Code for Modeling Propeller-airframe Interaction*, Master’s thesis, California Polytechnic State University, San Luis Obispo, 2010.

[17] Singh, P. and Friedmann, P. P., “Application of Vortex Methods to Coaxial Rotor Wake and Load Calculations in Hover,” *Journal of Aircraft*, Vol. 55, No. 1, 2017, pp. 373–381.

[18] Singh, P. and Friedmann, P. P., “Dynamic Stall modeling using Viscous Vortex Particle Method for Coaxial Rotors,” *Vertical Flight Society 75th Annual Forum and Technology Display, Philadelphia, Pennsylvania*, May 1316, 2019.

[19] Thepvongs, S., Cesnik, C. E. S., and Voutsinas, S. G., “Aeroelastic and Acoustic Analysis for Active Twist Rotors,” *31st European Rotorcraft Forum*, 2005, Florence, Italy, 13-15 September 2005.

[20] He, C. and Zhao, J., “Modeling Rotor Wake Dynamics with Viscous Vortex Particle Method,” *AIAA Journal*, Vol. 47, No. 4, 2009, pp. 902–915.

[21] Willis, D. J., *An Unsteady, Accelerated, High Order Panel Method with Vortex Particle Wakes*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, 2006.

[22] Borer, N. K., Moore, M. D., and Turnbull, A. R., “Tradespace Exploration of Distributed Propulsors for Advanced On-Demand Mobility Concepts,” *The Aviation and Aeronautics Forum and Exposition*, 2014, Atlanta, Georgia, 16-20 June.

[23] Borer, N. K. and Moore, M. D., “Integrated Propeller-Wing Design Exploration for Distributed Propulsion Concepts,” *53rd AIAA Aerospace Sciences Meeting*, 2015, Kissimmee, Florida, 5-9 January.

[24] Ortun, B., "A Coupled RANS/lifting-line Analysis for Modeling the Aerodynamics of Distributed Propulsion," *AHS Technical Conference on Aeromechanics*, 2018, San Francisco, California, 16-19 January.

[25] Fischer, J. and Ortun, B., "Simulation and Analysis of the Aerodynamic Interaction between Distributed Propulsion and Wings," *AHS Technical Conference on Aeromechanics*, 2018, San Francisco, California, 16-19 January.

[26] Escobar, M. T. and Moreno-Jiménez, J. M., "The Hierarchical compromise programming," *Top*, Vol. 5, No. 2, Dec 1997, pp. 253–281.

[27] Alvarez, E. J. and Ning, A., "Development of a Vortex Particle Code for the Modeling of a Wake Interaction in Distributed Propulsion," *AIAA Aviation Forum*, June 25-29, 2018, Atlanta, Georgia.

[28] Jamison, G. R., *Flight Test Investigation of Propeller Effects on the Static Longitudinal Stability of the E-2C Airplane*, Master's thesis, The University of Tennessee, Knoxville, 2006.

[29] Chen, G., Chen, B., Li, P., Bai, P., and Ji, C., "Numerical Simulation Study on Propeller Slipstream Interference of High Altitude Long Endurance Unmanned Air Vehicle," *2014 Asia-Pacific International Symposium on Aerospace Technology*, Shanghai, China, September 24-26, 2015.

[30] He, C. and Rajmohan, N., "Modeling the Aerodynamic Interaction of Multiple Rotor Vehicles and Compound Rotorcraft with Viscous Vortex Particle Method," *American Helicopter Society 72nd Annual Forum*, 2016, West Palm Beach, Florida, 17-19 May.

[31] Su, W. and Cesnik, C. E. S., "Dynamic Response of Highly Flexible Flying Wings," *AIAA Journal*, Vol. 49, No. 2, 2011, pp. 324–339.

[32] Su, W. and Cesnik, C. E. S., "Nonlinear Aeroelasticity of a Very Flexible Blended-Wing-Body Aircraft," *Journal of Aircraft*, Vol. 47, No. 5, 2010, pp. 1539–1553.

[33] Gamble, B. and Reeder, M., "Experimental Analysis of Propeller Interactions with a Flexible Wing Micro Air Vehicle," *36th AIAA Fluid Dynamics Conference and Exhibit, Fluid Dynamics and Co-located Conferences, AIAA Paper 2006-3916, 2006*, San Francisco, California, 05-08 June.

[34] Agostinelli, C., Liu, C., Allen, C. B., Rampuravawala, A., and Ferraro, G., "Propeller-Flexible Wing Interaction Using Rapid Computational Methods," *31st AIAA Applied Aerodynamics Conference, Fluid Dynamics and Co-located Conferences*, AIAA Paper 2013-2418, June 2013.

[35] Cravana, A., Manfreda, G., Cestino, E., Frulla, G., Carrese, R., and Marzocca, P., "Aeroelastic Behaviour of Flexible Wings Carrying Distributed Electric Propulsion Systems," *AeroTech Congress & Exhibition*, SAE Technical Paper 2017-01-2061, September 2017.

[36] Lanchester, F. W., *The Flying Machine*, Wiley, New York, 1917.

[37] Harris, R. G., "Forces on a Propeller Due to Sideslip," Tech. rep., Aeronautical Research Committee Reports and Memoranda No. 427, London, 1918.

[38] Glauert, H., "The Stability Derivatives of an Airscrew," Tech. rep., Aeronautical Research Committee Reports and Memoranda No. 642, London, 1919.

[39] Glauert, H., "Airplane propellers," Aerodynamics Theory, Miscellaneous Airscrew Problems, Vol. 4 div. L, Chapter XII, Secs. 5-6, Berlin, pp. 351-359., 1935.

[40] Katzoff, S., "Longitudinal Stability and Control with Special Reference to Slipstream Effects," Tech. rep., National Advisory Committee for Aeronautics, 1940.

[41] Ribner, H. S., "Propellers in Yaw," Tech. rep., National Advisory Committee for Aeronautics, 1945.

[42] Goraj, Z. J. and Cichocka, E., "Influence of Weak and Strong Gyroscopic Effects on Light Aircraft Dynamics," *Aircraft Engineering and Aerospace Technology*, Vol. 88, No. 5, 2016, pp. 613–622.

[43] Bouquet, T., *Modelling the Propeller Slipstream Effect on the Longitudinal Stability and Control*, Master's thesis, Delft University of Technology, 2016.

[44] Feldt, W. T. and Herrmann, G., "Bending-Torsional Flutter of a Cantilevered Wing Containing a Tip Mass and Subject to a Transverse Follower Force," *Journal of the Franklin Institute*, 1974, pp. 467–468.

[45] Quanlong, C., Jinglong, H., and Haiwei, "Effect of Thrust Engine on Nonlinear Flutter of Wings," *Journal of Vibroengineering*, 12 2013, pp. 1731–1739.

[46] Rezaeian, A., "Dynamic Stability Analysis of a Propeller-Wing Wind Tunnel Model," *Deutscher Luft- und Raumfahrtkongress*, 2011, Bremen, Deutschland.

[47] An, S., *Aeroelastic Design of a Lightweight Distributed Electric Propulsion Aircraft with Flutter and Strength Requirements*, Master's thesis, Georgia Institute of Technology, Atlanta, Georgia, 2015.

[48] Guruswamy, G. P., "Dynamic Aeroelasticity of Wings with Tip Propeller by Using NavierStokes Equations," *AIAA Journal*, Vol. 57, No. 8, 2019, pp. 3200–3205.

[49] Gur, O. and Rosen, A., "Comparison Between Blade-Element Models of Propellers," *The Aeronautical Journal*, Vol. 112, No. 1138, 2008, pp. 689–704.

[50] Abedi, H., Davidson, L., and Voutsinas, S., "Vortex Method Application for Aerodynamic Loads on Rotor Blades," *EWEA 2013: Europe's Premier Wind Energy Event*, 2013, Vienna, Austria, 4-7 February 2013.

[51] Khan, W. and Nahon, M., "Development and Validation of a Propeller Slipstream Model for Unmanned Aerial Vehicles," *Journal of Aircraft*, Vol. 52, No. 6, 2015, pp. 1985–1994.

[52] Wang, Q., Jiang, Z., and Q., Z., "Regionalized Actuador Disk Model Designed by Optimization Method for Propeller Slipstream Computation," *Engineering Applications of Computational Fluid Mechanics*.

[53] Horlock, J. H., *Actuador Disk Theory*, McGraw-Hill, 1978.

[54] Smelt, B. A. and Davies, H., *Estimation of Increase in Lift due to Slipstream*, Aeronautical Research Council, Royal Aircraft Establishment, 1937, RM No. 1788, London, UK.

[55] Kuhn, R. E., "Semiempirical Procedure for Estimating Lift and Drag Characteristics of Propeller-Wing-Flap Configurations for Vertical- and Short-Take-Off-and-Landing Airplanes," Tech. rep., NASA Memo 1-16-59L, 1959.

[56] Drzewiecki, S., *Théorie Générale de L'hélice*, Gauthier-Villars et cie, Paris, 1920.

[57] Katz, J. and Plotkin, A., *Low-Speed Aerodynamics*, 2nd Edition, Cambridge Aerospace Series, Cambridge University Press, New York, 2001.

[58] Anderson, J. D., *Fundamentals of Aerodynamics*, 3rd Edition, Aeronautical and Aerospace Engineering Series, McGraw-Hill, 2001.

[59] Clark, D. R. and Leiper, A. C., "The Free Wake Analysis A Method For The Prediction Of Helicopter Rotor Hovering Performance." 1969.

[60] McCormick, B. W., *Aerodynamics of V/STOL Flight*, Academic Press, New York, 1967.

[61] Theodorsen, T., *Theory of Propellers*, McGraw-Hill, 1948.

[62] Riso, C., Di Vicenzo, F. G., Ritter, M., Cesnik, C. E. S., and Mastroddi, F., "A FEM-Based Approach for Nonlinear Aeroelastic Trim of Highly Flexible Aircraft," *International Forum on Aeroelasticity and Structural Dynamics, IFASD 2017*, Como, Italy, 25-28 June 2017.

[63] Murua, J., Palacios, R., and Graham, J. M. R., "Applications of the Unsteady Vortex-Lattice Method in Aircraft Aeroelasticity and Flight Dynamics," *Progress in Aerospace Sciences*, Vol. 55, 2012, pp. 46 – 72.

[64] Simpson, R. J. and Palacios, R., "Numerical Aspects of Nonlinear Flexible Aircraft Flight Dynamics Modeling," *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, apr 2013.

[65] Hesse, H., Palacios, R., and Murua, J., "Consistent Structural Linearization in Flexible Aircraft Dynamics with Large Rigid-Body Motion," *AIAA Journal*, Vol. 52, No. 3, mar 2014, pp. 528–538.

[66] Del Carre, A. and Palacios, R., "Low-Altitude Dynamics of Very Flexible Aircraft," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, San Diego, California, 2019.

[67] "ASWING - MIT," `http://web.mit.edu/drela/Public/web/aswing/`, Accessed: 2019-08-05.

[68] "ASWING 5.97 User Guide," `http://web.mit.edu/drela/Public/web/aswing/aswing_doc.txt`, Accessed: 2019-08-05.

[69] "ASWING 5.99 Technical Description," `https://pdfs.semanticscholar.org/c919/59501949e68b0fd1005b097e1e4db2952eb0.pdf`, Accessed: 2019-08-05.

[70] Chang, C.-S., *Vibration and Aeroelastic Analysis of Highly Flexible HALE Aircraft*, Ph.D. thesis, Georgia Institute of Technology, USA, 2006.

[71] Mardanpour, P. and Hodges, D. H., "On the Importance of Nonlinear Aeroelasticity and Energy Efficiency in Design of Flying Wing Aircraft," *Advances in Aerospace Engineering*, Vol. 2015, 2015.

[72] Ritter, M., Cesnik, C. E. S., and Kruger, W. R., "An Enhanced Modal Approach for Large Deformation Modeling of Wing-Like Structures," *AIAA Science and Technology Forum and Exposition (SciTech2015), 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA Paper 2015-0176, January 2015.

[73] Teixeira, P. C. and Cesnik, C. E. S., "Inclusion of Propeller Effects on Aeroelastic Behavior of Very Flexible Aircraft," *International Forum on Aeroelasticity and Structural Dynamics, IFASD 2017*, Como, Italy, 25-28 June 2017.

[74] Teixeira, P. C. and Cesnik, C. E. S., "Propeller Effects on the Dynamic Response of HALE Aircraft," *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, 2018, Kissimmee, Florida, 812 January.

[75] Teixeira, P. C. and Cesnik, C. E. S., "Propeller Effects on the Response of High-Altitude Long-Endurance Aircraft," *AIAA Journal*, Vol. 0, No. 0, 0, pp. 1–15.

[76] Su, W., *Coupled Nonlinear Aeroelasticity and Flight Dynamics of Fully Flexible Aircraft*, Ph.D. thesis, University of Michigan, Ann Arbor, MI, 2008.

[77] Su, W. and Cesnik, C. E. S., "Strain-based Geometrically Nonlinear Beam Formulation for Modeling Very Flexible Aircraft," *International Journal of Solids and Structures*, Vol. 48, No. 16-17, 2011, pp. 2349–2360.

[78] Cesnik, C. E. S. and Brown, E. L., "Active Warping Control of a Joined Wing/-Tail Airplane Configuration," *44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*.

[79] Peters, D. A. and Johnson, M. J., "Finite-State Airloads for Deformable Airfoils on Fixed and Rotating Wings," *American Society of Mechanical Engineers*, Vol. 44, 1994, pp. 1–28.

[80] Peters, D. A. and Cao, W., "Finite State Induced Flow Models, Part 1: Two-Dimensional Thin Airfoil," *Journal of Aircraft*, Vol. 32, No. 2, 1995, pp. 313–322.

[81] Ritter, M. R., *An Extended Modal Approach for Nonlinear Aeroelastic Simulations of Highly Flexible Aircraft Structures*, Ph.D. thesis, Technischen Universitt Berlin, Berlin, 2018.

[82] Ritter, M., Teixeira, P. C., and Cesnik, C. E. S., "Comparison of Nonlinear Aeroelastic Methods for Maneuver Simulation of Very Flexible Aircraft," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, AIAA Paper 2018-1953, January 2018.

[83] Kima, D., Leeb, J.-S., Leeb, J.-H., and Hanb, J., "An Aeroelastic Analysis of a Flexible Flapping Wing Using Modified Strip Theory," *SPIE 15th Annual Symposium Smart Structures and Materials*, 2008.

[84] Patil, M. J., Hodges, D. H., and Cesnik, C. E. S., "Nonlinear Aeroelastic Analysis of Complete Aircraft in Subsonic Flow," *Journal of Aircraft*, Vol. 37, No. 5, 2000, pp. 753–760.

[85] Georgiou, G., Vio, G. A., and Cooper, J. E., "Aeroelastic Tailoring and Scaling Using Bacterial Foraging Optimisation," *Structural and Multidisciplinary Optimization*, Vol. 50, No. 1, Jul 2014, pp. 81–99.

[86] Weisshaar, T. A., "Aeroelastic Tailoring of Forward Swept Composite Wings," *Journal of Aircraft*, Vol. 18, No. 8, 1981, pp. 669–676.

[87] Huang, Y. and Su, W., "Linearization and Analytical Aerodynamic Sensitivity of Unsteady Vortex-Lattice Aerodynamics," *AIAA Scitech 2019 Forum*, 2019.

[88] Chao, Y., LiBo, W., ChangChuan, X., and Yi, L., "Aeroelastic Trim and Flight Loads Analysis of Flexible Aircraft with Large Deformations," *Science China Technological Sciences*, Vol. 55, No. 10, 2012, pp. 2700 – 2711.

[89] Xie, C., Wang, L., Yang, C., and Liu, Y., "Static Aeroelastic Analysis of Very Flexible Wings Based on Non-planar Vortex Lattice Method," *Chinese Journal of Aeronautics*, Vol. 26, No. 3, 2013, pp. 514 – 521.

[90] Guimaraes Neto, A. B. G., "Approximation of Aerodynamic Geometrical Nonlinearities in Aircraft with High-aspect-ratio Wings," *International Forum on Aeroelasticity and Structural Dynamics, IFASD*, Savannah, Georgia, USA, Jun 2019.

[91] Hesse, H. and Palacios, R., "Reduced-Order Aeroelastic Models for Dynamics of Maneuvering Flexible Aircraft," *AIAA Journal*, Vol. 52, No. 8, 2014, pp. 1717–1732.

[92] Nguyen, N., Reynolds, K., Trinh, K., and Frost, S., "Coupled Aeroelastic Vortex Lattice Modeling of Flexible Aircraft," *29th AIAA Applied Aerodynamics Conference*, 2012.

[93] James, R. M., "On the Remarkable Accuracy of the Vortex Lattice Method," *Computer Methods in Applied Mechanics*, Vol. 1, No. 1, 1972, pp. 59–79.

[94] Ritter, M., Dillinger, J., and Meddaikar, Y. M., "Static and Dynamic Aeroelastic Validation of a Flexible Forward Swept Composite Wing," *AIAA SciTech, 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Grapevine, Texas, Jan 2017.

[95] Falkner, V. M., "The Scope and Accuracy of Vortex Lattice Theory," Tech. rep., Reports and Memoranda 2740, London, 1949.

[96] Rosenhead, L., "Formation of Vortices from a Surface of Discontinuity," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 134, 170 - 192*, 1931.

[97] Falkner, V. M., *The Calculation of Aerodynamic Loading on Surfaces of Any Shape*, A.R.C. technical report, H.M. Stationery Office, 1943.

[98] Hedman, S. G., "Vortex Lattice Method for Calculation of Quasi Steady State Loadings on Thin Elastic Wings in Subsonic Flow," Tech. rep., Research Institute of Sweden, Report 105, 1965.

[99] Albano, E. and Rodden, W. P., "A Doublet-Lattice Method for Calculating Lift Distributions on Oscillating Surfaces in Subsonic Flows." *AIAA Journal*, Vol. 7, No. 2, 1969, pp. 279–285.

[100] Hess, J. L., "Calculation of Potential Flow about Arbitrary Three-dimensional Liftings Bodies," Tech. rep., Final Technical Report MDC J5679-01, Douglas Aircraft Co., Long Beach, CA, USA, October 1972.

[101] Belotserkovskii, S. M., "Study of the Unsteady Aerodynamics of Lifting Surfaces Using the Computer," *Annual Review of Fluid Mechanics*, Vol. 9, No. 1, 1977, pp. 469–494.

[102] Rehbach, C., "Numerical Calculation of Three-Dimensional Unsteady Flows with Vortex Sheets," *La Recherche Aerospatiale, Sept.-Oct. 1977, p. 289-298. In French.*, Oct. 1977, pp. 289–298.

[103] Atta, E. H., Kandil, O. A., MOOK, D. T., and Nayfeh, A. H., "Unsteady Aerodynamic Loads on Arbitrary Wings Including Wing-tip and Leading-edge Separation," *AIAA*, 1977.

[104] Konstadinopoulos, P., Thrasher, D. F., Mook, D. T., Nayfeh, A. H., and Watson, L., "A Vortex-Lattice Method for General, Unsteady Aerodynamics," *Journal of Aircraft*, Vol. 22, No. 1, 1985, pp. 43–49.

[105] Levin, D. and Katz, J., "Vortex-Lattice Method for the Calculation of the Nonsteady Separated Flow over Delta Wings," *Journal of Aircraft*, Vol. 18, No. 12, 1981, pp. 1032–1037.

[106] Katz, J., "Lateral Aerodynamics of Delta Wings with Leading-Edge Separation," *AIAA Journal*, Vol. 22, No. 3, 1984, pp. 323–328.

[107] de Souza, C. E., da Silva, R. G. A., and Cesnik, C. E. S., "An Object-oriented Unsteady Vortex Lattice Method for Aeroelastic Analysis of Highly Flexible Wings," *10th World Congress on Computational Mechanics, Blucher Mechanical Engineering Proceedings*, 2014.

[108] Mauermann, T., "Flexible Aircraft Modelling for Flight Loads Analysis of Wake Vortex Encounters," *DLR Deutsches Zentrum fur Luft- und Raumfahrt e.V. - Forschungsberichte*, 01 2011.

[109] Murua, J., Palacios, R., and Michael, R. G. J., "Modeling of Nonlinear Flexible Aircraft Dynamics Including Free-Wake Effects," 08 2010.

[110] Murua, J., Palacios, R., and Michael, R. G. J., "Assessment of Wake-Tail Interference Effects on the Dynamics of Flexible Aircraft," *AIAA Journal*, Vol. 50, No. 7, 2012, pp. 1575–1585.

[111] Barnes, J. P., "Hybrid Blade Element and Lifting Line for Propeller or Propfan Performance," *35th AIAA Applied Aerodynamics Conference, Denver, Colorado, 5-9 June, 2017*.

[112] Tan, J., Sun, Y., and Barakos, G. N., "Unsteady Loads for Coaxial Rotors in Forward Flight Computed Using a Vortex Particle Method," *The Aeronautical Journal*, Vol. 122, No. 1251, 2018, pp. 693714.

[113] Winckelmans, G. S. and Leonard, A., "Contributions to Vortex Particle Methods for the Computation of Three-Dimensional Incompressible Unsteady Flows," *Journal of Computational Physics*, Vol. 109, No. 2, 1993, pp. 247–273.

[114] Rosenhead, L., "The Spread of Vorticity in the Wake Behind a Cylinder," *Proceedings of the Royal Society of London Series A*, Vol. 127, June 1930, pp. 590–612.

[115] Chorin, A., "Vortex Models and Boundary Layer Instability," *SIAM Journal on Scientific and Statistical Computing*, Vol. 1, No. 1, 1980, pp. 1–21.

[116] Leonard, A., "Vortex Methods for Flow Simulation," *Journal of Computational Physics*, Vol. 37, No. 3, 1980, pp. 289 – 335.

[117] Beale, J. T. and Majda, A., "Vortex methods. I - Convergence in Three Dimensions. II - Higher Order Accuracy in Two and Three Dimensions," *Mathematics of Computation*, Vol. 39, July 1982, pp. 1–52.

[118] Beale, J. T., "A Convergent 3-D Vortex Method With Grid-Free Stretching," *Mathematics of Computation*, Vol. 46, No. 174, 1986, pp. 401–424.

[119] Beale, J. T., "On the Accuracy of Vortex Methods at Large Times," *Computational Fluid Dynamics and Reacting Gas Flows*, Springer New York, New York, NY, 1988, pp. 19–32.

[120] Winckelmans, G. and Leonard, A., "Improved Vortex Methods for Three-Dimensional Flows," *Mathematical Aspects of Vortex Dynamics*, 1989, pp. 25–35.

[121] Zhao, J. and He, C., "Enhancement of Viscous Vortex Particle Method for Fundamental Rotor Wake Dynamics Simulation," *New Horizons*, , No. January, 2012, pp. 1–11.

[122] Shi, Y., Xu, G., and Wei, P., "Rotor Wake and Flow Analysis Using a Coupled Eulerian-Lagrangian Method," *Engineering Applications of Computational Fluid Mechanics*, Vol. 10, No. 1, 2016, pp. 384–402.

[123] Battey, L. S., *A Hybrid Navier Stokes/Vortex Particle Methodology for Modeling Helicopter Rotors in Forward Flight and Maneuvers*, Master's thesis, Georgia Institute of Technology, 2018.

[124] Martin, E. H., *Assessment of Panel and Vortex Particle Methods for the Modelling of Stationary Propeller Wake Wash*, Master's thesis, Memorial University of Newfoundland, 2015.

[125] Arfken, G. B. and Weber, H. J., *Mathematical Methods for Physicists*, Academic Press: San Diego, 2005.

[126] Speck, R., *Generalized Algebraic Kernels and Multipole Expansions for Massively Parallel Vortex Particle Methods*, IAS Series Volume 7, Schriften des Forschungszentrums Jlich, 2011.

[127] Cottet, G.-H. and Koumoutsakos, P. D., "Vortex Methods: Theory and Practice," *Measurement Science and Technology*, Vol. 12, No. 3, feb 2001, pp. 354–354.

[128] Cottet, G.-H., "A new approach for the analysis of Vortex Methods in two and three dimensions," *Annales de l'Institut Henri Poincare (C) Non Linear Analysis*, Vol. 5, No. 3, 1988, pp. 227 – 285.

[129] Winckelmans, G. and Leonard, A., "Weak Solutions of the Threedimensional Vorticity Equation with Vortex Singularities," *The Physics of Fluids*, Vol. 31, No. 7, 1988, pp. 1838–1839.

[130] Beale, J. and Majda, A., "High Order Accurate Vortex Methods with Explicit Velocity Kernels," *Journal of Computational Physics*, Vol. 58, No. 2, 1985, pp. 188 – 208.

[131] Hald, O. and Del Prete, V. M., "Convergence of Vortex Methods for Euler's Equations," *Mathematics of Computation*, Vol. 32, No. 143, 1978, pp. 791–809.

[132] O. Nordmark, H., "Rezoning for higher order vortex methods," *Journal of Computational Physics*, Vol. 97, No. 2, 1991, pp. 366 – 397.

[133] Rajmohan, N., Zhao, J., and He, C., "A Coupled Vortex Particle/CFD Methodology for Studying Coaxial Rotor Configurations," 01 2014.

[134] Zhao, J. and He, C., "A Viscous Vortex Particle Model for Rotor Wake and Interference Analysis," *Journal of the American Helicopter Society*, Vol. 55, 01 2010, pp. 12007–1.

[135] Zhao, J. and He, C., "Real-Time Simulation of Coaxial Rotor Configurations with Combined Finite State Dynamic Wake and VPM," Vol. 3, 05 2014.

[136] Cottet, G.-H., Ph.D. thesis, Universit Paris 6, Paris, 1982.

[137] Anderson, C. and Greengard, C., "On Vortex Methods," *SIAM Journal on Numerical Analysis*, Vol. 22, No. 3, 1985, pp. 413–440.

[138] Perlman, M., "On the Accuracy of Vortex Methods," *Journal of Computational Physics*, Vol. 59, No. 2, 1985, pp. 200 – 223.

[139] Choquin, J. P. and Lucquin-Desreux, B., "Accuracy of a Deterministic Particle Method for NavierStokes Equations," *International Journal for Numerical Methods in Fluids*, Vol. 8, 11 1988, pp. 1439 – 1458.

[140] Barba, L. A., Leonard, A., and Allen, C. B., “Advances in Viscous Vortex Methods Meshless Spatial Adaption Based on Radial Basis Function Interpolation,” *International Journal for Numerical Methods in Fluids*, Vol. 47, No. 5, 2005, pp. 387–421.

[141] Knio, O. M. and Ghoniem, A. F., “Numerical Study of a Three-Dimensional Vortex Method,” *Journal of Computational Physics*, Vol. 86, No. 1, 1990, pp. 75 – 106.

[142] Winckelmans, G. S., Salmon, J. K., Warren, M. S., Leonard, A., and Jodoin, B., “Application of Fast Parallel and Sequential Tree Codes to Computing Three-Dimensional Flows with the Vortex Element and Boundary Element Methods,” *ESAIM: Proc.*, Vol. 1, 1996, pp. 225–240.

[143] Choquin, J.-P. and Cottet, J.-H., Serie I 306, 739, C. R. Acad. Sci. Paris, 1988.

[144] Barba, L. A., *Vortex Method for Computing High-Reynolds Number Flows: Increased Accuracy with a Fully Mesh-less Formulation*, Ph.D. thesis, California Institute of Technology, USA, 2004.

[145] Gallic, S. M., *Deterministic Particle Method: Diffusion and Boundary Conditions*, Vortex Dynamics and Vortex Methods, AMS Lectures in Applied Mathematics, Vol. 28, pp. 433-480, American Mathematical Society, 1991.

[146] Eldredge, J. D., Leonard, A., and Colonius, T., “A General Deterministic Treatment of Derivatives in Particle Methods,” *Journal of Computational Physics*, Vol. 180, No. 2, 2002, pp. 686 – 709.

[147] Barnes, J. and Hut, P., “A Hierarchical $O(NlogN)$ Force-Calculation Algorithm,” *Nature*, Vol. 324, No. 4, 1986, pp. 446–449.

[148] Salmon, J. K. and Warren, M. S., “Fast Parallel Tree Codes for Gravitational and Fluid Dynamical N-Body Problems,” *Int. J. High Perform. Comput. Appl.*, Vol. 8, No. 2, June 1994, pp. 129–142.

[149] Greengard, L. and Rokhlin, V., “A Fast Algorithm for Particle Simulations,” *Journal of Computational Physics*, Vol. 135, No. 2, 1997, pp. 280 – 292.

[150] Cheng, H., Greengard, L., and Rokhlin, V., “A Fast Adaptive Multipole Algorithm in Three Dimensions,” *Journal of Computational Physics*, Vol. 155, No. 2, 11 1999, pp. 468–498.

[151] Brown, E. L., *Integrated Strain Actuation In Aircraft With Highly Flexible Composite Wings*, Ph.D. thesis, Massachusetts Institute of Technology, USA, 2003.

[152] Shearer, C. M., *Coupled Nonlinear Flight Dynamics, Aeroelasticity, and Control of Very Flexible Aircraft*, Ph.D. thesis, University of Michigan, USA, 2006.

[153] Pang, Z. Y., *Modeling, Simulation and Control of Very Flexible Unmanned Aerial Vehicle*, Ph.D. thesis, University of Michigan, USA, 2018.

[154] "UM/NAST Documentation," University of Michigan, Aug 19, 2019.

[155] Bisplinghoff, R. L. and Ashley, H., *Principles of Aeroelasticity*, John Wiley, New York, 1962.

[156] Hassig, H. J., "An Approximate True Damping Solution of the Flutter Equation by Determinant Iteration," *Journal of Aircraft*, Vol. 8, No. 11, 1971, pp. 885–889.

[157] Abel, I., "An Analytical Technique for Predicting the Characteristics of a Flexible Wing Equipped with an Active Flutter-Suppression System and Comparison with Wind-Tunnel Data," Tech. rep., Technical Report NASA-TP-1367, L-12567, February 1979.

[158] Hallissy, B. and Cesnik, C. E. S., "High-fidelity Aeroelastic Analysis of Very Flexible Aircraft," *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*.

[159] McNamara, J. J. and Friedmann, P. P., "Flutter Boundary Identification for Time-Domain Computational Aeroelasticity," *AIAA Journal*, Vol. 45, No. 7, 2007, pp. 1546–1555.

[160] Hammond, C. E. and Dogget, J. R. V., "An Analytical Technique for Predicting the Characteristics of a Flexible Wing Equipped with an Active Flutter-Suppression System and Comparison with Wind-Tunnel Data," Tech. rep., NASA Scientific and Technical Information Office, Washington, October 1975.

[161] Bennett, R. G. and Desmarais, R., "Curve-Fitting of Aeroelastic Transient Response Data with Exponential Functions," Tech. rep., NASA SP-415, October 1975.

[162] Onoda, J., "Estimation of Dynamic Characteristics of a Wing from the Random Response to Turbulence," *Journal of the Japan Society for Aeronautical and Space Sciences*, Vol. 26, No. 299, 1978, pp. 649–656.

[163] Silva, W. A., "AEROM: NASAs Unsteady Aerodynamic and Aeroelastic Reduced-Order Modeling Software," *Aerospace*, Vol. 5, No. 2, 2018.

[164] Teixeira, P. C. and Cesnik, C. E. S., "Propeller Influence on the Stability of HALE Aircraft," *31st Congress of the International Council of the Aeronautical Sciences*, September 2018.

[165] Kerschen, G., Golinval, J.-C., Vakakis, A. F., and Bergman, L. A., "The Method of Proper Orthogonal Decomposition for Dynamical Characterization and Order Reduction of Mechanical Systems: An Overview," *Nonlinear Dynamics*, Vol. 41, No. 1, Aug 2005, pp. 147–169.

[166] Liang, Y. C., Lee, H. P., Lim, S. P., Lin, W., Lee, K. H., and Wu, C. G., "Proper Orthogonal Decomposition and its Applications Part I: Theory," *Journal of Sound and Vibration*, Vol. 252, No. 3, 2002, pp. 527 – 544.

[167] Lumley, J. L., *Stochastics Tools in Turbulence*, New York: Academic Press, 1970.

[168] Fukunaga, K., *Introduction to Statistical Pattern Recognition*, San Diego: Academic Press Inc., 1990.

[169] Jolliffe, I. T., *Principal Component Analysis*, New York: Springer-Verlag Inc, 1986.

[170] Klema, V. C. and Laub, A. J., "The Singular Value Decomposition: Its Computation and Some Applications." *IEEE: Transactions on Automatic Control*, Vol. 25, pp. 164–176.

[171] "MathWorks Doumentation: System Identification Toolbox," https://www.mathworks.com/help/ident/gs/about-system-identification.html#bsguh_9, Accessed: 2019-08-05.

[172] Zadeh, L. A., "On the Identification Problem," *IRE Transactions on Circuit Theory*, Vol. 3, No. 4, 1956, pp. 277 – 281.

[173] Ljung, L., "Perspectives on System Identification," *Annual Reviews in Control*, Vol. 34, No. 1, 2010, pp. 1 – 12.

[174] Gevers, M., "A Personal View on the Development of System Identification," *IFAC Proceedings Volumes*, Vol. 36, No. 16, 2003, pp. 747 – 758, 13th IFAC Symposium on System Identification (SYSID 2003), Rotterdam, The Netherlands, 27-29 August, 2003.

[175] Pajonk, O., "Overview of System Identification with Focus on Inverse Modeling Literature Review," Institute of Scientific Computing, Braunschweig, Germany, 2009, 2009.

[176] Van Overschee, P. and De Moor, B., "N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems," *Automatica*, Vol. 11, No. 2, 1963, pp. 431–441.

[177] Rainiere, C. and Fabbrocino, G., *Operational Modal Analysis of Civil Engineering Structures*, Spring-Verlag New York, 2014, pp. 182-183.

[178] Cesnik, C. E. S., Senatore, P. J., Su, W., Atkins, E. M., and Shearer, C. M., "X-HALE: A Very Flexible Unmanned Aerial Vehicle for Nonlinear Aeroelastic Tests," *AIAA Journal*, Vol. 50, No. 12, 2012, pp. 2820–2833.

[179] Jones, R. J., *Development of a Very Flexible Testbed Aircraft for the Validation of Nonlinear Aeroelastic Codes*, Ph.D. thesis, University of Michigan, Ann Arbor, MI, 2017.

[180] Cho, J. and Williams, M. H., "Propeller-Wing Interaction Using a Frequency Domain Panel Method," *Journal of Aircraft*, Vol. 27, No. 3, 1989, pp. 196–203.

[181] Usab, W. J., Lee, K. H., and Sullivan, J. P., "A Comparison of Numerical Simulation and Experimental Measurements of Flow through Propellers," *AIAA 26th Aerospace Sciences Meeting*, AIAA Paper 88-0367, January 1988.

[182] Brandt, J. B., *Small-scale Propeller Performance at Low Speeds*, Master's thesis, University of Illinois at Urbana-Champaign, Urbana-Champaign, 2005.

[183] Khan, W. and Nahon, M., "Development and Validation of a Propeller Slipstream Model for Unmanned Aerial Vehicles," *Journal of Aircraft*, Vol. 52, No. 6, 2015, pp. 1985–1994.

[184] Antcliff, K. R., Whiteside, S. K. S., Kohlman, L. W., and Silva, C., "Baseline Assumptions and Future Research Areas for Urban Air Mobility Vehicles," *AIAA Scitech Forum*, 2019.

[185] Patterson, A., Gahlawat, A., and N., H., "Propeller Phase Synchronization for Small Distributed Electric Vehicles," *AIAA Scitech Forum, San Diego, California, 7-11 January, 2019*, 2019.

[186] Kebbie-Anthony, A. B., Gumerov, N., Preidikman, S., Balachandran, B., and Azarm, S., "Fast Multipole Method for Nonlinear, Unsteady Aerodynamic Simulations," *2018 AIAA Modeling and Simulation Technologies Conference, Kissimmee, Florida, 812 January, 2018*.

[187] Kane, T. R. and Levinson, D. A., *Dynamic Theory and Applications*, The Internet-First University Press, NY, 2005.

APPENDIX A

Properties of 16-m Wing

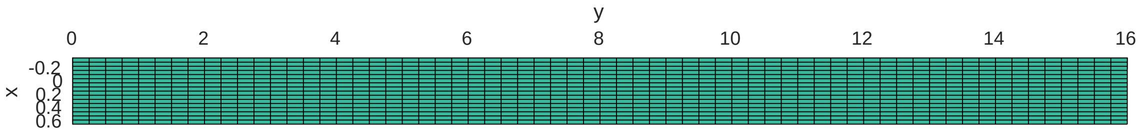


Figure A.1: Vortex Lattice representation of the 16-m wing model (units: meters)

The properties of the 16-m wing model introduced in Chapter 5 are given here as a reference. The data provided is related to a discretization of the wing in 32 UM/NAST beam elements, with 3 nodes each, totalizing 65 structural nodes evenly spaced. Table A.1 summarizes data about the stiffness distribution with values given for the mid node of each element. Table A.2 contains information about the distributed mass and inertia at the starting and ending node of each element. Table A.3 contains information about the lumped masses located at the indicated structural nodes. Finally, Table A.4 contains relevant information for the aerodynamic modeling.

Table A.1: Stiffness data for 16-m wing model. Values are for the mid node of each element.

Element	k_{11} (EA)	k_{22} (GJ)	k_{33} (EI_y)	k_{44} (EI_z)
1	5.68×10^8	2.88×10^6	1.91×10^6	7.65×10^6
2	5.68×10^8	2.70×10^6	1.80×10^6	7.19×10^6
3	5.68×10^8	2.54×10^6	1.69×10^6	6.74×10^6
4	5.68×10^8	2.37×10^6	1.58×10^6	6.31×10^6
5	5.68×10^8	2.22×10^6	1.47×10^6	5.90×10^6
6	5.68×10^8	2.07×10^6	1.37×10^6	5.50×10^6
7	5.68×10^8	1.92×10^6	1.28×10^6	5.12×10^6
8	5.68×10^8	1.79×10^6	1.19×10^6	4.75×10^6
9	5.68×10^8	1.65×10^6	1.10×10^6	4.40×10^6
10	5.68×10^8	1.53×10^6	1.02×10^6	4.06×10^6
11	5.68×10^8	1.41×10^6	9.36×10^5	3.74×10^6
12	5.68×10^8	1.29×10^6	8.60×10^5	3.44×10^6
13	5.68×10^8	1.18×10^6	7.88×10^5	3.15×10^6
14	5.68×10^8	1.08×10^6	7.19×10^5	2.88×10^6
15	5.68×10^8	9.86×10^5	6.55×10^5	2.62×10^6
16	5.68×10^8	8.95×10^5	5.95×10^5	2.38×10^6
17	5.68×10^8	8.10×10^5	5.39×10^5	2.15×10^6
18	5.68×10^8	7.32×10^5	4.86×10^5	1.95×10^6
19	5.68×10^8	6.59×10^5	4.38×10^5	1.75×10^6
20	5.68×10^8	5.92×10^5	3.93×10^5	1.57×10^6
21	5.68×10^8	5.31×10^5	3.53×10^5	1.41×10^6
22	5.68×10^8	4.76×10^5	3.16×10^5	1.27×10^6
23	5.68×10^8	4.27×10^5	2.84×10^5	1.13×10^6
24	5.68×10^8	3.84×10^5	2.55×10^5	1.02×10^6
25	5.68×10^8	3.46×10^5	2.30×10^5	9.21×10^5
26	5.68×10^8	3.15×10^5	2.09×10^5	8.37×10^5
27	5.68×10^8	2.90×10^5	1.92×10^5	7.70×10^5
28	5.68×10^8	2.70×10^5	1.80×10^5	7.18×10^5
29	5.68×10^8	2.57×10^5	1.71×10^5	6.82×10^5
30	5.68×10^8	2.49×10^5	1.65×10^5	6.62×10^5
31	5.68×10^8	2.47×10^5	1.64×10^5	6.57×10^5
32	5.68×10^8	2.51×10^5	1.67×10^5	6.69×10^5

Table A.2: Distributed mass of the 16-m wing model. Values of inertia are for the starting and ending node of each element.

Element	Mass	I_{xx} (start)	I_{xx} (end)	I_{yy} (start)	I_{yy} (end)	I_{zz} (start)	I_{zz} (end)
[-]	[kg/m]	[kgm]	[kgm]	[kgm]	[kgm]	[m]	[m]
1	6.4	8.88×10^2	8.36×10^2	2.22×10^2	2.09×10^2	8.88×10^2	8.36×10^2
2	6.4	8.36×10^2	7.84×10^2	2.09×10^2	1.96×10^2	8.36×10^2	7.84×10^2
3	6.4	7.84×10^2	7.35×10^2	1.96×10^2	1.84×10^2	7.84×10^2	7.35×10^2
4	6.4	7.35×10^2	6.87×10^2	1.84×10^2	1.72×10^2	7.35×10^2	6.87×10^2
5	6.4	6.87×10^2	6.42×10^2	1.72×10^2	1.60×10^2	6.87×10^2	6.42×10^2
6	6.4	6.42×10^2	5.98×10^2	1.60×10^2	1.49×10^2	6.42×10^2	5.98×10^2
7	6.4	5.98×10^2	5.55×10^2	1.49×10^2	1.39×10^2	5.98×10^2	5.55×10^2
8	6.4	5.55×10^2	5.15×10^2	1.39×10^2	1.29×10^2	5.55×10^2	5.15×10^2
9	6.4	5.15×10^2	4.76×10^2	1.29×10^2	1.19×10^2	5.15×10^2	4.76×10^2
10	6.4	4.76×10^2	4.39×10^2	1.19×10^2	1.10×10^2	4.76×10^2	4.39×10^2
11	6.4	4.39×10^2	4.04×10^2	1.10×10^2	1.01×10^2	4.39×10^2	4.04×10^2
12	6.4	4.04×10^2	3.71×10^2	1.01×10^2	9.27×10^3	4.04×10^2	3.71×10^2
13	6.4	3.71×10^2	3.39×10^2	9.27×10^3	8.48×10^3	3.71×10^2	3.39×10^2
14	6.4	3.39×10^2	3.09×10^2	8.48×10^3	7.73×10^3	3.39×10^2	3.09×10^2
15	6.4	3.09×10^2	2.81×10^2	7.73×10^3	7.03×10^3	3.09×10^2	2.81×10^2
16	6.4	2.81×10^2	2.55×10^2	7.03×10^3	6.38×10^3	2.81×10^2	2.55×10^2
17	6.4	2.55×10^2	2.31×10^2	6.38×10^3	5.76×10^3	2.55×10^2	2.31×10^2
18	6.4	2.31×10^2	2.08×10^2	5.76×10^3	5.20×10^3	2.31×10^2	2.08×10^2
19	6.4	2.08×10^2	1.87×10^2	5.20×10^3	4.67×10^3	2.08×10^2	1.87×10^2
20	6.4	1.87×10^2	1.68×10^2	4.67×10^3	4.19×10^3	1.87×10^2	1.68×10^2
21	6.4	1.68×10^2	1.50×10^2	4.19×10^3	3.76×10^3	1.68×10^2	1.50×10^2
22	6.4	1.50×10^2	1.35×10^2	3.76×10^3	3.37×10^3	1.50×10^2	1.35×10^2
23	6.4	1.35×10^2	1.21×10^2	3.37×10^3	3.02×10^3	1.35×10^2	1.21×10^2
24	6.4	1.21×10^2	1.09×10^2	3.02×10^3	2.72×10^3	1.21×10^2	1.09×10^2
25	6.4	1.09×10^2	9.86×10^3	2.72×10^3	2.47×10^3	1.09×10^2	9.86×10^3
26	6.4	9.86×10^3	9.01×10^3	2.47×10^3	2.25×10^3	9.86×10^3	9.01×10^3
27	6.4	9.01×10^3	8.34×10^3	2.25×10^3	2.08×10^3	9.01×10^3	8.34×10^3
28	6.4	8.34×10^3	7.84×10^3	2.08×10^3	1.96×10^3	8.34×10^3	7.84×10^3
29	6.4	7.84×10^3	7.53×10^3	1.96×10^3	1.88×10^3	7.84×10^3	7.53×10^3
30	6.4	7.53×10^3	7.39×10^3	1.88×10^3	1.85×10^3	7.53×10^3	7.39×10^3
31	6.4	7.39×10^3	7.43×10^3	1.85×10^3	1.86×10^3	7.39×10^3	7.43×10^3
32	6.4	7.43×10^3	7.64×10^3	1.86×10^3	1.91×10^3	7.43×10^3	7.64×10^3

Table A.3: Lumped mass data. Masses are located on the position of the indicated structural nodes (model has 32 elements with 3 nodes each, totalizing 65 structural nodes. Node 0 is at the root).

Node	Mass	I_{xx}
	[kg]	[kgm^2]
4	5.000	0.200
7	4.970	0.199
10	4.939	0.198
13	4.909	0.196
16	4.879	0.195
19	4.848	0.194
22	4.818	0.193
25	4.788	0.192
28	4.758	0.190
31	4.727	0.189
34	4.697	0.188
37	4.667	0.187
40	4.636	0.185
43	4.606	0.184
46	4.576	0.183
49	4.545	0.182
52	4.515	0.181
55	4.485	0.179
58	4.455	0.178
61	4.424	0.177
64	4.394	0.176
67	4.364	0.175
70	4.333	0.173
73	4.303	0.172
76	4.273	0.171
79	4.242	0.170
82	4.212	0.168
85	4.182	0.167
88	4.152	0.166
91	4.121	0.165
94	4.091	0.164
96	4.061	0.162

Table A.4: 16-m wing aerodynamic model description.

Span	Chord	E. Axis	Dihedral	Incidence	Airfoil
[m]	[m]	[% chord]	[deg]	[deg]	[-]
16	1	35	0	0	FLAT

APPENDIX B

Properties of X-HALE Model Used in This Dissertation

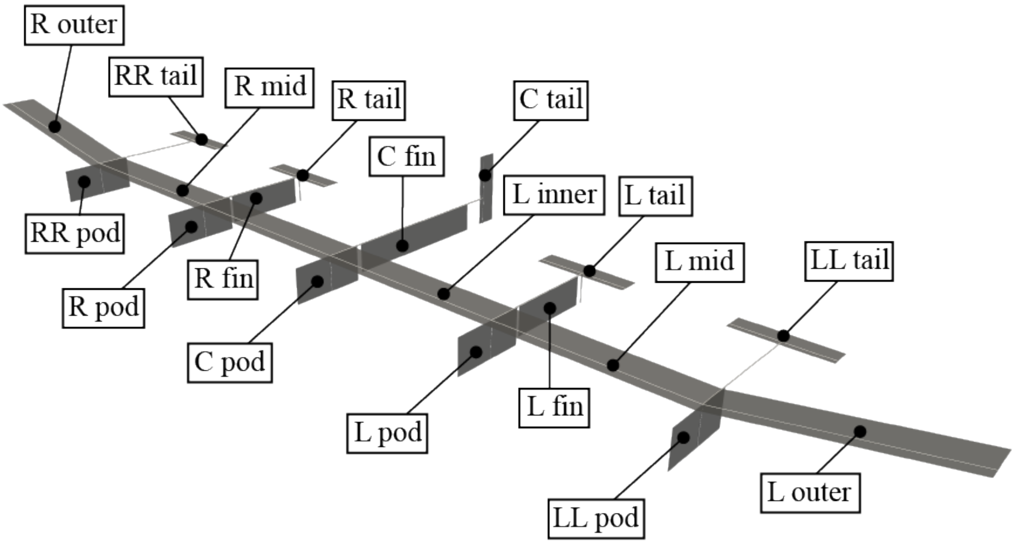


Figure B.1: Nomenclature followed in the tables describing X-HALE properties [1]

The properties of the X-HALE model used in this work and introduced in Chapter 5 are given here as a reference. Table B.1 summarizes the data associated with the stiffness distribution. Table B.2 contains information about each member distributed mass. Table B.3 contains information about the lumped masses. Finally, Table B.4 contains relevant information for the aerodynamic modeling.

The plastic blade mass (0.023 kg) is incorporated as a lumped mass at the propeller hub location. The parts of the aircraft referenced on the tables are illustrated in Figure B.1.

Table B.1: Stiffness data for X-HALE model.

Member	k_{11} (EA)	k_{22} (GJ)	k_{33} (EI_y)	k_{44} (EI_z)	k_{14} (axial-IP)	k_{34} (OOP-IP)
[-]	[Nm^2]	[Nm^2]	[Nm^2]	[Nm^2]	[Nm^2]	[Nm^2]
R inner	2.14×10^6	5.93×10^1	1.12×10^2	6.35×10^3	0	4.63×10^1
R mid	2.14×10^6	5.93×10^1	1.12×10^2	6.35×10^3	0	4.63×10^1
R outer	2.14×10^6	5.93×10^1	1.12×10^2	6.35×10^3	0	4.63×10^1
fuselage	5.39×10^7	5.39×10^7	5.39×10^7	5.39×10^7	0	0
tail	3.21×10^6	2.14×10^1	9.10×10^1	4.27×10^3	7.44×10^4	2.26×10^{-6}
fin	5.39×10^7	5.39×10^7	5.39×10^7	5.39×10^7	0	0

Table B.2: Distributed mass of the X-HALE model. Member abbreviations are defined in Figure B.1.

Member	Mass	I_{xx}	I_{yy}	I_{zz}	I_{yz}	x_{cg}	y_{cg}	z_{cg}
[-]	[kg/m]	[kgm]	[kgm]	[kgm]	[kgm]	[m]	[m]	[m]
R inner	3.94×10^{-1}	8.1×10^{-4}	1.22×10^{-5}	7.97×10^{-4}	6.5×10^{-6}	0	2.94×10^{-2}	0
R mid	3.94×10^{-1}	8.1×10^{-4}	1.22×10^{-5}	7.97×10^{-4}	6.5×10^{-6}	0	2.94×10^{-2}	0
R outer	5.0×10^{-1}	8.1×10^{-4}	1.22×10^{-5}	7.97×10^{-4}	6.5×10^{-6}	0	2.14×10^{-2}	0
fuselage	0.0429	2.91×10^{-9}	1.46×10^{-9}	1.46×10^{-9}	0	0	0	0
tail	0.2614	1.6×10^{-4}	2.910×10^{-6}	1.57×10^{-4}	0	0	0.0144	0
pod	1×10^{-8}	2×10^{-8}	1×10^{-8}	1×10^{-8}	0	0	0	0
C fin	0.5092	3.19×10^{-3}	9.34×10^{-5}	3.28×10^{-3}	0	0	0	0
L/R fin	0.3208	8.17×10^{-4}	5.88×10^{-5}	8.76×10^{-4}	0	0	0	0

Table B.3: Lumped mass data. Relative positions are given with respect to the pod/wing spar intersection node. Frame of reference: x right wing, y upstream, z up. Member abbreviations are defined in Figure B.1.

Mass [kg]	x_{cg} [m]	y_{cg} [m]	z_{cg} [m]	I_{xx} [kgm^2]	I_{xy} [kgm^2]	I_{xz} [kgm^2]	I_{yy} [kgm^2]	I_{yz} [kgm^2]	I_{zz} [kgm^2]
C pod									
0.3746	0	0.1	0	1.15×10^{-3}	0	0	8.90×10^{-4}	0	8.90×10^{-4}
1.0462	3.97×10^{-3}	0.0612	-0.0168	1.48×10^{-2}	2.32×10^{-4}	2.27×10^{-5}	2.82×10^{-3}	4.50×10^{-4}	2.50×10^{-4}
0.023	0	0.260	-0.023	0	0	0	0	0	0
L pod									
0.548	-0.01	0.090	0	1.54×10^{-3}	0	0	8.90×10^{-4}	0	8.90×10^{-4}
0.929	2.14×10^{-3}	0.04	-1.39×10^{-2}	1.13×10^{-2}	-1.21×10^{-3}	1.06×10^{-5}	3.21×10^{-3}	4.60×10^{-5}	8.48×10^{-3}
0.023	0	0.259	-0.023	0	0	0	0	0	0
R pod									
0.548	-0.01	0.090	0	1.54×10^{-3}	0	0	8.90×10^{-4}	0	8.90×10^{-4}
0.929	2.14×10^{-3}	0.04	1.39×10^{-2}	1.13×10^{-2}	-1.21×10^{-3}	1.06×10^{-5}	3.21×10^{-3}	4.60×10^{-5}	8.48×10^{-3}
0.023	0	0.259	-0.023	0	0	0	0	0	0
LL pod									
0.571	-0.01	0.091	0	1.54×10^{-3}	0	0	8.90×10^{-4}	0	8.90×10^{-4}
0.929	2.14×10^{-3}	0.04	-1.39×10^{-2}	1.13×10^{-2}	-1.21×10^{-3}	1.06×10^{-5}	3.21×10^{-3}	4.60×10^{-5}	8.48×10^{-3}
0.023	0	0.259	-0.023	0	0	0	0	0	0
RR pod									
0.571	-0.01	0.091	0	1.54×10^{-3}	0	0	8.90×10^{-4}	0	8.90×10^{-4}
0.929	2.14×10^{-3}	0.04	-1.39×10^{-2}	1.13×10^{-2}	-1.21×10^{-3}	1.06×10^{-5}	3.21×10^{-3}	4.60×10^{-5}	8.48×10^{-3}
0.023	0	0.259	-0.023	0	0	0	0	0	0

Table B.4: X-HALE aerodynamic model description.

Member	Span	Chord	Elastic Axis	Dihedral	Incidence	Airfoil
[-]	[m]	[m]	[% chord]	[deg]	[deg]	[-]
inner/mid wings	1	0.2	28.8	0	5	EMX-07
outer wings	1	0.2	28.8	10	5	EMX-07
C tail	0.385	0.11	32.35	0	0	flat plate
LL/L/R/RR tail	0.48	0.11	32.35	0	0	flat plate
C fin	0.15	0.78	122.56	0	0	flat plate
L/C/R pod	0.184	0.38	60.93	0	0	flat plate

APPENDIX C

Analytical Example for Gyroscopic Loads Verification

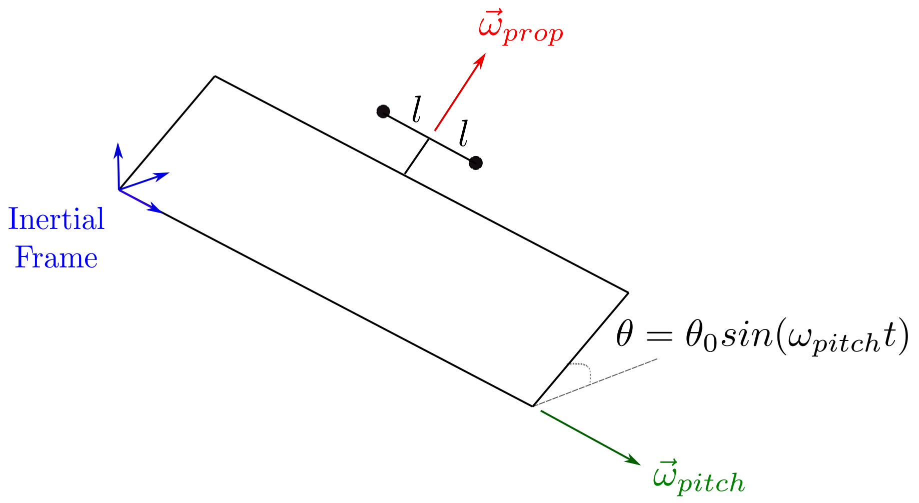


Figure C.1: Pitching wing with rotating propeller represented by a rigid, massless rod, with concentrated masses on its ends

In order to verify the implementation of the additional term given by Eq. 2.85, which accounts for propeller gyroscopic moment, a comparison of the additional moment was performed against the analytical solution for a simple case of a rigid wing undergoing a pitching sinusoidal motion with a propeller located in its middle (Figure C.1). Table C.1 summarizes the data for this comparison. In this case, the propeller blades are represented as rigid massless rods with concentrated masses on its ends. The wing is undergoing a pitching motion given by:

$$\theta = \theta_0 sin(\omega_{pitch} t) \tag{C.1}$$

where θ is the pitching angle, θ_0 the angular amplitude of the oscillatory motion and ω_{pitch} the pitch angular velocity. For this simple example, an analytical solution is possible, as described next.

Table C.1: Parameters used for the verification of gyroscopic moment

Propeller RPM	6000
Propeller mass	0.027 kg
Pitch motion amplitude	30 degrees
Pitch frequency	5 Hz

In this problem one is interested in the gyroscopic moment of the rotating rod/-mass body transmitted to the point where the propeller is attached to the wing structure. Note that this results does not depend on the spanwise position of the propeller in this rigid wing example. The analytical solution for the gyroscopic moment was obtained based on the theory presented in Kane and Levinson [187] and described below.

The central angular momentum $\vec{H}$ of a rigid body B in a reference frame A can be expressed as:

$$\vec{H} = I_1 \omega_1 \hat{b}_1 + I_2 \omega_2 \hat{b}_2 + I_3 \omega_3 \hat{b}_3 \tag{C.2}$$

where $\hat{b_1}, \hat{b_2}, \hat{b_3}$ are a set of mutually perpendicular unit vectors fixed in B, $\omega_j =^A \vec{\omega}^B \cdot \hat{b}_j$, and I_1, I_2, I_3 are the central principal moments of inertia of the rigid body B.

The gyroscopic moment acting on the rigid body B is given by the first time derivative of the central angular momentum H with relation to the inertial frame A, and can be written in terms of the unit vectors attached to B as:

$$\frac{{}^{A}d\vec{H}}{dt} = M_1\hat{b}_1 + M_2\hat{b}_2 + M_3\hat{b}_3 \tag{C.3}$$

where M_i are values that we want to determine.

Applying identity given by Eq. 2.75, in Chapter 2, one has:

$$\frac{{}^{A}d\vec{H}}{dt} = \frac{{}^{B}d\vec{H}}{dt} +^{A} \vec{\omega}^{B} \times \vec{H} \tag{C.4}$$

Also, from Eq. C.2:

$$\frac{{}^{B}d\vec{H}}{dt} = I_1\dot{\omega}_1\hat{b}_1 + I_2\dot{\omega}_2\hat{b}_2 + I_3\dot{\omega}_3\hat{b}_3 \tag{C.5}$$

and by definition:

$${}^{A}\vec{\omega}^{B} = \omega_1\hat{b}_1 + \omega_2\hat{b}_2 + \omega_3\hat{b}_3 \tag{C.6}$$

Then:

$${}^{A}\vec{\omega}^{B} \times \vec{H} = (\omega_2 I_3 \omega_3 - \omega_3 I_2 \omega_2)\hat{b}_1 + ... \tag{C.7}$$

Substituting (C.5) and (C.7) into (C.4):

$$\frac{{}^{A}d\vec{H}}{dt} = [I_1\dot{\omega}_1 - (I_2 - I_3)\omega_2\omega_3]\hat{b}_1 + ... \tag{C.8}$$

Finally, comparing (C.8) and (C.3), one has:

$$M_1 = I_1\dot{\omega}_1 - (I_2 - I_3)\omega_2\omega_3 \tag{C.9}$$

$$M_2 = I_2\dot{\omega}_2 - (I_3 - I_1)\omega_3\omega_1 \tag{C.10}$$

$$M_3 = I_3\dot{\omega}_3 - (I_1 - I_2)\omega_1\omega_2 \tag{C.11}$$

Then, considering A as the inertial frame and B as a frame attached to the propeller blades (represented by the rigid rod), and remembering that the gyroscopic moment transmitted to the structural node is the negative of the gyroscopic moment acting on the rod, the moment transmitted can be finally calculated by:

$$M_{gyro} = -\frac{{}^{A}d\vec{H}}{dt} \tag{C.12}$$

where $\frac{{}^{A}d\vec{H}}{dt}$ is given by Eq. C.3.

The following MATLAB script contains the implementation of the analytical solution for this case:

```
% theoretical test case for propeller inertial loads


%Parameters used:
l = 0.085; % (m); this is the length from the hub to the mass position
mass = 0.027/2; %for two blade, mass of each blade equals (mass prop) / 2
time = 1;
w_prop = 6000 * (2*pi)/60; % propeller angular velocity
theta_0_degree = 30; % pitch angular velocity
w_pitch = 10 * pi; % 2 * pi * f, f = 5 Hz


% Principal moment of inertias of rod with mass at the tips:
I1 = 2 * mass * l^2;
I2 = 2 * mass * l^2;
I3 = 0;
```

```
syms t

theta = theta_0_degree * sin (w_pitch * t) * pi/180;
theta_prop = w_prop * t;

% Here we are considering 3 frames: b -> blade frame;
% moves with relation to prop frame
%   p -> prop frame, atatched to prop hub, does not rotate with
% relation to hub
%   B -> inertial frame (for this example, will be coincident to body frame
% defined in UM/NAST

Mp2B = [1 0 0; 0 cos(theta) -sin(theta); 0 sin(theta) cos(theta)];
Mb2p = [cos(theta_prop) 0 sin(theta_prop);
         0 1 0;
        -sin(theta_prop) 0 cos(theta_prop)];
Mb2B = Mp2B * Mb2p;

%angular velocity of blade frame with relation to propeller frame:
P_omega_b = [0;w_prop;0];

% angular velocity of blade frame with relation to propeller frame,
% written in body frame:
P_omega_b_Bf = Mb2B * P_omega_b;

% angular velocity of propeller frame with relation to body frame:
B_omega_P_Bf = [diff(theta); 0; 0];
```

```
%Writting blade frame unit vectors in terms of body frame system:
b1_Bf = Mb2B * [1; 0; 0];
b2_Bf = Mb2B * [0; 1; 0];
b3_Bf = Mb2B * [0; 0; 1];

% Calculating w_i's:
w1 = dot(P_omega_b_Bf + B_omega_P_Bf, b1_Bf);
w2 = dot(P_omega_b_Bf + B_omega_P_Bf, b2_Bf);
w3 = dot(P_omega_b_Bf + B_omega_P_Bf, b3_Bf);

% Calculating first derivative w_i's
w1p = diff(w1);
w2p = diff(w2);
w3p = diff(w3);

%Calculating the coefficients M_1, M_2 and M_3:
M1 = I1 * w1p - (I2 - I3) * w2 * w3;
M2 = I2 * w2p - (I3 - I1) * w3 * w1;
M3 = I3 * w3p - (I1 - I2) * w1 * w2;

%Finally, transferring moments to node, remembering the moments transmitted
% are minus the moments acting on the rotating rod:
M = Mb2B * [-M1; -M2; -M3];

clf
figure(1)
```

```
hold on
fplot(M(1),[0,time],'b')
fplot(M(2),[0,time],'r')
fplot(M(3),[0,time],'g')
hold on
xlabel('time (sec)')
ylabel('Gyroscopic Moment (Nm)')
legend('Mx from theoretical','My from theoretical','Mz from theoretical')
```